Resource Reservation in Shared and Switched Demand Priority Local Area Networks

by

Peter Kim

A dissertation submitted for the degree of

Doctor of Philosophy

of the

UNIVERSITY OF LONDON

Department of Computer Science
University College London

September 1998

Abstract

Packet switching data networks such as the Internet are migrating towards Integrated Services networks. To provide end-to-end service guarantees across those networks requires supporting mechanisms on all links along the data path including Local Area Networks (LAN) which are typically deployed at the leaves of the Internet. There is however no standard mechanism for building advanced services in existing LANs because the medium access mechanisms of these technologies differ.

This dissertation is about providing Integrated Services in IEEE 802.12 networks. 802.12 is the standard for a shared 100 Mbit/s LAN. Its Medium Access Control (MAC) protocol is called Demand Priority. In this work, we have proved that the Guaranteed and the Controlled Load service proposed for a future multi-services Internet, can be provided across shared and switched 802.12 LANs, even when the network is overloaded with best effort traffic. This is achieved using resource reservation with admission control and based on the Integrated Services Packet Network (ISPN) framework.

The key design constraints of our reservation scheme were the variable data throughput in 802.12 networks and the fact that hubs are not able to identify and isolate single data connections. We found that the Demand Priority signalling overhead may have a significant impact on the network performance when shared topologies become large or small sized data packets are used for data transmissions. To describe this overhead, a theoretical analysis is performed in which we derive results for topology and physical layer specific network parameters. Measurements in different test networks were used to confirm these results.

The following part of the dissertation defines the admission control conditions for the Guaranteed service. When used with the parameters derived in the analysis, we find that these conditions enable us to accurately compute the minimum network throughput and thus the resource allocation limit. We also studied the delay characteristics and how network resource can be partitioned. The Controlled Load service was designed based on traffic aggregation and simple static priority scheduling within switches. This ensures low implementation costs and a deployment in existing or next generation LAN switches.

Acknowledgments

There are a number of people which I would like to thank for their support and encouragement during the years of my doctoral research. First, I would like to express my gratitude to my supervisor, Jon Crowroft, for his help, assistance and for letting me pursue my own ideas. In spite of the distance to Bristol, he somehow always seemed to know when I needed support and encouragement. Thanks Jon.

This thesis would further not have been written without the support from my employer Hewlett-Packard. I would especially like to thank John Grinham and Steve Wright for their understanding and for giving me the environment to conduct my research. I have also benefited from many stimulating discussions with my colleagues at the Hewlett Packard Research Laboratories in Bristol. I would like to thank Costas Calamvokis, David Cunningham, Chris Dalton, Aled Edwards, John Grinham, Dirk Kuhlmann, Neil O'Connell, Michael Spratt and Greg Watson for sharing their expertise with me. I further thank Cristina and Dirk Kuhlmann for their encouragement and for making life more bearable throughout the more difficult times in my research.

Contents

List of Figures **xi**

List of Tables **XV**

Mathematical Notation **XVII**

1. Introduction **1**

1.1 Background ... 1
1.1.1 Motivation for an Integrated Services Network ... 1
1.1.2 Circuit Switching versus Packet Switching ... 2
1.1.3 Congestion in Packet Switched Networks ... 4
1.1.4 Reactive Congestion Control ... 5
1.1.5 Proactive Congestion Control ... 6
1.1.6 Combining Reactive and Proactive Approaches in Integrated Services Networks ... 7
1.2 Problem Specification ... 8
1.2.1 Framework ... 8
1.2.2 Hypothesis and Research Goals ... 9
1.2.3 Thesis Contributions ... 10
1.2.4 Research Methodology ... 11
1.3 Overview of the Thesis ... 11

2. Integrated Services Packet Network Architecture (ISPN) **13**

2.1 Quality of Service Requirements ... 13
2.1.1 Application Classes ... 14
2.1.2 Application Characteristics and QoS Requirements ... 14
2.2 Advanced Services and Their Services Interfaces ... 17
2.2.1 Definitions ... 17
2.2.2 The Guaranteed Service ... 18
2.2.3 The Controlled Load Service ... 20
2.3 Packet Scheduling and Admission Control ... 20
2.3.1 Desired Properties ... 21
2.3.2 On Fundamental Design Choices and Trade Offs ... 22
2.3.3 Related Work ... 24
2.4 Dynamic Reservation Setup in the ISPN ... 26
2.4.1 Fundamental RSVP Concepts ... 26
2.4.2 Reservation Model - OPWA ... 27
2.5 Policy Control, Reservation Request Authentication and Pricing Issues ... 28
2.6 The ISSLL Framework for Reserving Resources in Shared and Switched LANs ... 29
2.6.1 Definitions ... 29
2.6.2 Mapping Integrated Services onto IEEE 802 MAC Service Mechanisms ... 29
2.6.3 The Difference to the Network Layer Integrated Services Architecture ... 30
2.6.4 Link Layer Signalling Issues ... 31
2.6.5 Why Resource Reservation in LANs ? ... 32
2.7 Relation to the Differentiated Services Approach ... 34

3. Measurement Methodology 37
3.1 Clock Terminology and Characteristics ... 37
3.2 Generating Realistic Traffic Patterns in the Test Network ... 38
3.2.1 The Test Network ... 38
3.2.2 Traffic Trace Driven Measurements ... 39
3.3 A Kernel Based Traffic Monitor ... 40
3.3.1 Design and Implementation Issues ... 40
3.3.2 Performance and Measurement Accuracy ... 42
3.4 A Trace Driven Traffic Generator ... 44
3.4.1 Design and Implementation Issues ... 44
3.4.2 The Accuracy of the Approach ... 45
3.5 Measuring the Throughput in Shared and Switched LANs ... 47
3.6 Measuring End-to-End Delay ... 50
3.6.1 A Centralistic Measurement Approach ... 50
3.6.2 Accuracy Issues and Alternative Approaches ... 51
3.7 Measuring the Packet Loss Rate ... 52

4. Quality of Service under Network Overload 55
4.1 Classifying 802.12 Networks ... 55
4.2 Traffic Traces and Traffic Models ... 57
4.2.1 Application Test Traces ... 57
4.2.2 Source Model Traces and Parameter Selection ... 62
4.3 802.12 Network Overload Behaviour ... 65
4.3.1 Available Bandwidth in Cascaded Network Topologies ... 65
4.3.2 Available Bandwidth in Switched Networks ... 67
4.3.3 Network Delay and Loss Characteristics ... 68
4.3.4 Impact of the Amount of Buffer Space within Switches ... 72
4.4 Approaches to maintain QoS under Overload Conditions ... 75
4.5 Summary ... 77

5. 802.12 Network Analysis 79
5.1 802.12 and Demand Priority ... 79
5.1.1 Network Operation ... 80
5.1.2 Performance Parameters and their Dependencies ... 81
5.2 Performance Parameters for the UTP Physical Layer ... 82
5.2.1 The Per-Packet Overhead in Single Hub Networks ... 82
5.2.2 The Per-Packet Overhead in Multi-Hub Networks ... 86
5.2.3 The Per-Packet Overhead in Half-Duplex Switched Links ... 88
5.2.4 The Interrupt Time in Single Hub Networks ... 90
5.2.5 The Interrupt Time in Multi-Hub Networks ... 93
5.2.6 The Interrupt Time in Half-Duplex Switched Links ... 99
5.3 Performance Parameters for the Fibre-Optic Physical Layer ... 102
5.3.1 The Per-Packet Overhead in Cascaded Networks ... 103
5.3.2 The Per-Packet Overhead in Half-Duplex Switched Links ... 104
5.3.3 The Interrupt Time in Cascaded Networks ... 106
5.3.4 The Interrupt Time in Half-Duplex Switched Links ... 108
5.4 The Impact of the 802.5 Frame Format on the Performance Parameters ... 109
5.5 Summary ... 110

6. Deterministic Service Guarantees in 802.12 Networks 113
6.1 Packet Scheduling ... 113
6.1.1 Design Decisions and Constraints ... 113
6.1.2 Traffic Characterisation ... 115
6.1.3 Packet Scheduling Process ... 116
6.2 Admission Control ... 119
6.2.1 Bandwidth Test ... 119
6.2.2 Delay Bound Test ... 121
6.2.3 End-to-End Delay Characteristics ... 124
6.2.4 Buffer Space Requirements ... 127
6.2.5 Resource Partitioning ... 129
6.3 A Time Window Algorithm for the Packet Count Estimation ... 130
6.3.1 The Estimation Process ... 130
6.3.2 Admission Control and Service Issues ... 132
6.4 Implementation Issues ... 134
6.4.1 Signalling and Resource Management ... 134
6.4.2 Packet Classifier and Rate Regulator ... 135
6.4.3 Timer Issues ... 135
6.5 Performance Evaluation ... 136
6.5.1 Throughput ... 136
6.5.2 Delay Characteristics ... 140
6.5.3 Results for the Time Window Algorithm ... 146
6.5.4 Resource Utilization ... 150
6.5.5 Performance Parameters ... 153
6.6 Related Work ... 156
6.7 Summary ... 158

7. An Approximation of the Controlled Load Service 161
7.1 The Packet Scheduling Process ... 163
7.2 Admission Control ... 165
7.2.1 Bandwidth Test ... 166
7.2.2 Deriving the Output Traffic Constraint Function ... 168
7.2.3 Buffer Space Test ... 187
7.3 Performance Evaluation ... 188
7.3.1 The Impact of the Traffic Characteristics on the Buffer Space Requirements ... 188
7.3.2 The Admissible Region ... 194
7.3.3 Delay and Loss Characteristics in the 1L1S Test Network ... 197
7.3.4 Delay and Loss Characteristics in the 1HDL Test Network ... 203
7.3.5 Delay and Loss Characteristics in the 4HDL Test Network ... 207
7.3.6 Resource Utilization ... 213
7.4 Related Work ... 216
7.5 Summary ... 218

8. Summary and Future Work 221
8.1 Thesis Summary ... 221
8.2 Areas for Future Work ... 224

Bibliography 227

Peter Kim, September 1998

List of Figures

Figure 3.1: Maximum Data Rate monitored without a Packet Loss in Dependence of the Packet Size used for the Data Transmission. 42
Figure 3.2: Difference of the Interpacket Arrival Times between sent and measured Audio and Video Data Traces. 45
Figure 3.3: Difference of the Interpacket Arrival Times between sent and measured Pareto Test Traces. 46
Figure 3.4: Delay Distribution (Density) for Curve 3 (*POO: peak/average = 10*) in Figure 3.3. 46
Figure 3.5: Control Message Sequence for Measuring the Network Throughput. 48
Figure 3.6: Traffic Generator Performance on a HP C100 / 100 MHz. 49
Figure 3.7: Traffic Generator Performance on a HP 725 / 75 MHz. 49
Figure 3.8: Setup for Measuring End-to-End Delay in a shared Network. 50

Figure 4.1: Cascaded 802.12 Network Topologies. 56
Figure 4.2: The Rate Characteristics of the Application Test Traces. 57
Figure 4.3: Peak to Average Bandwidth Ratio's for the Application Traces in Table 4.1. 59
Figure 4.4: Variance-Time Plot for Application Traces (a). 61
Figure 4.5: Variance-Time Plot for Application Traces (b). 61
Figure 4.6: Variance-Time Plot for Application Traces (c). 61
Figure 4.7: Peak to Average Bandwidth Ratio for the Pareto Sources in Table 4.2. 64
Figure 4.8: Rate Characteristics of two Test Flows generated according to the Pareto Source Models POO1 and POO3. 65
Figure 4.9: Measured Worst-Case Throughput in Cascaded 802.12 Networks using a UTP Physical Layer. 66
Figure 4.10: Measured Worst-Case Throughput for a half-duplex switched Link using a UTP Physical Layer. 68
Figure 4.11: Maximum Packet Delay for different Flow Types in Dependence of the Network Load. 70
Figure 4.12: Average Packet Delay for different Flow Types in Dependence of the Network Load. 70
Figure 4.13: Packet Loss Rate for different Flow Types in Dependence of the Network Load. 70
Figure 4.14: The Packet Loss Rate for different Sets of MMC2 Flows in Dependence of the Buffer Space in the Switch. 72
Figure 4.15: The Packet Loss Rate for the different Sets of POO3 Flows in Dependence of the Buffer Space in the Switch. 74
Figure 4.16: The Impact of the Buffer Space in *Switch 1* on the Maximum End-to-End Packet Delay. 74
Figure 4.17: The Impact of the Buffer Space in *Switch 1* on the Average End-to-End Packet Delay. 74

Figure 5.1: Worst-Case Signalling on a Single Hub Network using a UTP Physical Layer. ... 82
Figure 5.2: Worst-Case Signalling on a Level-2 Cascaded Network using a UTP Physical Layer. ... 86
Figure 5.3: Worst-Case Signalling on a Half-Duplex Switched UTP Link. ... 88
Figure 5.4: The Model for Computing the Worst-Case Interrupt Time in a Single Hub Network using a UTP Physical Layer. ... 91
Figure 5.5: The Model for Computing the Worst-Case Interrupt Time in a Level-2 Cascaded Network using a UTP Physical Layer. ... 94
Figure 5.6: Measured Interrupt Times in Cascaded Networks using a UTP Physical Layer. ... 99
Figure 5.7: The Model for Computing the Worst-Case Interrupt Time on a Half-Duplex Link using a UTP Physical Layer. ... 100
Figure 5.8: Measured Interrupt Time on a Half-Duplex Switched UTP Link. ... 101
Figure 5.9: End-to-End Delay in a Setup with *Switch 2* operating in RMAC Mode. ... 102
Figure 5.10: Worst-Case Signalling on a Fibre-Optic Half-Duplex Switched Link for $l \geq L$. 104
Figure 5.11: The Model for Computing the Worst-Case Interrupt Time in a Level-2 Cascaded Network using a Fibre-Optic Physical Layer. ... 107

Figure 6.1: The Packet Scheduling Process in a Single Network Segment. ... 116
Figure 6.2: Packet Forwarding in a Network with Rate Controlled Servers. ... 125
Figure 6.3: The Measurement Process for Flow *i* ... 130
Figure 6.4: Comparison: Measured Throughput and Computed Allocation Limit in a Single Hub 802.12 Network using 100 m UTP Cabling. ... 136
Figure 6.5: Comparison: Measured Throughput and Computed Allocation Limit in a Single Hub 802.12 Network using 200 m UTP Cabling. ... 137
Figure 6.6: Comparison: Measured Throughput and Computed Allocation Limit in a Level-2 Cascaded 802.12 Network using 100 m UTP Cabling. ... 138
Figure 6.7: Comparison: Measured Throughput and Computed Allocation Limit in a Level-3 Cascaded 802.12 Network using 100 m UTP Cabling. ... 138
Figure 6.8: Comparison: Measured Throughput and Computed Allocation Limit for a 100 m UTP Half-Duplex Switched 802.12 Link. ... 139
Figure 6.9: Resource Allocation Limit in a Level-2 Cascaded Network for a High Priority Utilization Factor of: $f = 0.6$. ... 140
Figure 6.10: End-to-End Delay using the High Priority Service in a Setup with several High Priority Traffic Clients. ... 141
Figure 6.11: The Delay Distribution (Density) for the Results of Test 7 in Table 6.2 ... 144
Figure 6.12: The Delay Distribution (Density) for the Results of Test 8 in Table 6.2 ... 144
Figure 6.13: The Delay Distribution (Density) for the Results of Test 9 in Table 6.2 ... 144
Figure 6.14: The Distribution Function for the Results of Test 7, Test 8, Test 9 in Table 6.2 .. 145
Figure 6.15: The Impact of the Interrupt Time on the Average Delay in Test 7 in Table 6.2 ... 146
Figure 6.16: Data Rate generated by *vic* during the Packet Count Estimation. ... 148
Figure 6.17: Packet Count Estimation Process for *vic* ... 148
Figure 6.18: The Packet Count Estimation Process for the Applications: *MMC*, *nv*, *vat* and *OptiVision*. ... 149
Figure 6.19: Impact of the Time Frame on the Allocation Limit in a Single Hub Network using 100 m UTP Cabling ... 154

Figure 6.20: Impact of the Time Frame on the Allocation Limit in a Level-2 Cascaded Network using 100 m UTP Cabling. 154
Figure 6.21: Impact of the Timer Granularity on the Allocation Limit (*TF* = 20 ms) in a Single Hub Network using 100 m UTP Cabling. 155
Figure 6.22: Impact of the Timer Granularity on the Allocation Limit (*TF* = 10 ms) in a Single Hub Network using 100 m UTP Cabling. 155

Figure 7.1: Traffic Reshaping Points for the Controlled Load Service. 164
Figure 7.2: Network Model for Computing the Traffic Constraint Function of Flow *i*. 169
Figure 7.3: Timing Constraints for the Proof of Theorem 7.2. 172
Figure 7.4: Example Data Arrival and Departure Function for FLOW 1, FLOW 2 and FLOW 3. 178
Figure 7.5: Data Arrival and Departure Function to compute Parameter *X* when: $C_s - R_{MIN\_N1} \leq r^3$ 180
Figure 7.6: Data Arrival and Departure Function to compute the Time Interval *H*, where: $H > X$ 183
Figure 7.7: Buffer Space in Dependence of the Number of Cross Traffic Nodes and their Burst Sizes. 189
Figure 7.8: Buffer Space of FLOW 1 on a Half-Duplex Switched Link in Dependence of the Data Rate and the Cross Traffic Burst Size. 191
Figure 7.9: Example Network Topology for Results in Table 7.1 and Table 7.2. 192
Figure 7.10: Two Examples for the Admissible Region on a single Half-Duplex Switched Link. 195
Figure 7.11: Measurement Setup in the 1L1S Test Network. 199
Figure 7.12: Distribution Function for Tests 2d - 2f (MMC1) and 5d - 5f (POO3) in the 1L1S Test Network. 202
Figure 7.13: Distribution Density corresponding to Test 2f (MMC1, 1L1S Topology) in Figure 7.12. 202
Figure 7.14: Distribution Density corresponding to Test 5f (POO3, 1L1S Topology) in Figure 7.12. 202
Figure 7.15: Measurement Setup for the Half-Duplex Switched Link. 203
Figure 7.16: Distribution Function for Tests 2d - 2f (MMC1) and 5d - 5f (POO3) in the 1HDL Test Network. 206
Figure 7.17: Distribution Density corresponding to Test 2f (MMC1, 1HDL Topology) in Figure 7.16. 206
Figure 7.18: Distribution Density corresponding to Test 5f (POO3, 1HDL Topology) in Figure 7.16. 206
Figure 7.19: Measurement Setup in the 4HDL Test Network. 208
Figure 7.20: Distribution Function for Tests 2d - 2f (MMC1) and 5d - 5f (POO3) in the 4HDL Test Network. 211
Figure 7.21: Distribution Density corresponding to Test 2f (MMC1, 4HDL Topology) in Figure 7.20. 211
Figure 7.22: Distribution Density corresponding to Test 5f (POO3, 4HDL Topology) in Figure 7.20. 211

Peter Kim, September 1998

List of Tables

Table 2.1: Application Classes and Example Applications [Garr96]. 14

Table 3.1: MIB Counters used for Throughput Measurements. 48

Table 4.1: Basic Application Trace Characteristics. 59
Table 4.2: Pareto Source Characteristics. 63

Table 5.3: Breakdown of the Grant-Signalling Delay for a UTP Physical Layer. 84
Table 5.4: Breakdown of the Data Transmission Delay for a UTP Physical Layer. 84
Table 5.5: Per-Packet Overhead for Cascaded Networks using a UTP Physical Layer. 88
Table 5.6: Per-Packet Overhead for Half-Duplex Switched UTP Links. 90
Table 5.7: Breakdown of the Delay required for Signalling the Control Signals Req_H, Req_N and Incoming across a single UTP Link. 92
Table 5.8: Normal Priority Service Interrupt Times in Cascaded Networks using UTP Cabling. 98
Table 5.9: Normal Priority Service Interrupt Times on Half-Duplex Switched UTP Links. 101
Table 5.10: Breakdown of the Data Transmission Delay for a Fibre-Optic Physical Layer. .. 103
Table 5.11: Per-Packet Overhead for Fibre-Optic Cascaded Networks. 104
Table 5.12: Breakdown of the Grant-Signalling Delay for a Fibre-Optic Physical Layer. 106
Table 5.13: Per-Packet Overhead for Fibre-Optic Half-Duplex Switched Links. 106
Table 5.14: Breakdown of the Delay required for Signalling the Control Signals Req_H, Req_N and Incoming across a single Fibre-Optic Link. 108
Table 5.15: Normal Priority Service Interrupt Times in Fibre-Optic Cascaded Networks. 108
Table 5.16: Normal Priority Service Interrupt Times on a Fibre-Optic Half-Duplex Switched Link. 109

Table 6.1: Source and Token Bucket Parameters for the Delay Tests in a Level-2 Cascaded Network. 142
Table 6.2: Comparison: Computed and Measured Delay in a Level-2 Cascaded 802.12 Network. 143
Table 6.3: Parameters of the Time Window Algorithm used for the Packet Count Estimation. 147
Table 6.4: Maximum High Priority Network Utilization in a Single Hub Network. 151
Table 6.5: Maximum High Priority Network Utilization in a Level-2 Cascaded Network. 152
Table 6.6: Maximum High Priority Network Utilization for different Time Frames in a single Hub Network. 153

Table 7.1: Buffer Space Requirements for FLOW 1 in Dependence of the Cross Traffic reserved along the Data Path. 193
Table 7.2: Buffer Space Requirements of FLOW 1 and FLOW 2 for the Setup in Table 7.1. 193
Table 7.3: Source and Token Bucket Parameters for the Application Traces *MMC1*, *MMC2* and *OVision*. 197
Table 7.4: Source and Token Bucket Parameters for the Pareto Sources. 198
Table 7.5: Measured Packet Delay and Loss Rate for the Application Traces: *MMC1*, *MMC2* and *OVision* in the Level-1 Cascaded Test Network. 200
Table 7.6: Measured Packet Delay and Loss Rate for the Pareto Sources in the Level-1 Cascaded Test Network. 200
Table 7.7: Measured Packet Delay and Loss Rate for the Application Traces: *MMC1*, *MMC2* and *OVision*, across 2 LAN Switches interconnected by a single Half-Duplex Switched Link. 204
Table 7.8: Measured Packet Delay and Loss Rate for the Pareto Sources across 2 LAN Switches interconnected by a single Half-Duplex Switched Link.204
Table 7.9: Measured Packet Delay and Loss Rate for the Video Sources across 5 LAN Switches and 4 Half-Duplex Switched Links. 209
Table 7.10: Measured Packet Delay and Loss Rate for the Pareto Sources across 5 LAN Switches and 4 Half-Duplex Switched Links. 209
Table 7.11: Maximum High Priority Utilization using the Controlled Load Service in a Single Hub Network..... 214
Table 7.12: Maximum High Priority Utilization using the Controlled Load Service in a Level-2 Cascaded Network. 214

Mathematical Notation

Note that this notation only includes the parameters used in our analysis but not those used to refer to related work.

$b_{in}^{I}(t)$: Input Traffic Constraint Function of *FLOW I* ($I \in 1, 2, 3$) at the entrance of a segment. FLOW *I* may describe an aggregation of flows *i*, where $i \in N$.

$b_{k}^{i}(t)$: Traffic Constraint Function for flow *i* on network node *k*.

$b_{out}^{I}(t)$: Output Traffic Constraint Function of *FLOW I* ($I \in 1, 2, 3$) at the exit of a segment. FLOW *I* may describe an aggregation of flows *i*, where $i \in N$.

C_l: link speed not including the data transmission overhead ($C_l = 100$ Mbit/s for 802.12 networks).

C_s: service rate (data throughput for a particular time frame *TF* and a particular set of Packet Counts $pcnt_k^i$).

C_{tot}: total rate dependent error term defined in the Guaranteed service specification.

dE_k: External Packet Transmission Delay of network node *k*.

$dE_{k,j}$: External Packet Transmission Delay imposed on node *k* by packet transmissions from node *j* on the same network segment.

dL_k: Local Packet Transmission Delay of network node *k*.

$dN^{i}_{End\text{-}ToEnd}$: end-to-end delay bound for flow *i* across a bridged network.

dO_k^i: constant overhead delay introduced for flow *i* on node *k*.

dR_k^i: maximum delay of flow *i* in the corresponding rate regulator on network node *k*.

dS_k: maximum queueing and propagation delay for all real-time data packets from node *k* on a single segment.

D_{it}: Normal Priority Service Interrupt Time (general).

D_{it_LN}: Normal Priority Service Interrupt Time in a *Level-N* cascaded 802.12 network.

D_{it_HD}: Normal Priority Service Interrupt Time for a half-duplex switched link.

D_{Incom}: maximum time to signal *Incoming* across a single link (see Chapter 5).

D_{pp}: Per-Packet Overhead (general).

D_{pp_LN}: Per-Packet Overhead in a *Level-N* cascaded 802.12 network.

D_{pp_LN}: Per-Packet Overhead for a half-duplex switched link.

D_{Req_H}: maximum time to signal *Req_H* across a single link (see Chapter 5).

D_{RMAC_Data}: maximum delay encountered by data packets in the 802.12 hub (see Chapter 5).

D_{Signal_Ctrl}: maximum time to signal a Demand Priority control signal (*Incoming*, *Req_H*, *Req_L*, *ENA_HO*) across a single link (see Chapter 5).

D_{Signal_Grant}: maximum time to signal *Grant* across a single link (see Chapter 5).

D_{tot}: total rate independent error term defined in the Guaranteed service specification.

D_{Tx_Data}: maximum time to transmit a data packet across a single link (see Chapter 5).

D_IPG: 802.12 timer accounting for clock differences between different hubs in the shared network (DELTA_IPG_WINDOW).

f: High Priority Utilization Factor ($0 \le f \le 1$).

H: time interval in which: $R^1_{out}(t) = 0, R^2_{out}(t) > 0, R^3_{out}(t) > 0$ (see Chapter 7).

I_BST: 802.12 Idle Burst Timer interval (SEND_IDLE_BURST).

IPG: 802.12 Inter-Packet Gap (IPG_WINDOW).

l: cable length of a single link.

LTT: minimum normal priority data transmission time ($D_{it} \le LTT \le TF$).

m: number of nodes with reservations on the network segment.

MAX_PCNT^i: worst-case Packet Count for flow *i* (Time Window Algorithm).

n: number of flows on a particular node in the network.

N: cascading level to classify multi-hub network topologies (Chapter 5).

r^i: token generation rate of flow *i* (part of the (δ^i, r^i) traffic characterisation).

r^i_{alloc}: allocated data rate for flow *i* (Time Window Algorithm).

r^i_{TW}: measured data rate for flow *i* over the time interval *TW* (Time Window Algorithm).

p: packet size ($P_{min} \le p \le P_{max}$).

$pcnt^i_k$: Packet Count (maximum number of packets allowed per time frame *TF*) of flow *i* on network node *k*.

$PCNT_k$: Packet Count of network node *k* ($PCNT_k = \sum_{i \in n} pcnt^i_k$).

P_{max}: maximum link packet size.

P_{min}: minimum link packet size.

$P_{MIN\_AVE\_Nk}$: Minimum Average Packet Size of all real-time data packets sent by network node *k* averaged over the time frame *TF*.

$P_{MIN\_AVE\_S}$: Minimum Average Packet Size on segment *S* over the time frame *TF* ($P_{MIN\_AVE\_S} = \sum_{k \in m} P_{MIN\_AVE\_Nk}$).

$R^I_{in}(t)$: Rate Function of *FLOW I* ($I \in 1, 2, 3$) at the entrance to a segment (Input Rate Function). FLOW *I* may describe an aggregation of flows *i*, where $i \in N$.

$R^i_{min\_Nk}$: minimum service rate of flow *i* on network node *k*.

R_{MIN_Nk}: minimum service rate of node *k* ($0 < R_{MIN\_Nk} \leq C_s$, $R_{MIN\_Nk} = \sum_{i=1}^{n} R^i_{min\_Nk}$).

$R^I_{out}(t)$: Rate Function of *FLOW I* ($I \in 1, 2, 3$) at the exit of a segment (Output Rate Function). FLOW *I* may describe an aggregation of flows *i*, where $i \in N$.

$scnt^i$: number of packets received from flow *i* within the current time frame *TF* (Time Window Algorithm).

$scnt^i_{TW}$: maximum value observed for $scnt^i$ within the current time window *TW*.

sQ^i_k: buffer space (upper bound) required for flow *i* in the output queue at node *k*.

sR^i_k: buffer space (upper bound) required for the rate regulator of flow *i* at switch *k*.

sS^i_k: total buffer space (upper bound) required for flow *i* at switch *k*.

T_k: timer granularity of all rate regulators on node *k*.

TF: resource allocation time frame.

TW: time window used in the Time Window Algorithm.

wm^i: high watermark for flow *i* (Time Window Algorithm).

Z: time interval in which: $R^1_{out}(t) = R^1_{min\_N1}$, $R^2_{out}(t) > 0$, $R^3_{out}(t) > 0$ (see Chapter 7).

α^i: parameter determining the conservativeness of the Packet Count estimation for flow *i* (Time Window Algorithm).

β^i: uncertainty parameter for flow *i* (Time Window Algorithm).

Δ: time interval, where: $\Delta = (m-1) \cdot P_{max} / C_s + H + Z$ (see Chapter 7).

δ^i: token bucket depth (burst size) of flow *i* (part of the (δ^i, r^i) traffic characterisation).

κ: packet count update factor (Time Window Algorithm).

Peter Kim, September 1998

Chapter 1
Introduction

The use of applications with a variety of performance constraints and the widening commercial use of the Internet are driving its migration to an Integrated Services Packet Network (ISPN) [CSZ92], [BCS94], [WhCr97]. In contrast to the current Internet, which only provides the traditional best-effort service, the new architecture will additionally offer advanced services called Integrated Services. The differentiator of these new services is the Quality of Service (QoS) and the diverse service commitments e.g. probabilistic or deterministic performance guarantees which are assured by the network. Quality of service will be required for supporting applications with stringent performance constraints like Internet telephony, video conferencing, or distributed virtual reality over the Internet, but will also be useful for ensuring a minimum bandwidth for traditional data transfers over congested links.

In this chapter, we introduce the research area and specify the problem to which this thesis is dedicated. Section 1.1 motivates the need for an Integrated Services network and discusses the two traditional network approaches that could be used to achieve this. We believe that future Integrated Services networks will be based on the packet switching approach because of its ability to support resource sharing and statistical multiplexing. Packet switching and resource sharing however also cause network congestion. We discuss the different concepts to control the congestion in packet switched networks and argue that a proactive scheme is required to support deterministic service guarantees. Section 1.2 contains the problem specification. We first outline the framework in which our research was performed. This is the ISPN architecture that has been proposed by the Internet Engineering Task Force[1] (IETF) for a future multi-services Internet. We then describe the hypothesis and motivate our work. It follows a list with the contributions made by this thesis and a description of the research methodology which we adopted to achieve these results. Section 1.3 finishes the chapter with an overview of the thesis.

1.1 Background

1.1.1 Motivation for an Integrated Services Network

Large traditional networks like the telephone network, the Internet or the cable TV network have been mainly designed to offer a single specific service. The phone network is specialized to carry interactive voice. For this, it provides a full-duplex, ordered, low delay, low jitter and fixed band-

1. See http://www.ietf.cnri.reston.va.us/

width service based on a circuit switched network [Tane89 - Chapter 2]. In contrast, data networks have been designed to carry digital data between computers. A large existing data network is the Internet which consists of a multitude of autonomous networks connected in a world wide hierarchy. Data is carried in containers called packets or datagrams. Switching nodes within the network use a store and forward mechanism to transfer data packets to their destination. This is called packet switching. The service offered by the network is a simple, unreliable packet delivery service. Service guarantees in respect to throughput, delay or an ordered packet delivery are not provided. Finally, the cable TV network was designed to carry high bit rate video. It offers a simplex, ordered, high bandwidth and low jitter service. A low end-to-end delay between the source and the destinations is not required because the network is only used for one way video broadcasts without time sensitive receiver interactions.

Offering these services and other services in a single communication network could lead to a multitude of advantages which include the economy, the flexibility, the connectivity, and the way resources can be accessed in future networks [CSZ92]. Lower costs can be achieved by using a single information infrastructure which promotes resource sharing and statistical multiplexing. A user only connects to a single network, but can reach millions of other users using various types of media e.g. electronic mail, voice or video, and has access to information in world wide distributed data bases. Furthermore, being able to support a multitude of existing and future applications with different performance constraints increases the flexibility of the network and ensures growth.

1.1.2 Circuit Switching versus Packet Switching

There has been much discussion about whether the new infrastructure should be based on a circuit switching or packet switching approach. The traditional circuit switching approach as used in the telephone network is based on circuits and a connection setup. A circuit is normally a fixed data path with a fixed bandwidth between the source and the destination. The connection setup is used to pre-allocate a circuit and the corresponding resources along the data path in the network. This is carried out before the actual communication. Once the circuit is established, data can be transmitted simultaneously in both directions between the data source and the destination. Network resources are released when a user hangs up. The main advantage of this approach is the quality of service which is guaranteed and allocated for the lifetime of the connection. This has high costs because allocated but unused resources are not available for other users in the network.

In contrast to this, existing packet switching networks use a more dynamic allocation strategy. Resources in switching nodes such as for example buffer space are occupied when a data packet enters the switch and become released immediately after the packet was forwarded to the next switch in the data path. Switching nodes in traditional packet switched networks further do not maintain per-connection state and thus do not need a connection setup to install these informations. In the Internet, data packets are transmitted to their destination based on three fundamental concepts [Kesh97 - Chapter 3]: Addressing, Routing, and the Internet Protocol (IP). Addressing is the mech-

anism to identify each node in the network by using a unique identifier. Routing determines the path taken by data packets through the meshed network. This is based on address information additionally carried within each data packet. The Internet Protocol [Post81a] provides a standardized way of interpreting the addresses and the control informations in data packets across different link technologies. Network layer switching nodes selecting the data path are called *Routers*. Each data packet is routed independently through the network. Different packets may thus follow a different route through the meshed network and can arrive at the destination out of order. For a comprehensive discussion of addressing and routing issues, see [Perl92 - Chapter 6, Chapter 9].

The main advantage of the packet switching approach is its flexibility and its ability to support statistical resource sharing. Flexibility is given because network resources are consumed based on current availability and do not have to be pre-allocated. A single data source could thus potentially, if there were no other active sources, take advantage of the entire network performance while e.g. transferring a data file to the desired destination.

The traffic in data networks is bursty and unpredictable [PaFl95], [WTSW95]. Resource sharing works well because it is statistically not likely that all network sources are active at the same time and send data at peak rate. This is because they are typically independent of each other. The network may therefore be oversubscribed according to the call and traffic characteristics of the data sources connected. Bursty traffic and resource sharing however also potentially cause network overload, long packet delays and packet loss in the network. This is discussed in the next section. To minimize or prevent overload, the traffic passed into the network needs to be regulated. This is hard to do in such a way that network resources are efficiently used, but overload is avoided because data sources do not know the end-to-end network capacity and the cross traffic characteristics along the data path. Another problem is the quality of service. Service guarantees in traditional stateless packet switched networks are hard to quantify due to resource sharing and bursty traffic characteristics.

It however seems that Integrated Services networks will be based on the packet switching concept because of its flexibility and the potentially higher resource utilization that can be achieved by exploiting resource sharing and statistical multiplexing. One example is the Internet which currently evolves from a simple data network into a multi-service network [WhCr97].

Another approach taken forward by the International Telecommunications Unit - Telecommunications Standardizations Sector[1] (ITU-T) and the ATM Forum is the Asynchronous Transfer Mode (ATM) in the context of the Broadband Integrated Services Digital Network (B-ISDN) [Minz89], [DeTr97]. ATM is based on five important concepts [Kesh97 - Chapter 4]: (1) virtual circuits, (2) fixed sized packets and packet switching, (3) small packets, (4) statistical multiplexing, and (5) Integrated Services. These enable ATM to offer flexibility, scalability, high bandwidth and quality of service guarantees. On the basis of these properties, ATM is often seen as the one-for-all purpose technology which might become the core of the future Internet and of the telephone network.

1. See http://www.itu.ch/

In Local Area Networks (LANs) however, it seems that in the near future, ATM will not be able to play a major role. This is due to the cost effective solutions which are available for other high speed LAN technologies like Ethernet, FDDI, Token Ring or 802.12 Demand Priority. In contrast to these technologies, ATM is currently still expensive. Trends like the development of Gigabit Ethernet, port trunking and the migration to switched networks will further ensure that high bandwidth demands can be satisfied at competitive costs. Finally, both, the Institute of Electrical and Electronics Engineers[1] (IEEE) and the IETF are currently standardizing the mechanisms required to support QoS across IEEE 802 local area networks. This might make these technologies even more popular.

1.1.3 Congestion in Packet Switched Networks

Network overload, long delays and packet loss appear when the aggregate input rate into the network, or in a single part of the network (the bottleneck), exceeds the service and buffer capacity of the network. This is called congestion [Jaco88]. One might view congestion as the price for the flexibility and efficiency gained by exploiting statistical multiplexing.

Congestion usually occurs at switches in the network. It is a high load phenomenon [Kesh92]. In times of overload, switches first try to queue any data packet which they can not forward instantly. This may cause long packet delays especially on slow speed links. If the buffer capacity is exceeded then any incoming packet is dropped. To nevertheless ensure a reliable data transmission, end-to-end error discovery and recovery mechanisms are used. These could for example be based on sequence numbers, positive acknowledgements and timeout driven packet retransmissions as applied in the Transmission Control Protocol (TCP) [Post81b], [Stev94].

Sustained congestion, if not appropriately controlled, may lead to a substantial loss in performance and quality of service [Jaco88], [Tane89 - Chapter 5]. To prevent congestion in the network, Congestion Control is applied. In general, it has three objectives: (1) to prevent overload and packet loss in the network, (2) the efficient use of network resources, and (3) to ensure that the available network resources are shared in a fair way amongst all individual users. This is hard to achieve because (1) a high network utilization also increases the risk of overload, and (2) malicious users might try to increase their fair share by aggressively sending data into the network. The control is further complicated because it is usually a global network issue and thus often involves the participation of all data sources in the network.

The mechanisms for congestion control can be classified in *reactive* and *proactive* schemes. Reactive approaches are based on control mechanisms within hosts and on feedback from the network. By monitoring the state of the network, data sources try to detect symptoms of network congestion. Switches either provide: (1) *explicit* feedback e.g. by setting a congestion indication bit [RaJa90] or sending Source Quench messages [PrPo87], or (2) *implicit* feedback by dropping data packets [Jaco88]. After receiving the feedback, the data sources then adjust their transmission rate.

1. See http://www.ieee.org/

In contrast, proactive congestion control schemes prevent overload by reserving resources within the network. However, to design a multi services network, a hybrid scheme seems to be the most attractive approach. By using a different control scheme for different services, the advantages of the reactive and the proactive congestion control can be exploited. All three approaches are outlined in the following.

1.1.4 Reactive Congestion Control

Much research has been done in the past on reactive (or feedback based) congestion control algorithms [PrPo87], [Jaco88], [Jain89], [RaJa90]. The scheme that has become the standard for TCP congestion control was devised by Van Jacobson [Jaco88], [Brad89] and verified in [SZC90], [ZSC91]. It uses a timeout mechanism to detect network congestion. The scheme deployed today consist of a set of algorithms [Stev94 - Chapter 20, Chapter 21]: (1) the Slow Start algorithm, and (2) the Fast Retransmit and the Fast Recovery algorithms. Due to its importance in the vastly growing Internet, these are briefly outlined in the following.

The Slow Start algorithm is used at the beginning of the data transmission or after a timeout. The algorithm probes the available network capacity by gradually increasing the amount of data in transit. To achieve a fast adaptation rate, Slow Start first uses an exponential increase and, after reaching the Slow Start threshold, continues linearly to avoid congestion. The Fast Retransmit algorithm allows to recover from a packet loss without having to wait for the timeout. After a Fast Retransmit, Fast Recovery allows a data source to quickly reopen the congestion window. Both mechanisms rely on counting the number of duplicate acknowledgments which are sent by a TCP receiver in response to data packets received after a packet has gone missing in the network.

Since 1988, various proposals have been made to improve the performance of Jacobson's algorithm. These are often based on exploiting additional symptoms of network congestion. The scheme proposed in [WaCr92], takes advantage of changes in the Round Trip Time (RTT). It uses the fact that the queueing delay in switches, and thus the RTT, increases significantly when the network becomes overloaded. The algorithms in [WaCr91], [BOP94] are based on measuring the throughput, which typically decreases as the network reaches congestion. The authors of [MaMa96] proposed a forward acknowledgement congestion control algorithm to be used with the TCP SACK option [MMFR96]. In [SMM98], performance improvements are achieved by dynamically adjusting the socket buffers for each connection. Furthermore, research on how gateways should drop data packets such that fairness and throughput are maintained has also been pursued [Mank90], [FlJa93], [LiMo97]. All these mechanisms however do not enable the network to provide stringent service guarantees.

TCP uses a *window-based* flow control scheme [Stev94 - Chapter 20]. The receiver controls the number of data packets that the source may send. An alternative approach is to use a *rate-based* algorithm such as for example employed in the Xpress Transfer Protocol (XTP) [SDW92]. In a rate-based flow control scheme, the receiver specifies the data rate that the source is allowed to send.

Existing flow control schemes are typically assigned to one of these two classes. For a discussion see [MaZa90]. In respect to resource reservation, a rate-based control mechanism seems to be more appropriate since resources are typically allocated based on the bandwidth and the delay requirements of the application. Some data sources such as audio or video are further self rate-limiting which fits well into a rate based scheme.

Even though end-to-end reactive congestion control schemes are able to efficiently control the overall network load, there are several reasons why they are not suitable for providing hard service guarantees. First, a large *Bandwidth-Delay* product may lead to quality degradation and congestion [Kesh92]. This is caused by the attempt of data sources to fully utilize the network resources based on a Round Trip Time (RTT) estimation. The Bandwidth-Delay product describes the maximum amount of data, a data source has in transit in the network. It is computed by multiplying the link bandwidth with the RTT, where the RTT denotes the transmission time for a data packet from the data source to the receiver plus the time it takes to transmit the acknowledgement back to the source.

In networks with higher capacity, the bandwidth increases and the RTT decreases. The RTT is however always bounded from below by the data propagation delay in the network. As the link speed increases, the Bandwidth-Delay product will thus also grow. The potential problem is caused by the fact that congestion control can only be enforced across time scales in the order of one RTT, because this is the minimum time that is needed for a reactive data source to determine the impact of its sending rate [Kesh92]. In the event that the network's service rate suddenly drops, it thus takes at least RTT time units before a data source can lower its transmission rate. Data packets equivalent to the Bandwidth-Delay product are however already in the network and may cause unfairness or congestion since they cannot be controlled any more.

Once packet loss occurred, data is retransmitted. Since it takes at least one RTT to detect the packet loss, retransmitted data can reach the receiver only after about 1.5 RTT units. For delay sensitive applications e.g. Internet telephony, the retransmitted information might however already be outdated due to real time constraints and can thus not be used any more. Correlating traffic bursts may further always lead to a degradation in the service quality. For applications which require stringent performance guarantees, a proactive congestion control approach is thus required [Zhan93].

1.1.5 Proactive Congestion Control

Proactive congestion control is based on reserving resources such as bandwidth or buffer space within the network. A reservation may belong to a single connection or to a group of connections. It may thus for example be used for all packets between two remote sites. To receive service guarantees, network resources must be reserved prior to the actual communication [Ferr90], [CSZ92]. The corresponding reservation request typically specifies: (1) the characteristics of the data traffic passed into the network, and (2) the service requested for it. This information is then distributed to all switching nodes along the data path. With information about individual connections in switches, congestion can then be accurately controlled at the place in the network where it usually occurs.

Traffic control is enforced at two different levels: (1) at the connection level through the use of Admission Control, and (2) at the packet switching level through Traffic Enforcement and the Service Discipline used in switching nodes. Admission Control is the decision about the resource availability [BCS94]. It restricts the access to the network resources and tests that: (1) the service requirements specified in the new request can be provided by the network, and (2) that the service guarantees given to already accepted service users are not violated by the admission of the new connection. If appropriate network resources are not available, the new request is rejected.

The Traffic Enforcement ensures that data sources do not use more network resources then reserved for them. For this, the network monitors the traffic of the user as it enters the network and compares it against the traffic specification received at connection setup. This operation is called *Traffic Policing*. If data sources violate their traffic specification and send more data than announced, then excess data packets can be marked [SSC97], or shaped as e.g. performed by the (δ, r) *Regulator* [Cruz91a] discussed later in Section 6.1.2 and Section 6.4.2. Marked data packets are forwarded but are at higher risk to become delayed or dropped at downstream switching nodes in the data path. The traffic shaping carried out by the (δ, r) Regulator is basically a data rate enforcement.

The service discipline is implemented in the switch's packet scheduler. It determines how data packets are processed (scheduled) and thus what service guarantees can be met. Two different resources are managed: the bandwidth of the outgoing link, and the buffer space within the switch. Guarantees for data throughput, packet delay and delay jitter are achieved by: (1) changing the packet order in which packets from different connections are forwarded, and (2) by controlling the packet departure times in switches. Both is performed on a per-packet basis. The packet loss characteristics are principally determined by the buffer management and the packet discard policies implemented in the switch.

A simple service discipline is Static Priority (SP) studied in [Cruz91a]. A static priority scheduler consists of a fixed number of prioritized First-Come-First-Served (FCFS) queues. Data packets from these queues are served according to strict priorities. Higher priority packets are always processed first. Lower priority queues are only served when all higher priority queues are empty. Connections that use the same priority level receive the same service but may interact with each other. More sophisticated schedulers can protect the QoS by isolating single connections. This is for example achieved by adding the (δ, r) Regulator, one per-connection, to the Static Priority scheduler [ZhFe93], or by using a round robin approach [Nagl87], [Hahn87], [ShVa95]. We discuss packet scheduling issues more precisely in the context of the Integrated Services Packet Network in Section 2.3 in Chapter 2.

1.1.6 Combining Reactive and Proactive Approaches in Integrated Services Networks

In a strict proactive scheme, resources are reserved for all network users. Hybrid approaches combine reactive and proactive control mechanisms. A network might for example provide the traditional Best Effort service based on reactive control but additionally services such as the Guaranteed

service described in Section 2.2.2 by using a proactive approach. Such a strategy is for example used in the Tenet Scheme [FeVe90], [FBZ92], Golestani's DLSM approach [Gole91], Zhang's Flow Network [Zhan91], or in the Integrated Service Packet Network [BCS94]. The ISPN approach will be discussed in more detail in Chapter 2.

Using a reactive congestion control ensures simplicity and flexibility. Data packets may be sent without any reservation or a connection setup. It typically however requires a cooperative environment where all data sources behave well. In contrast, proactive schemes can isolate traffic and can provide service guarantees. The drawbacks are higher costs and in general a lower resource utilization. Higher costs are caused by the resource management, and the often more advanced packet scheduler. The resource utilization can become low when resources for bursty data sources are allocated at peak rate due to deterministic service constraints.

Hybrid schemes can take advantage of both approaches. The flexibility increases since the network is able to support several network services. A key advantage is that statistical multiplexing between these services can be exploited. This could e.g. be performed according to the scheme in [FlJa95]. Any resources reserved but unused by the user can instantly be used for services with a lower service commitment e.g. the best effort service. This allows a high network utilization and thus lower costs even when the resources for the Guaranteed service become allocated at peak rate. To further improve the statistical multiplexing gain, hybrid schemes might additionally provide network services with statistical guarantees as offered in [FeVe90] and [Gole91]. It remains to remark that a hybrid solution simplifies the migration to a multi-service Internet because the existing best effort service is maintained.

1.2 Problem Specification

Embedding a proactive control scheme into an existing data network which only provides the best effort service is hard. For the Internet, this requires a significant change of the packet forwarding mechanisms currently deployed. Even though the basic IP service will still be supported, new mechanisms and components need to be deployed at almost all layers of the data transport system.

1.2.1 Framework

The ISPN architecture [BCS94] describes the extensions required to provide Integrated Services across the Internet. This architecture was used as basic framework for our research. A core component of the ISPN is the extended service model because this defines the visible end-to-end behaviour of the network. So far, the Guaranteed- [SPG97] and the Controlled Load [Wro97a] service have been put forward as proposed standards. Both services require admission control and the reservation of resources in the network.

End-to-end service guarantees can only be provided when the service is maintained at all intermediate links along the data path in the network. If a single element does not support the service requirements, then stringent guarantees cannot be given. The resulting quality of service can nevertheless

be acceptable for the user if sufficient (best-effort) resources are available in the reservationless part of the data path. LAN technology is typically located at both ends of this data path, or in Intranets, where large bridged networks often interconnect many users. There is however no standard mechanism for providing service guarantees across existing LANs such as 802.3 Ethernet, 802.5 Token Ring, or 802.12 Demand Priority. This is because each technology has a different medium access mechanism and therefore schedules data packets according to its own policy. Shared LANs can thus be viewed as having a built-in link layer service discipline. Another factor to be considered is the bridged LAN topology which typically includes shared, half-duplex- or full-duplex switched links. On half-duplex switched links for example, the medium access contention can only occur between two network nodes which may simplify the admission control. This is discussed in Section 4.1. The service discipline and the admission control conditions used to enforce service guarantees will thus typically be technology specific, sometimes even topology dependent, and must be defined separately for each LAN technology. This significantly differs from a wide area network environment where routers are typically interconnected by full-duplex links.

The IETF Integrated Services over Specific Link Layers (ISSLL) working group was chartered with the purpose of exploring the mechanisms required for supporting Integrated Services over various link layer technologies. Reference [GPS+98] describes the framework for providing this functionality in shared and switched IEEE 802 type LANs. Our work was carried out in this context.

1.2.2 Hypothesis and Research Goals

In this thesis, we prove that service guarantees, in particular the Integrated Services standardized for a future Internet, can be provided across multi-hub shared and half-duplex switched Demand Priority (IEEE 802.12) [ISO95] networks, even when the network is highly utilized or becomes overloaded with best effort traffic. This is performed in two steps: (1) the definition of the packet scheduling process and the corresponding admission control conditions, and (2) the verification of the guarantees given to service users. Two fundamental constraints can be identified: (1) the kind of service guarantee to be provided by the network, and (2) the performance of the underlying 802.12 network in various topologies.

The Guaranteed service implies a deterministic service guarantee for the maximum packet transmission delay in the network. In contrast, the Controlled Load service trades off a weaker service commitment for a higher network resource utilization. It does not provide stringent service guarantees, but guarantees the approximation of an unloaded network, even when the network is actually overloaded. Chapter 2 describes both services more precisely.

IEEE 802.12 is the standard for a shared 100 MBit/s LAN. Its Medium Access Control (MAC) protocol is called Demand Priority. Its main characteristic in respect to QoS is the support of the two priority levels: normal and high priority. A simple network consists of a single hub (repeater) and several nodes such as hosts or routers, each separately connected to the hub creating a star topology. The standard further allows multi-hub network topologies in which all hubs become connected in a

rooted tree like network structure. This is called *Cascading*. The resulting shared networks are called *Cascaded Networks*. Each hub in a multi-hub network may have many links which either connect to a lower level hub or to a network node. Cascaded topologies are thus able to incorporate hundreds of network nodes and may have a physical extension of many hundred meters. Furthermore, the network may contain bridges/switches interconnected through shared or switched 802.12 links.

Our work has several motives. First, we believe that providing service guarantees in shared packet switching data networks is an interesting problem due to the QoS constraints of the shared environment. Particularly challenging was to devise the mechanisms for providing a Guaranteed service across cascaded Demand Priority networks considering the variable data throughput which does not only depend on the network's topology but also on the size of the packets used for the data transmission. Furthermore, it still seems a wide spread belief that useful stringent delay bounds either require ATM technology to the desktop, or LANs consisting of complicated switches interconnected by full-duplex point-to-point links. This thesis shows that deterministic service guarantees in the order of a few milliseconds can be provided in shared 802.12 networks of large size and physical extension.

Secondly, IEEE 802.12 is a LAN standard. The IETF and the IEEE are currently standardizing the mechanisms required to extend multi-service architectures like the ISPN network, to shared and switched LANs. An important goal for our research was to devise a solution which does not require additionally changes to the 802.12 standard. We further aimed at a cost effective solution for both services, where possible. This resulted from the strong costs constraints in the LAN market.

1.2.3 Thesis Contributions

This thesis has three contributions. It first (1) contains a detailed performance analysis of 802.12 networks in respect to quality of service. This includes the shared single-hub network, multi-hub cascaded topologies and half-duplex switched links operating according to the Demand Priority MAC protocol. The results of the theoretical analysis enable us to accurately determine the minimum available data throughput in the network. They are thus essential to build a Guaranteed service, but can also be used as the basis in developing advanced services with a lower assurance level. During the analysis, we focus on an Unshielded Twisted Pair (UTP) physical layer since this represents the most interesting case. We however also derive the equivalent results for Fiber-Optic 802.12 networks.

The thesis further (2) defines the packet scheduling process and the corresponding admission control conditions for providing a deterministic delay bound. This is sufficient for supporting the Guaranteed service. The new service is built on top of the 802.12 high priority access mechanism. Best effort traffic is served at normal priority.

Thirdly (3), we show how the Controlled Load service could be realized in shared and switched 802.12 networks. In contrast to the Guaranteed service, this was based on simple Static Priority

packet scheduling in LAN switches which allows a straightforward deployment in existing or next generation switch products. The service specification also requires the use of admission control. In deriving the corresponding conditions, we could however utilize basic results received for the deterministic case.

It remains to remark that although our research was performed in the context of the ISPN and ISSLL framework, the results might also be used to support differential services discussed in Section 2.7 in Chapter 2. Furthermore, parts of the work reported in this thesis can also be found in [Kim96], [Kim97a], [Kim97b] and [GPS+98].

1.2.4 Research Methodology

We use the following research methodology: we first review the ISPN architecture and study the Integrated Services proposed. This defines the properties to be provided by 802.12 networks supporting these services. We then investigate the best effort service quality under overload and discuss solutions to maintain the QoS in the event of network overload. To be able to allocate resources in the network, the network's performance must be known. Our network analysis identifies two topology specific parameters whose results are sufficient to accurately perform admission control. We then define the scheduling process and the admission control conditions that are used to provide the Guaranteed- and the Controlled Load service across 802.12 networks. Both algorithms are verified and evaluated.

Our research is based on two methods: a theoretical analysis and experimental measurements. An analytical approach is chosen to analyse 802.12 specific performance parameters, and to derive admission control conditions. Measurements were performed to confirm our worst case network model and the parameters derived from it. Experimental results were further achieved for service parameters such as bandwidth, end-to-end packet delay and packet loss. This was to confirm the service guarantees given to applications, but also to compare these results with the theoretical results obtained in the analysis.

1.3 Overview of the Thesis

This thesis is organized as follows.

Chapter 2 reviews the Integrated Services Packet Network architecture and discusses LAN specific issues. For each component of the architecture, we summarize previous work related to resource reservation and quality of service. Further outlined are the fundamental trade-offs that can be made in the design of a multi-service network.

Chapter 3 introduces our measurement methodology and discusses the measurement accuracy of the solutions chosen. This starts with the trace driven approach that was applied to generate realistic traffic patterns within the test network. Afterwards, we discuss the methods used to measure the data throughput, the packet delay and the packet loss rate.

Chapter 4 looks at the network overload behaviour and studies the data throughput, packet delay and loss characteristics. We then investigate the capability of the network to buffer temporary traffic bursts. Results which show the impact of additional buffer space in switches on the packet delay and the packet loss rate are also presented.

Chapter 5 analyses IEEE 802.12 networks in respect to quality of service. This is focused on two parameters: (1) the Normal Priority Service Interrupt Time and (2) the Per-Packet Overhead, which we use later in the admission control to describe the Demand Priority signalling overhead. For both of them we derive deterministic upper bounds assuming UTP- and Fibre-Optic physical layers. This was based on worst-case performance models identified for different 802.12 network topologies.

Chapter 6 proposes a resource allocation scheme which enables Demand Priority networks to provide deterministic service guarantees in shared and bridged network topologies. First described are the overall design and the packet scheduling process. We then define and prove the admission control conditions. Afterwards we outline our implementation and evaluate the performance of the new service. This for example includes a comparison between the analytical and the experimental results received. Related work providing deterministic service guarantees within LANs is also discussed in this chapter.

Chapter 7 proposes an equivalent allocation scheme for Controlled Load type service guarantees. We first describe the packet scheduling and define the admission control conditions. We then study the performance of the service in three different network topologies, using five different types of data sources. Finally, related work for the Controlled Load service is outlined.

Chapter 8 summarizes the thesis and discusses future work.

Chapter 2
Integrated Services Packet Network Architecture (ISPN)

The existing Internet only offers the traditional Best Effort service which attempts to deliver data as best as possible, but without giving any service guarantees. The Integrated Service Packet Network (ISPN) is an extension to the existing Internet architecture. It was devised to provide a variety of additionally services with different qualities and service commitments. The service model is based on controlling the per-packet delay [BCS94], which implicitly includes a bandwidth guarantee. It does however not attempt to explicitly control the delay jitter in the network.

The key components of the ISPN are: (1) the Integrated Services offered, (2) the traffic control including the packet service discipline and the admission control, and (3) the reservation management. These components are outlined in this chapter. In Section 2.1, we however first discuss the QoS requirements imposed on the network by applications. This was motivated by the fact that these requirements were a fundamental driver for Integrated Services and the ISPN architecture. Section 2.2 then analyses the service specifications of the Guaranteed- and the Controlled Load service. Packet scheduling and admission control issues are discussed in Section 2.3. In this section, we look at compromises and design choices that can be made for the packet scheduler and summarize related work proposed for wide area networks. Section 2.4 describes the reservation management mechanisms including the setup of resources in the network and the reservation model. In Section 2.5, we then outline additional administrative control mechanisms such as Policy Control and Reservation Request Authentication. These are beneficial because reserving resources within switches and routers may enforce (a controlled) unfairness in the Internet. Section 2.6 describes the ISSLL framework and link layer specific aspects of the ISPN architecture. Finally, in Section 2.7, we discuss the relation of Integrated Services to the Differentiated Services approach.

2.1 Quality of Service Requirements

The characteristics of a variety of existing applications like telephony or video conferencing differ substantially from the traditional data applications such as file transfer or electronic mail. Differences can be found in: (1) the traffic pattern generated e.g. a constant- or variable data rate, (2) the communication type used e.g. unicast or multicast, or (3) in the network service guarantees required to perform well. In the following we first provide a taxonomy to classify applications. Afterwards, typical characteristics of these classes and the resulting network service requirements are discussed.

2.1.1 Application Classes

Table 2.1 shows five general application classes categorizing a multitude of applications used in today's data networks. These were identified in [Garr96] and consider existing audio, video, image and data applications in interactive, messaging, distribution and retrieval modes. We added a class for network management traffic since we believe this will play an important role in future networks. Since new applications are rapidly developed, Table 2.1 can not be complete. It however covers a representative set of characteristics which are likely to be found again in future solutions. Applications will further not always fit exactly into one of these classes. Virtual reality for example includes elements of the remote procedure call and of interactive audio and video.

No.	Application Class	Example Applications
1	Interactive Video	Video Conferencing, Distributed Classroom
2	Interactive Audio	Telephone
3	Interactive Text / Data	Banking Transactions, Credit Card Verification,
4	Interactive Conferencing	Multimedia Conferencing
5	Video Messaging	Multimedia E-Mail
6	Audio Messaging	Voice Mail,
7	Text / Data Messaging	Electronic Mail, Telex, Fax
8	Image Messaging	High Resolution Fax
9	Video Distribution	Television
10	Audio Distribution	Radio
11	Text Distribution	News
12	Image Distribution	Weather Satellite Pictures
13	Video Retrieval	Video on Demand
14	Audio Retrieval	Audio Library
15	Text / Data Retrieval	File Transfer
16	Image Retrieval	Library Browsing
17	Remote Terminal	Telecommuting, Telnet
18	Remote Procedure Call	Distributed Simulations, Distributed Games
19	Distributed File Service	Network File System (NFS)
20	Signalling Traffic	Network and Resource Management

Table 2.1: Application Classes and Example Applications [Garr96].

2.1.2 Application Characteristics and QoS Requirements

QoS requirements are typically specified in terms of: the bandwidth, the end-to-end delay, the delay jitter and the packet loss rate which are required by an application to operate well over the network. Other desired service properties may include: a failure recovery, security, message ordering, the absence of duplications or a fast service setup [Ferr90]. These will however be ignored in our discussion.

A popular way of defining the performance is by specifying a bound [Ferr90]. In this thesis, we follow [Ferr90] and define a *deterministic bound* as: $var \leq b \; = \; TRUE$, where *var* is the performance parameter to be controlled by the network and *b* is the bound. A deterministic bound implies an absolute, mathematically provable worst-case result. If *var* is for example the end-to-end packet delay then the above expression requires that *all* data packets conforming to the user's traffic speci-

fication must be transmitted by the network within less or equal than b time units. A statistical bound is defined as: $Prob(var \leq b) \leq probability\_bound$, where *probability_bound* is the probability that condition $var \leq b$ occurs. Providing a service with a statistical bound typically allows the network to achieve a higher resource utilization by weakening the assurance level of the service.

Interactive Applications

Interactive communications have time constraints which are often enforced by human beings exchanging informations. Telephony, video conferencing or certain banking transactions belong into this category. To achieve an interactive audio communication with a quality similar to that provided by the existing phone network, an end-to-end delay bound of 150 ms or less is required [G114_96]. The same bound should be requested for interactive video [WGS97]. The results reported in the literature however vary. In [BaOf98] and [GaDi97] for example, the authors request an end-to-end delay of about 100 ms for video conferencing and a distributed multi-user game, respectively. All values already include the data encoding and decoding times and the data transmission delay. In long distance calls, a substantial fraction of the end-to-end delay bound is already consumed by the propagation delay which is mainly determined through the speed of the physical transmission medium[1]. As a result, the delay budget for a LAN might only be in the order of 10 ms.

Additional constraints arise when audio and video data are to be synchronized. A skew of less than 80 ms was reported in [Stei96] to be acceptable by most casual observers. Informations exchanged by interactive applications are typically of less value or even become useless to the receiver when they arrive after a deadline. For audio and video applications this deadline is also called the playback point. Any data that arrives before the playback point is used to reconstruct the audio or video signal, whereas data that doesn't arrive in time is considered as lost and usually leads to glitches in the data output. Depending on the encoding scheme and the implementation, interactive applications are however more or less tolerant of packet loss. Intolerant applications e.g. a circuit emulation carrying audio traffic require a deterministic delay bound. This bound is then used as playback point. The deterministic nature of the bound ensures that all data packets arrive at the receiver before the deadline. In contrast, loss tolerant applications could be efficiently served with a statistical delay bound because they can tolerate an occasional packet loss. Adaptive applications are loss tolerant but can additionally vary their playback point according to the delay observed in the network [CSZ92]. One can expect that most of the audio and video applications built today will to some extent be loss tolerant and adaptive[2].

Messaging Applications

The second class in Table 2.1 contains messaging applications. These imply a person talking to a machine. In general, these applications do not have any stringent network service requirements

1. The propagation delay halfway around the globe is in the order of 100 ms assuming $5\ \mu s/km$ in fiber.
2. It remains to remark that there are already applications which do not only adapt the playback point but also their data rate. Example algorithms for this can be found in [BTW94], [MJV96] or [VCR98].

other than that the data are transmitted reliably and as fast as possible. The traffic generated is typically bursty and has a short lifetime since users attempt to utilize any spare network capacity in order to transmit data quickly.

Distributing Applications

Applications which distribute data to passively listening or watching users are listed in rows 9 - 12. A typical example is broadcast television. Since user interactions are not possible, there are also no hard constraints on the absolute end-to-end delay as long as a bound can be identified. End-to-end delays in the order of a few seconds as reached over satellite links are thus acceptable for these applications. If the delay becomes large then the delay jitter in the network needs to be controlled to minimize the buffer space requirements at the receiver.

Retrieval Applications

In contrast to broadcast applications, information retrieval has some interactive elements. The typical semantic implies users downloading information from a remote server. The level of interaction highly depends on the application type and the user behaviour. For Video on Demand for example, a delay in the order of 1 second from the time the user presses the playback button until the video appears on the screen seems to be acceptable for us, but will depend much on the activity and expectations of the user. The traffic characteristics can vary significantly. Video on Demand may e.g. generate a constant bit rate data stream, whereas File Transfer and Web browsing typically produce bursty and short lived traffic.

Computer Applications

The last group in Table 2.1 shows interactive computer applications. These typically imply a user driven and transaction based communication between computers in the network. QoS constraints may arise in respect to packet loss and delay. The packet loss rate may be critical when the missing data need to be retransmitted (Telnet) or impair the quality of the application service (NFS). A low end-to-end delay might additionally be required to satisfy the interactive user (Telnet). We believe that the smallest delays will be requested by distributed adventure games (or virtual reality systems) using e.g. remote procedure calls to update the view in the headset of each player. In this case, the LAN component of the end-to-end delay could well be in the order of just a few milliseconds.

Mapping Requirements to a Network Service

Based on their fundamental service constraints, most of the existing applications can be assigned into one of three groups. The performance requirements for a variety of applications such as data messaging or data retrival can be classified as elastic: applications are able to adapt to the resources available and utilize whatever spare capacity the network can offer. This does not imply that these applications are insensitive to the quality of service. Their performance typically improves significantly when they receive additional resources, but they also work correctly when the network is

highly loaded. The quality provided by the best-effort service might thus be sufficient to support this group of applications.

In contrast, interactive and intolerant applications like certain voice and video decoders, circuit emulation or time critical bank transactions require deterministic service guarantees. These applications cannot adapt to changing network conditions and usually do not work, or show a poor performance, when the service requirements are not met. To guarantee the quality of service desired, resources must be reserved within the network - unless the network load can always be maintained at a low level. In an Integrated Services network, applications in this group would request the Guaranteed service.

A third large group includes time sensitive, but adaptive applications. These applications work well in lightly loaded networks, but become more and more unusable as the network load increases. They do not require deterministic service guarantees covering every single data packet. Instead, maintaining a certain pre-defined bandwidth share and a low packet loss rate is sufficient for them to remain functional. It seems that in an Integrated Services network this group is most efficiently supported with the Controlled Load service.

2.2 Advanced Services and Their Services Interfaces

Beside the Guaranteed- and the Controlled Load service, a number of other services has been proposed for the ISPN. Examples are the Controlled Delay- [SPW95], the Predictive- [SPDB95], and the Committed Rate service [BGK96]. These services however have not been accepted for standardization and are thus not further considered in this thesis. Instead we focus on the former two proposed standards. Before we discuss their specifications, we make a few important definitions frequently used in the context of the ISPN architecture.

2.2.1 Definitions

A data transmission in the network is represented by an abstraction called a *flow.* A flow is a simplex stream of related data packets, all of which require the same network service [Zhan91], [BCS94]. In general, a flow relates to data packets from a single application. An example is a single unicast or multicast video packet stream. Full-duplex unicast communications thus imply two single flows, one in each direction. Multicast communications may require one multicast flow from each group member. In the absence of network topology changes, data packets from a single flow are expected to follow the same route through the network. A flow could however also be viewed as an aggregation of data streams from different applications. This is determined by the packet classifier and could for example be used for tunnels or Virtual Private Networks.

A network *service* can in general be viewed as a contract between the service user and the network. The user promises that its traffic will stay within the bounds specified, whereas the network agrees to deliver the user's data according to a pre-defined or pre-negotiated delivery policy. Reference [SW97a] defines a service as a named, coordinated set of QoS control capabilities provided by the

network. The service capabilities are declared in the service definition. It additionally specifies the information required by the network to provide the quality of service offered to the user. The term *Quality of Service* (QoS) refers to the properties of the packet delivery process and is described by parameters such as the available bandwidth, the packet delay and the packet loss rate [SW97a]. It can be viewed as a performance evaluation of the network's service.

Service definitions typically describe the end-to-end behaviour of the network. Internally, the network may however consist of a multitude of components such as routers, switches, gateways, connecting wide area links and shared or switched LANs. Within the ISPN, any component that is potentially capable of exercising QoS control over data packets traversing it, is called a *network element* [SW97a]. To provide end-to-end guarantees to the user, appropriate service guarantees must thus be provided by all network elements along the data path.

The finest granularity of packet stream for which resources can be allocated in the ISPN is the flow. For different flows, different services and service parameters can thus be selected. Applications negotiate the service with the top resource management layer. On each network element, the resource management then requests the service on behalf of the application from the underlying link layer.

2.2.2 The Guaranteed Service

The Guaranteed service [SPG97] provides a deterministic end-to-end delay bound for all data packets of a flow provided that the flow's traffic conforms to the specified traffic parameters. This implies a guaranteed bandwidth and the assurance that no data packets will be lost due to a queue overflow within the network. The service includes a delivery model similar to that offered by traditional circuit switched networks and will thus allow the support of legacy applications across packet switching data networks.

The Guaranteed service is specified based on two concepts [SPG97]: the token bucket filter and an approximation of the fluid model. The token bucket filter is used to describe a flow's traffic. It contains two parameters: the flow's token bucket rate r and the token bucket depth δ. How these parameters can be used to characterize the flows data output, is described precisely later in Section 6.1.2 in Chapter 6. The fluid model is an abstraction that attempts to hide the network's complexity. In a perfect fluid model network, a flow essentially receives the service that would be provided by a dedicated wire of bandwidth R between source and receiver. In this case the delay through the network is bounded by δ / R, provided that $R \geq r$ and the flow's traffic stays within its specified token bucket parameters. Real networks however differ from this simple model. This is considered in two error terms C_{tot} and D_{tot}, which are used to describe how a particular implementation deviates from the fluid model. The differences arise because in real networks, the time required to access the physical medium, and to pre-empt a running network service can not be neglected. Furthermore, data is transmitted in packets which are usually not divisible. This may have an impact on the delay bound

provided by the service discipline, as shown for Weighted Fair Queuing (WFQ)[1] in [PaGa93], [PaGa94]. Using the terms C_{tot} and D_{tot}, the end-to-end delay bound then becomes [SPG97]:

$$D_{End\text{-}to\text{-}End} = \frac{\delta}{R} + \frac{C_{tot}}{R} + D_{tot} \qquad \text{for} \qquad R \geq r \tag{2.1}$$

The error terms C_{tot} and D_{tot} are end-to-end quantities. They are computed by adding up the *C* and *D* error terms, respectively, for all network elements along the data path between the data source and the receiver. The parameter *C* is the rate-dependent error component describing the data backlog caused by the packetization effect. In a WFQ scheduler for example, *C* would be the maximum packet size which the flow uses [SPG97]. The parameter *D* specifies the rate-independent deviation of the network element from the fluid model. In a shared LAN with bounded medium access time D_{it} for example, the minimum for *D* would be D_{it}.

The guaranteed service is invoked by specifying the flow's traffic characteristic called *TSpec* and the reservation requests called the *RSpec*. The TSpec includes: (1) the token bucket rate *r*, (2) the token bucket depth δ, (3) the peak rate *A*, (4) a minimum policed unit p_{min}, and (5) a maximum packet size p_{max}. The RSpec contains: (1) the service rate *R*, and (2) a slack term *S*. The first two parameters in the TSpec are the flow's token bucket parameters. The parameter *A* denotes the peak data rate generated by the flow. p_{max} is the size of the biggest packet that is said to be conform to the traffic specification. The minimum policed unit p_{min} allows an estimate of the per-packet resources needed. Any data packet smaller than p_{min} will be treated as being of size p_{min}. The exact formats are given in [SW97b], [Wro97b][2]. Furthermore, we have: $P_{min} \leq p_{min} \leq p_{max} \leq P_{max}$, where P_{min} and P_{max} denote the minimum and maximum link packet sizes, respectively.

The service interface does not allow an explicit specification of the end-to-end delay desired for a flow. Instead the network provides a delay bound for the traffic characteristic and the reservation request specified. Applications however can control the end-to-end delay bound by adjusting the service rate *R* in the service request, where higher service rates will typically reduce the queuing delay bound as can be observed in Equation 2.1. The service rate however may not be below the token bucket rate *r* ($R \geq r$). To be able to use Equation 2.1, all network elements along the data path must export a value for the *C* and *D* error term, so that the end-to-end parameters C_{tot} and D_{tot} can be computed. The parameters are carried to the user by the reservation setup protocol discussed later in Section 2.4. The slack term *S* in the RSpec is the delay difference by which the end-to-end delay bound desired by an application is higher than the requested delay bound δ / R computed with the service rate *R*. Specifying a non-zero slack term offers more flexibility to network elements in reserving resources and might thus increase the change of the reservation request to become accepted.

1. Weighted Fair Queueing (WFQ) is also known as Packet Generalized Processor Sharing (PGPS).
2. Note here that some of the *TSpec* variables used in this thesis differ from the ones used in [SW97b] and [Wro97b]. The mapping is: $b = \delta$, $p = A$, $m = p_{min}$ and $M = p_{max}$.

Whenever the peak rate A of a flow is unknown or specified as infinite[1] then the end-to-end delay bound is computed using Equation 2.1. A known and finite peak rate leads to a tighter bound for the end-to-end delay [SPG97]:

$$D_{End\text{-}to\text{-}End} = \frac{(\delta - p_{max})(A - R)}{R(A - r)} + \frac{(p_{max} + C_{tot})}{R} + D_{tot} \qquad \text{for} \qquad A > R \geq r \tag{2.2}$$

$$D_{End\text{-}to\text{-}End} = \frac{(p_{max} + C_{tot})}{R} + D_{tot} \qquad \text{for} \qquad R \geq A \geq r \tag{2.3}$$

Equation 2.2 provides an optimized result for the case that $A > R \geq r$ holds. It consists of three additive components. The first term describes the time it takes to clear the burst δ sent at peak rate A. The second and third components represent the delay introduced by the composed error terms C_{tot} and D_{tot}. If the service rate R is higher than the peak rate A then there is no queuing delay caused by burst δ which leads from Equation 2.2 to Equation 2.3.

2.2.3 The Controlled Load Service

The Controlled Load service [Wro97a] attempts to approximate the service that an application would receive from the best-effort service under unloaded network conditions. No absolute guarantees for service parameters such as the end-to-end delay or the packet loss rate are given.

The specification of the service is intentionally minimal which will allow a wide range of implementation approaches and trade-offs between e.g. resource utilization and implementation costs. An unloaded network is understood as *not heavily loaded or congested*. Admitted flows may assume: (1) a very low packet loss rate *close to the packet error rate* of the transmission medium, and (2) a low average delay in the order of the path's minimum transmission delay. More precisely, the average queuing delay should not be significantly larger than the flows *burst time*. If the flows traffic is characterized using a token bucket filter then the burst time is given by δ / r. The difference to the best effort service is that the above conditions are guaranteed even when the underlying link is congested. This is achieved by using admission control and by isolating Controlled Load traffic.

The service is invoked by specifying the flows TSpec as defined in [SW97b]. The TSpec parameters are the same as discussed for the Guaranteed service. In contrast to the latter, the Controlled Load service does not export any service parameters e.g. the expected end-to-end delay or packet loss rate to the user.

2.3 Packet Scheduling and Admission Control

Network service guarantees primarily depend on the packet scheduling algorithm used in switching nodes and the corresponding admission control conditions. The IETF however does not standardise

1. If known then the line rate of a link could always be used as the flows peak data rate.

these algorithms and conditions. This allows designers to trade-off switch functions with other properties such as costs or flexibility, provided that the implemented service matches its service specification.

2.3.1 Desired Properties

Each service discipline could be viewed as a compromise between: (1) its cost and complexity, (2) its isolation capabilities, (3) its efficiency, and (4) its flexibility. A low complexity ensures that the service discipline can actually be implemented in high speed switching nodes. Relevant constraints in respect to LAN switches are: (1) the costs for implementing the algorithm in hardware e.g. the gatecount, the number of memory accesses, etc. (2) the performance, for example how much it slows down the packet forwarding process in comparison to the traditional First-Come-First-Served (FCFS)[1] service discipline, and (3) the amount of status information required to support the algorithm.

Traffic isolation in the network ensures that data packets generated by non-characterized or misbehaving data sources do not degrade the quality of service given to other flows. This can be seen as the basic property required to provide service guarantees in existing data networks. It can be implemented for each individual flow or for classes of flows. Isolation and service protection might also be desired for the best-effort service such that: (1) it does not starve due to excessive prioritized traffic admitted, and (2) all best-effort flows receive a fair share from the total resources available for this service.

The efficiency of the service discipline describes how well resources are managed. This aims at an allocation that uses as few network resources as possible while still providing the requested quality of service for each admitted user. In contrast, flexibility reflects the ability to support service guarantees for a wide range of performance requirements including requests for different delay bounds. A static priority scheduler with l priority levels, where $l > 1$, for example can typically only support $l - 1$ delay bounds, assuming that the lowest level is used for best-effort traffic and admission control is applied for all levels $l > 1$. A scheduler with $l = 2$ can thus only provide a single delay bound which might however not match the bound each service user actually wanted. Often observed is also a coupling between bandwidth and delay allocation such that more bandwidth needs to be reserved in order to reduce the queuing delay in the scheduler [PaGa93], [Gole94], [FiPa95]. This leads to a low resource utilization when low delay bounds are requested for low bitrate flows.

In general, a service discipline with low implementation costs, per-flow isolation, a high efficiency and flexibility is desired. This however can typically not be achieved since all four properties have strong dependencies. Improving one of them often has a negative impact on another. The WFQ scheduler for example isolates single flows based on a sorted priority queue mechanism. This enforces a high degree of service protection and tight delay bounds. Sorting data packets into a

1. See for example [Cruz91a] for an analysis of the FCFS service discipline.

queue at high speed however also increases the complexity and may degrade the forwarding performance. This has been evaluated in [Kesh91]. Additionally costs are created by maintaining per-flow state in switching nodes.

The optimum compromise between these constraints depends on the special case. Basic factors to be considered include (1) the service to be supported, (2) the target device type e.g. a router, a LAN switch and the constraints of the corresponding market such as costs or the target customers, (3) the properties of the links connected to the device (in a LAN environment, this may include a shared medium access), (4) the target network location e.g. the backbone, the segment-, workgroup-, or desktop level, and (5) the state of the technology available for the implementation. Alternative solutions for providing quality of service should also be considered. We do this for a LAN environment in Section 4.4 after we studied the best-effort performance of 802.12 networks. We continue with the basic mechanisms that can be used in the design and discuss their constraints in providing the properties introduced in this section.

2.3.2 On Fundamental Design Choices and Trade Offs

There are four basic degrees of freedom in designing a service discipline [Kesh97 - Chapter 9]: (1) the number of priority levels, (2) the service order within each of these levels, (3) the degree of flow aggregation within each level, and (4) whether a level is work-conserving or non-work conserving. In the following, we briefly discuss these principles in the context of the CSZ scheme [CSZ92], which is used as reference in [BCS94] to demonstrate how Integrated Services can be realised. In our considerations, we however substitute the Predictive service with the Controlled Load service since the former is currently not considered in the standardization.

The CSZ scheduler is composed of two different service disciplines: Weighted Fair Queuing (WFQ) as described in [DKS89] and Static Priorities (SP). Both are arranged in a hierarchy: WFQ - SP - WFQ. WFQ is used at the top of the hierarchy to provide Guaranteed service on a per-flow basis. This scheduler further separates the Guaranteed service from the Controlled Load- and the Best Effort service such that a certain resource share is guaranteed for the latter two services. An SP scheduler with two priority levels is employed to isolate the Controlled Load from the Best Effort service. All flows receiving Controlled Load service are aggregated into the high priority queue of the SP scheduler and receive service according to the FCFS service discipline. Best effort data packets assigned to the low priority level are however served according to the WFQ discipline to support a controlled link sharing on a per traffic class basis. This could for example be used to control the resources consumed by different organizations or different network protocols [FlJa95].

Considering existing LAN environments, the CSZ scheme seems to be too complex to become widely implemented in LAN switches or hubs in the near future. This is due to the low cost and high speed constraints imposed on these devices. Traffic isolation is nevertheless required but at a lower granularity to keep the costs down. In general, traffic isolation in a packet switching network can be achieved based on: (1) static priorities [ZhFe93], (2) a round-robin service [KaKa90], [ShVa95], (3),

a sorted priority queue mechanism [DKS89], [VZF91], [BeZh96a], or (4) a time framing strategy [Gole90].

The simplest way of isolating traffic is performed by the Static Priority scheduler. The number of priority levels determines the costs and the number of different service qualities supported by the scheduler. Flows with similar service requirements are aggregated into the same level. Within each priority level, data packets may however be served according to different service disciplines, just as could be observed for the Controlled Load- and the Best-Effort service in the CSZ scheme. The FCFS service discipline provides the highest level of aggregation and is the simplest to implement. It however also allows a maximum interaction between different flows which may lead to a loose delay bound or unfairness when data sources sends more data than their reserved share.

The level of control can be improved by using a discipline with higher isolation capabilities, as implemented for Best Effort traffic in the CSZ scheme. WFQ might here for example differentiate three different classes of best effort traffic identified based on the protocol identifiers: IP, IPX[1] and SNA[2]. Within each class, flows are still aggregated and may thus interact. WFQ however isolates each traffic class and can thus guarantee that the SNA traffic always receives its allocated share from the total best effort resources.

From this example, one can identify three general levels of flow aggregation: (1) none - as performed for Guaranteed service users in the CSZ scheme, (2) per-class - as implemented for Best-Effort traffic, and (3) a total aggregation - as used for Controlled Load flows. Increasing the level of aggregation however typically also decreases the efficiency and the flexibility of the service discipline due to the loss in control. Flows receive the quality of service of the class they are in and not a tailored delay bound. The advantages are lower implementation costs due to less status information to be managed and a lower processing overhead in switching nodes.

Finally, both schedulers used in the CSZ scheme can be classified as work conserving. A work conserving scheduler is one that only runs idle when there is no data packet in the system. In contrast, a non-work conserving system may hold data packets but its output may nevertheless run idle. This is based on an eligibility time explicitly or implicitly assigned to each data packet in existing non-work-conserving schemes [ZhKe91]. The eligibility time determines how long a data packet must be held before it can be forwarded. The scheduler may thus run idle when (1) there are no data packets in the system, or (2) there are packets in the system, but all these packets are waiting to become eligible for departure.

In networks consisting of switching nodes running a work-conserving scheme, the traffic pattern of a flow may become more and more distorted due to network load fluctuations [Zhan95]. To provide service guarantees in such a network, the distortions introduced at each hop along the flow's data path must be characterized. This may be difficult, especially in meshed networks with feedback

1. See [Siga94] for informations about the Internet Packet eXchange protocol.
2. For a brief overview on the System Network Architecture (SNA) see [Tane89 - Chapter 1].

effects since different flows may interact across different segments. Once the distortions have been characterized, appropriate resources can be reserved to maintain the quality of service. This often results in buffer space requirements increasing monotonically with the number of hops in the data path [PaGa94], [BeZh96a], [GVC96]. The important advantage of work conserving schemes however is that resource shares are only enforced under overload. Whenever flows do not use the bandwidth reserved for them, then this can be used by other flows using the same or any other service.

In contrast, non-work conserving service disciplines reshape arriving data flows and thus reconstruct a flow's traffic pattern before the forwarding to the next switch. This simplifies the network analysis and ensures that buffer space requirements remain constant along the data path [Zhan95]. Beside providing a delay bound, some schemes can additionally control the delay jitter [KaKa90], [ZhFe93]. Holding data packets in switches however results in higher average packet delays [Zhan95] and requires a traffic shaping mechanism such as the (δ, r) regulator [ZhFe93] or a framing strategy [Gole90]. In contrast to work conserving schemes, non-work conserving service disciplines enforce resource shares regardless of the current work load. Data flows are thus rate regulated even when sufficient free network capacity is available.

2.3.3 Related Work

There has been much research on service disciplines for Integrated Services packet networks. In contrast to previous work on queueing analysis, these schemes can provide deterministic delay bounds on a per-flow basis. Related research on work-conserving schemes includes: Weighted Fair Queueing (WFQ) [DKS89], [PaGa93], [PaGa94], and its derivations [Gole94], [BeZh96a], [BeZh96b], [GVC96], Delay Earliest-Due-Date (Delay-EDD) [FeVe90], and Virtual Clock [Zhan91], [FiPa95]. All these schemes use a sorted priority queue mechanism for allocating bandwidth and delay. They however differ in respect to the way the packet indices used in the packet reordering process are computed.

None of the admission control conditions and delay bounds derived for these schemes however apply to existing shared medium networks such as Demand Priority or Token Ring LANs. This is because these disciplines require exclusive access to network resources as provided by full-duplex point-to-point links. WFQ for example controls the order in which data packets are sent based on *finish numbers*. A finish number is assigned to each data packet as it arrives at the server. It depends on the length of the data packet and on the arrival history of the corresponding flow. Data packets are served with increasing finish number: whenever a transmission is finished then the next packet to be sent is the one with the smallest finish number. In a shared medium LAN with several WFQ servers, each of these servers will however forward data packets independently, without considering other servers on the network. Due to the work-conserving character of the scheme, servers may then transmit data packets with high finish numbers too early such that data packets with lower finish numbers queued at another server miss the delay bound. Similar considerations can be made for other work-conserving service disciplines using a sorted priority queue mechanism.

Related work on non-work conserving service disciplines includes: Stop-and-Go [Gole90], Hierarchical Round Robin (HRR) [KaKa90], Jitter Earliest Due Date (Jitter EDD) [VZF91], and Rate-Controlled Static Priorities (RCSP) [ZhFe93]. Stop-and-Go relies on one fundamental mechanism: a timed, network wide framing structure similar to Time-Division Multiplexing (TDM). Since this cannot be efficiently performed on Demand Priority LANs without significant standard changes, the scheme is not considered any further. Jitter EDD is an extension of Delay EDD that also uses a sorted priority queue mechanism. The scheme has thus similar constraints as Delay EDD. Furthermore, to provide a bound for the delay, Jitter EDD assumes point-to-point network links with a constant propagation time. The delay in shared medium networks may however be variable.

HRR is based on a hierarchically, multi-level framing concept. Each frame is divided into a fixed number of time slots. Bandwidth is allocated by reserving time slots at a selected frame level. All time slots and all frames are served in round-robin order. The basic concept of HRR could potentially also be used in LAN switches to enforce a deterministic delay bound. This is because the framing concept is able to restrict the network access for all real-time flows across defined time intervals. The number of different delay bounds provided by such as server will however depend on supporting mechanisms e.g. the number of priority levels of the underlying link layer technology. The existing admission control conditions thus do not hold in shared 802 type LANs and would have to be modified to reflect relevant technology constraints such as e.g. the Demand Priority overhead. Furthermore, the efficiency of HRR relies on fixed packet sizes as found for example in ATM. In a LAN environment, where flows use variable sized data packets, this may lead to a poor resource utilization since all time slots would have to be allocated equivalent to the maximum packet size used.

The key feature of the RCSP service discipline is the separation of the server into two components [ZhFe93]: a set of rate regulators and a Static Priority scheduler. This decouples the bandwidth- from the delay allocation. The rate regulators control the traffic distortion for each real-time flow in the network. The SP scheduler enforces the service quality. The number of different delay bounds is determined by the number of priority levels. The RCSP packet scheduling concept can also be used in LAN switches interconnected by shared medium networks but requires supporting mechanisms in the underlying link technology. It is attractive because existing shared and switched LAN technologies often already provide one or more priority levels with a bounded access delay. In this case, the SP scheduler in the scheme can be replaced by the link layer medium access mechanism. The admission control conditions may therefore depend on the constraints of the technology specific medium access.

It can be concluded that most of the service disciplines discussed in this section would show a poor performance when used in their existing form in shared medium or half-duplex switched networks. This is not surprising because all of them were designed for switching nodes in wide area networks. The exception is RCSP whose packet scheduling concept can also be used in existing LAN technologies including Demand Priority networks. We nevertheless found it beneficial to look at all these

solutions because this helped us to clarify some of the fundamental mechanisms required in shared medium networks to control the packet delay. It remains to remark that excellent comparisons of the concepts, the properties and the complexity for most of the service disciplines in this section can be found in [ZhKe91], [Zhan95] and [Kesh97 - Chapter 9].

2.4 Dynamic Reservation Setup in the ISPN

The ISPN offers two mechanisms to setup reservations. The first is based on the traditional network management using the Integrated Services MIB [BKS97a], [BKS97b], [BKS97c]. The second is a dynamic reservation protocol called Resource ReSerVation Protocol (RSVP) [ZDE+93], [BZB+97]. Its basic concepts are outlined in the following. For the details we however refer to [ZDE+93].

2.4.1 Fundamental RSVP Concepts

Applications use RSVP to dynamically setup, modify and tear-down reservations in the network. It is a signalling mechanism which is used to carry control information between source and receiver, and to all intermediate network elements such as routers in the data path. Resources are reserved for single flows on a hop-by-hop basis. Carried control information includes the flow's traffic specification TSpec, the reservation request RSpec, and additionally control information required e.g. for classification and Policy Control. At each network element, RSVP first communicates with the local Policy Control to check whether the originator of the request has administrative permission to make the reservation. Afterwards, the reservation request is passed to the local Admission Control to check the resource availability. If the data path includes a bridged LAN then this might trigger a link layer reservation request and additional signalling to a LAN resource manager. Relevant link layer specific mechanisms are discussed in Section 2.6.4. When the reservation request is accepted, control information is passed to the local classifier and scheduler to enforce the service quality for the flow. Afterwards, control information is sent to the next network element, which then performs the same control actions, and so on, until resources are setup at all network elements along the data path. If the reservation request is however rejected then the reservation setup is stopped and a reject message is sent back to the user.

RSVP supports unicast and multicast reservations. When multicast is used, different group members may request a different service quality. Reservations are initiated by the receiver. Before a receiver may however ask for resources, information about the data source and the data path, called *Path State*, must be installed in all network elements between the data source and the receiver. This is similar to setting up a circuit in the telephone network. Path State is installed using RSVP *Path* control messages. These are periodically multicasted by the source and exactly follow the data path. Multicast receivers may request resources after they received a Path message. Reservation requests are sent towards the data source using RSVP *Resv* messages. These contain the receiver's RSpec, classification-, policy control information and always travel along the reverse path established by the last Path message.

RSVP allows receivers to dynamically select which data sources may use the network resources reserved for the receiver. For this, two general degrees of freedom can be identified. First, a receiver may either select data sources explicitly using an address identifier or may use a wildcard and thus select all sources sending to a particular multicast group. Secondly, a reservation can either be distinct, which means assigned to a single data source, or can be shared by many sources. The combination of these attributes allows different types of reservations called *Reservation Styles*.

RSVP currently supports three different reservation styles: *FixedFilter-* (*FF*), *SharedExplicit-* (*SE*), and *WildcardFilter* (*WF*). The FF style implies a distinct resource reservation and an explicit sender selection, and is thus similar to the reservations made in the traditional telephone network, even though the latter uses a sender based reservation setup. A SE style reservation allows data packets from different data sources to share the resources reserved for a receiver. All data sources must however be explicitly listed. The WildcardFilter reservation implies the attributes shared reservation and wildcard sender selection. It allows all sources of the same multicast group to share the resources. Data sources however do not need to be explicitly specified. To efficiently support reservations made by different multicast receivers, reservation requests are merged at branch points in the multicast data distribution tree.

During the reservation setup, Path and Resv messages only install *Soft-State* in the network. To prevent this information from timing out, it must be periodically refreshed. Path State is refreshed by re-sending a Path message. This is carried out by the data source. A reservation is refreshed by the receiver by re-sending a Resv message towards the data source.

RSVP was primarily designed for supporting resource reservation on a per-flow basis. Considering however that existing Internet backbone routers can serve up to 100.000 simultaneous connections [Kesh97 - Chapter 9], per-flow reservations in the backbone do not seem to be economically feasible at the moment, given the amount of memory available in existing routers. The use of RSVP is however encouraged within a single or a small number of administrative domains of an intranet [MFB+97], since in those networks, scalability and security issues will be more manageable or do not occur. We refer to [Schw97] for a more detailed discussion of RSVP's limitations and constraints.

2.4.2 Reservation Model - OPWA

The reservation model describes how an application negotiates for a quality of service level [BCS94]. In RSVP, the reservation of resources is only initiated by RSVP path messages as these travel from the receiver towards the source. On each network element the receiver's request is either accepted or rejected. This is called a *One Pass* reservation model.

To assist receivers in constructing an appropriate reservation request, Path messages carry the traffic characterisation of the data source (TSpec) to all receivers. RSVP additionally supports an enhancement known as *One-Pass-With-Advertising* (OPWA) [ShBr95]. The basic idea is to supply sufficient network information to the receiver so that resources can be reserved successfully and accurately.

This additional information is carried in Path messages. For the Guaranteed service this includes the composed network error terms C_{tot} and D_{tot} which enable the receiver to compute the resulting delay bound. Other advertised parameters are the maximum hop-count, the minimum bandwidth and the minimum path latency. Indicated is also whether a particular service is available on all network elements along the data path or not. We refer to [Wro97b] for the rules and details of how these parameters are collected. Reference [ShBr95] contains a comparison between One- and Two Pass reservation models. The latter approach is for example used in the *Tenet Scheme* [BFM+96].

2.5 Policy Control, Reservation Request Authentication and Pricing Issues

Policy Control determines who is allowed to use how much and what sort of network resources. It implies a resource access control according to administrative rules. Control is required because resource reservation enforces a discrimination between users in the network such that selected users may receive more resources than their fair share. The user selection process could e.g. be determined by the user's position in an organisation as provided by an hierarchical quota system, or can be motivated through a pricing scheme. A few sample scenarios can be found in [Herz96]. Policy rules are however not standardized. They are a local matter within administrative domains and will be proprietarily negotiated between different organisations e.g. Internet service providers. Required are however transport mechanisms to carry policy informations: (1) from the receiver to network elements e.g. the network edge router, and (2) between network elements and a policy server. The former is provided by RSVP. [HPRG97] contains a proposal for the second requirement based on a simple client-server model.

Authentication can be used as protection against forged reservation requests. It seems to be mandatory when pricing is used to regulate the resource access and resources are setup dynamically through an untrusted domain. The RSVP authentication is based on the *IP Authentication Header* [Atki95], and performed on a hop-by-hop basis between neighbour routers. The IP Authentication Header can be used to carry authentication data within each RSVP packet. It is typically inserted between the traditional IP header and upper layer protocol headers such as UDP or TCP. The authentication data is e.g. computed using the MD5 [Rive92] hash function. The basic idea is to create a hash from any transmitted data and a secret identifier which is shared between the communicating parties. Modification of the RSVP packet will be detected by a non-matching hash computed at the receiver of the message.

Pricing is used by network providers to recover operational costs and to create an incentive to use network resources efficiently. Higher charges will further justify the better quality of service provided for Guaranteed and Controlled Load service users in the Internet. Many approaches for this have been proposed in the literature. A discussion of their details however would lead far beyond the scope of this thesis. Traditionally, network providers charged flat fees for a given access link. To charge costs fairly, however, typically requires a usage based pricing scheme. A simple approach might for example be based on the peak data rate and the duration of the call. More complex

schemes additionally consider parameters such as the average data rate or an estimation of the burstiness of the traffic. Pricing can also be viewed as an approach to reduce congestion in the network. This is discussed in Section 4.4 in the context of Local Area Networks.

2.6 The ISSLL Framework for Reserving Resources in Shared and Switched LANs

The Integrated Services over Specific Link Layers (ISSLL) working group adopted the ISPN service model for a use at the link layer. This ensures a simple one-to-one service mapping from the network layer to the link layer. The working group investigated related standards, link technology specific characteristics and general link layer mechanisms required for supporting these services across IEEE 802 type LANs, ATM and slow speed serial links. In this section, we however only focus on issues relevant to 802 type networks because 802.12 LANs belong into this category. Before these issues are outlined, we introduce definitions used in the link layer and ISSLL context.

2.6.1 Definitions

The following discussion is an excerpt of [GPS+98 - Section 3]. It assumes the reader to be familiar with the principles of layering as provided by the ISO OSI Reference Model (see for example [Tane89 - Chapter 1]).

A *LAN bridge* or *switch* is a link layer packet forwarding device as defined in the 802.1D standard [ISO93]. In this context, the terms bridge and switch refer to the same device and are thus completely interchangeable. Bridges may connect a multitude of network *segments* to form a link layer network. A network *segment* denotes a single physical layer connecting two or more devices with link layer functionality e.g. bridges or hosts. It may imply a shared, half-duplex or a full-duplex medium. The term segment is however typically used in the context of a shared medium network which may include one or more physical layer forwarding devices called *repeaters* or *hubs*. These devices can potentially connect many hosts to the network. Finally, the term *subnetwork* is used to denote a group of devices with network layer functionality such as hosts and routers sharing the same link layer network. This implies that there is no router in the data path between any two network layer devices in the subnetwork.

Note that in the ISPN architecture, a subnetworks is considered as a single network element. This is because the link layer topology remains hidden to the network layer. In the case that the subnetwork supports the Guaranteed Service, it thus only has to export a single *C* and *D* error term to the network layer, even though it may consist of many QoS aware switches.

2.6.2 Mapping Integrated Services onto IEEE 802 MAC Service Mechanisms

The MAC datagram service standardized in [ISO93] allows data sources to specify a *User Priority* for each data packet passed to the link layer for transmission. The User Priority is a 3 bit label which potentially enables bridges to differentiate data packets without having to parse the packet in more detail. This however can only be exploited when the label is actually carried in the packet

header, which unfortunately is not the case for all data packets in existing LANs. When using the 802.5 frame format for example, the User Priority is encoded in the Frame Control field of the link level header. In contrast, the traditional Ethernet and 802.3 data packets do not carry this identifier at all. If the 802.3 frame format is used across 802.12 networks then the User Priority is first mapped to either normal or high priority. The result is then encoded in the starting delimiter of the 802.12 data packet.

Recently, two enhancements to the MAC bridge standard have been proposed in the IEEE P802.1Q and P802.1p supplements [ISO97a], [ISO97b]. These define (1) a consistent way of carrying the User Priority across heterogeneous link technologies by using an extended link layer frame format, and (2) a general model for differential queueing within bridges based on the User Priority. The User Priority allows to differentiate up to seven different services. P802.1p however neither specifies the service that could be built on top of this mechanism nor the service discipline that must be used to achieve the corresponding service quality. It only defines a mapping between the User Priority and a particular queueing mechanism in the switch. Assuming for example static priority scheduling, which is defined as the default service discipline in switches supporting P802.1p, the User Priority may identify the priority queue in which the data packet it to be placed. This is a one-to-one mapping when the switch supports seven priority queues, but might imply the aggregation of several User Priority levels into a single queue when less priority levels are implemented. The default mapping for a static priority switch with two priority levels for example is 0 -3 to priority level 0 (lower priority), and 4 -7 to priority 1 (higher priority). Switches may further use any other appropriate service discipline to enforce quality of service such as Weighted Fair Queuing or Rate Controlled Static Priorities.

The mapping of IETF Integrated Services into the above extended MAC service model is defined in [SSC97]. The Controlled Load service currently maps into User Priority 4, the Guaranteed service into 5 and 6. The two levels assigned for the Guaranteed service differ by different delay bounds.

2.6.3 The Difference to the Network Layer Integrated Services Architecture

The ISSLL framework described in [GPS+98] allows a wide range of mechanisms, including significant trade-offs between complexity and supported features, to provide Integrated Services across shared and switched LANs. This differs substantially from the architecture standardized for the network layer which requires that routers implement core mechanisms such as packet classification, merging, policing and/or reshaping on a per-flow basis.

The ISSLL framework defines a simple taxonomy for LAN switches including four basic categories. All classes differ in respect to their classification and isolation capabilities. The *Class 0* switch is a standard 802.1D switch without any QoS support. A *Class I* switch supports the default classification and priority queuing mechanisms specified in P802.1p. This represents the minimum standard for supporting Integrated Services. It can for example be used to isolate data packets according to their network service, but not to differentiate single flows within each service class. This limita-

tion allows designers to build low cost but QoS aware switches. The simplicity however also results in a loss of features. A Class I switch basically can only support FixedFilter reservations. Shared reservations require per-flow policing/reshaping mechanisms as e.g provided by the (δ, r) Regulator in switches in order to protect other flows using the same service. Furthermore a Class I switch cannot support multicast receiver heterogeneity because it cannot queue data packets differently on different output ports.

Class II switches differ from this by their ability to change the User Priority for a data packet on a per-output port basis. Multicast receivers may thus either use the advanced service but may also refrain from making a reservation and thus receive best effort service. The support for complete multicast heterogeneity is however not required. *Class III* switches are able to classify data packets on a per-flow basis using the RSVP filter specification carried in the FilterSpec. They might additionally support per-flow policing and traffic control. A *Class IV* switch which is however not specified in the ISSLL framework, could then be viewed as a switch with the same capabilities as an ISPN Integrated Services router, however such a switch is also likely to have the same complexity and costs. Our solution for the Guaranteed service described in Chapter 6 requires a network consisting of Class III or Class IV switches. Bridged networks providing the Controlled Load service according to the scheme in Chapter 7 may also include (or may only include) Class I and/or Class II switches.

Even though the ISSLL framework offers a high degree of implementation flexibility it also specifies fundamental mechanisms which must be used when providing Integrated Services across shared and switched LANs. These are similar to the network layer ISPN requirements. First, Integrated Services must be provided using resource reservation and admission control. The network must be able to police and isolate single or classes of flows such that service guarantees according to the service definitions can be provided. This may however be based on mechanisms in hosts or at the edge of the bridged LAN. Other requirements include the installation of Soft State in switches, scalability and the ability to interwork with existing solutions like the Synchronous Bandwidth Manager defined for FDDI networks. We refer to [GPS+98] for a further discussion of these topics.

2.6.4 Link Layer Signalling Issues

The reservation setup across a switched network requires similar signalling mechanisms as used at the network layer within the ISPN. During the reservation setup, applications or the upper layer resource management e.g. RSVP requests the service from the underlying link layer and specify the service identifier, the TSpec, RSpec, and the IP source and destination addresses of the flow. The link level resource management then reserves the requested resources along the link layer data path. Returned is the result of the admission control which is typically a yes or no answer. Note that a positive result may only reflect the successful reservation for the first segment in the data path. The request may later become rejected in the case that: (1) the data path contains several segments, (2) resources are reserved on a segment-by-segment basis using an independent link level resource

manager on each segment, and (3) the admission control for one of these segments failed. This is identical to the optimistic approach taken by RSVP at the network layer.

There are three additional mechanisms to be supported by the resource management at the link layer: (1) the support for shared medium segments, (2) the ability to translate IP network addresses into MAC addresses, and (3) the support of a dynamic User Priority selection. The first requires a static or dynamic election mechanism such that resources on shared segments are managed by a single resource manager. The address translation between IP and MAC addresses is performed using the standard Address Resolution Protocol (ARP) [Plum82]. During the reservation setup, the result of this translation is carried to bridges within the data path to assist them in resolving the data path to the destination. A dynamic User Priority selection enables switches to control the mapping between Integrated Services and the User Priority. The key idea is to enable the network to overrule any User Priority value suggested by a data source for a particular reservation request. This was motivated by the fact that it is typically much easier to upgrade the mapping table in switches than to change this at each host on the LAN [SSC97]. Network switches are assumed to be upgraded using the traditional network management or a manual configuration.

The link layer signalling mechanism proposed in the IETF for RSVP based admission control across 802 type LANs is called Subnet Bandwidth Manager (SBM) [YHBB97]. SBM extends RSVP such that the link layer reservation setup is piggybacked onto the layer 3 RSVP signalling. The key design idea is that the link layer resource manager inserts itself as a hop into the data path of the RSVP flow. This causes all RSVP related messages, in particular the Resv message, to be routed through that link layer resource manager. Utilizing this, the SBM can support exactly the same features as RSVP. Additionally it provides solutions for the three issues discussed above.

2.6.5 Why Resource Reservation in LANs ?

There are two fundamental drivers for an Integrated Services network: (a) economical benefits from exploiting resource sharing, and (b) service guarantees and quality of service. We believe that both drivers also apply to Local Area Networks even though costs and performance aspects in LANs differ substantially from the wide area. Note here that a LAN does not necessarily have to only interconnect hosts. It could for example also be used in Network Access Points (NAPs) to interconnect routers from different Internet Service Providers (ISPs).

Economical Aspects

Resource reservation allows service guarantees for selected flows even when the network is operated at a high load. Provided that solutions are cost competitive to pure bandwidth, an Integrated Services LAN may enable a network administrator to reduce costs through resource sharing. A simple example might be a University Campus LAN shared between businesses receiving Controlled Load service and students using Best Effort. The service selection is enforced through pricing. An Integrated Services LAN will further be beneficial for higher level services such as network man-

agement. Breakdowns in today's LANs are often caused by just a single faulty application flooding the network with multicast or broadcast traffic. Traffic isolation and service guarantees can reduce the time required to identify the misbehaving data source and to recover from the breakdown. Even though faulty applications might also generate priority traffic, this will be limited through the traffic enforcement in the host's Operating System kernel or through traffic control mechanisms within LAN switches. Out-sourcing and remote management based on reliable local network management capabilities might further reduce the costs and may be inevitable in the future.

Service Guarantees and Quality of Service

There are several reasons why it is hard for existing LANs to provide service guarantees. These are outlined in the following:

1. Control Time Scale in Feedback Schemes: reactive control schemes as used for best effort traffic cannot control congestion that occurs over timescales shorter than the Round Trip Time. This was discussed in Section 1.1.4 in Chapter 1.

2. LAN traffic properties: LAN traffic is extremely bursty across time scales from milliseconds to hours [LeWi91], [LTWW94]. A considerable part of this traffic is transmitted using the User Datagram Protocol (UDP) [Post81c]. An extreme example is given in [Claf94 - Chapter 5] for a departmental LAN whose traffic traces showed a UDP share of over 90%, mostly caused by Network File System (NFS) (see for example [Stev94 - Chapter 29]) data packets. The corresponding campus backbone still carried between 37.7 and 62.4% UDP traffic. Since UDP, in contrast to TCP, does not include a congestion control mechanism, large percentages of UDP traffic increase the risk for a temporarily overload considering that existing hosts are sufficiently powerful to fill up a high speed link with a capacity of e.g. 100 Mbit/s. Furthermore, the bursty nature of the traffic and the use of UDP make it harder for reactive congestion control schemes to adapt to the changing network conditions because the available LAN capacity is continuously changing as data sources start and stop transmitting data.

3. LAN topology: todays LANs are heterogeneous in respect to the link capacities and technologies used. Speed mismatches may cause buffer overflow when the load is high and traffic bursts are forwarded onto links with a lower capacity e.g. from a 1 Gbit/s link to a 100 Mbit/s segment. A modest increase of the buffer space in switches will in general however not prevent packet loss due to congestion [FoLe91]. The congestion problem is also not likely to be solved with high-speed links [Jain90]. Similar considerations can be made for switches with a large number of ports. Correlating traffic burst arriving from several input ports may temporarily overload an output link and cause congestion. Existing LAN switches typically have eight to thirty-two ports. It can be expected that this significantly increases in the next few years.

Many existing LANs however do not exibit signs of congestion because they are always operated at a low network load. This is one alternative solution to ensure a probabilistic quality of service as we

will discuss later in Section 4.4 in Chapter 4. To explicitly compute this probability however is hard if not impossible because (1) best-effort traffic can typically not be characterized and enforced, and (2) the constraints of the underlying link technologies make a network analysis difficult.

2.7 Relation to the Differentiated Services Approach

Recently, the Differentiated Services architecture was proposed by the IETF [BBC+98], [BBB+98]. It has the same fundamental goal as the ISPN: to extend the existing Internet architecture such that additional services providing quality of service can be supported. The key difference between both approaches is that unlike the ISPN which reserves resources on a per-flow basis, the Differentiated Services architecture provides quality of service for traffic aggregations which may include a multitude of flows. This was motivated by potential scaling problems that may occur when per-flow state, which basically scales linearly with the number of admitted flows, needs to be maintained at routers in the core of the Internet. Furthermore, reserving resources for aggregated flows allows to simplify the packet classification in core routers. This is achieved by exploiting the IP Precedence Field [Post81a] to identify the packet forwarding policy in the router. Data packets with the same identifier will thus receive the same treatment independent of the actual flow to which they belong to. Reference [NBB+98] contains the new definition of the IP Precedence Field, now called Differentiated Services Field. It is intended to supersede the definition in [Post81a]. The new field includes a number of Differentiated Services Code Points (DSCP) each of which identifying a particular packet forwarding policy called Per-Hop-Behaviour (PHB). The PHB specified in [JNP98] for examples provides the equivalent service that a user would receive from a leased line of fixed bandwidth. The corresponding DSCP to be carried by all data packets using this service is: 101100. The particular mechanisms to implement a Per-Hop-Behaviour will however not be standardized. For the example in [JNP98], the authors suggested the use of Static Priorities or Weighted Fair Queueing as service discipline.

If we compare the Differentiated Services architecture with the ISPN and the ISSLL framework, many similarities can be identified. First, to enforce a Per-Hop-Behaviour providing service guarantees requires the same fundamental traffic control mechanisms as discussed for the ISPN. In particular this includes traffic policing and/or traffic reshaping, the service discipline in switching nodes and the corresponding admission control conditions. Furthermore, resources are reserved for simplex data streams. Unlike the ISPN, the Differentiated Services architecture however attempts to move more expensive functionality to the edge of the Internet. Complex classification, policing and reshaping mechanisms may for example be only performed at the edge of the network such as the WAN access router. Routers in the core of the Internet may only support a simple priority scheduler and a packet dropping mechanism. This ensures simplicity in the core where the highest traffic density can be expected. The ISSLL framework implies a similar concept for bridged LANs. Our approach to provide Controlled Load quality of service in Chapter 7 for example is based on: (1) traffic reshaping mechanisms that are only implemented in hosts and routers, (2) a simple static priority scheduler in LAN switches, and (3) admission control. Furthermore, the User Priority dis-

cussed in Section 2.6.2 can be viewed as a link-layer Differentiated Services Field since it simplifies the packet classification in a similar way.

We believe that the mechanisms used within LANs to provide Integrated Services can be re-used to enforce Differentiated Services, for example when the admission control conditions become applied to aggregated flows in the LAN. Our results in Chapter 7 have further shown that Controlled Load quality of service can be achieved based on a very simple packet scheduler. We thus do not expect that implementations of the Differentiated Services approach will offer a simpler solution for LAN switches.

Chapter 3
Measurement Methodology

This chapter describes the methods which we used to achieve the measurement results presented in this thesis. We begin with a summary of the clock terminology and characteristics. Section 3.2 then introduces the basics of the test network and the traffic trace driven approach for generating realistic traffic patterns in the network. The latter is based on two tools: (1) a LAN Traffic Monitor which we used to record data flows, and (2) a traffic-trace driven Traffic Generator which generated data traffic with pre-defined characteristics during the tests. In Section 3.3, the design and the performance of the LAN monitor are reported. Section 3.4 describes the Traffic Generator and addresses accuracy issues of the traffic trace driven approach.

In the second part of this chapter, we turn to the methods for measuring performance parameters in shared and switched networks. We begin with the parameter bandwidth in Section 3.5. Section 3.6 discusses our centralistic approach for measuring packet delay and why we did not choose a distributed solution based for example on the widely available Network Time Protocol (NTP) [Mill92]. Finally, Section 3.7 describes the method that was used to determine the packet loss rate in different network topologies.

3.1 Clock Terminology and Characteristics

In this thesis, we closely follow the terminology defined in [Mill92] and [Paxs97 - Chapter 12]. In general, computer clocks are used to measure time. They typically consist of a precision quartz oscillator[1]. The smallest frequency at which the time is updated is the clock's *Resolution* or *Precision*. Despite of a high precision, a clock can still be inaccurate when its time differs from the Absolute Time defined by the national standard. A clock's *Accuracy* is thus defined as how close the clock's knowledge of time is to the Absolute Time. Another characteristic is the frequency stability. It describes the clock's ability to maintain the Absolute Time after being set. The frequency difference between a clock and the national standard at a particular moment is defined as the clock's *Skew*. The variation of the Skew is denoted as the clock's *Drift*.

In all our experiments, the clock accuracy did not have any impact on the measurements results, because our measurements are based on time *differences* between two events, both of which are time-stamped. Using an appropriate, centralized measurement setup ensured that time-stamps related to each other were taken by the same clock. The end-to-end packet delay for example is

1. For a discussion of computer clock models see for example [Mill92 - Appendix G].

measured by comparing time stamps taken at the entrance-point and at the exit-point of the tested network. This is performed by the same workstation and for each data packet of a pre-selected flow.

More relevant for the accuracy of our measurements are drift and skew. In [Mill94] three general components of these frequency errors are identified: (1) noise, (2) wander effects, and (3) the mean frequency error. Noise occurs across intervals of less than a minute and is for example caused by variations of the power supply regulation. [Mill94] remarks that this is typically not a problem. Wander effects are observed over timescales from several minutes to hours and mainly depend on temperature variations. Even though wander effects typically have a strong impact on the frequency of the quartz oscillator, they are not significant in our case because all measurement results received for the packet end-to-end delay are far below 100 milliseconds; mainly in the order of a few milliseconds. The same consideration could also be made for noise. Furthermore, since all results are based on a single clock, error sources such as relative skew and drift occuring between different computer clocks [Paxs97 - Chapter 12] do not have to be considered. Mean frequency errors can be neglected for similar reasons because they typically occur over intervals greater than an hour [Mill94]. All these considerations however assume a stable workstation clock oscillator that is capable to provide accurate time stamps.

Finally, we use the term *measurement accuracy* to denote the accuracy of the entire measurement approach. This implies a bound for all relevant errors which distort the final measurement results such as DMA time variations, possible clock reading errors or hardware latency variations.

3.2 Generating Realistic Traffic Patterns in the Test Network

3.2.1 The Test Network

The test network consisted of a number of standard 802.12 hubs, switches and HP 9000/700 workstations. Measurements were carried out in: (a) single hub, (b) cascaded (multi-hub), and (c) half-duplex switched topologies. The network included a maximum of 15 workstations, 5 802.12 LAN switches or 10 hubs. All devices were connected to each other via Category 3 UTP links of defined length, with a maximum of 200 m. The exact topology varied according to the needs of the particular experiment and is thus described with the setup and the measurement results.

In the experiments, all workstations used the HP-UX 9.05 operating system and standard EISA 802.12 LAN adapter cards. The switches were HP Switch 2000 LAN switches. The Switch 2000 is an output buffered, modular switch based on a single system bus that is shared by all switch ports. It has a bus performance of 1 Gbit/s and can support a maximum of 12 802.12 ports. The switch is thus slightly oversubscribed.

Whenever performance parameters were measured in the network, each active workstation was configured to run in one of three configurations. These differed by the software running in user space and in the kernel during the experiments. Delay measurements were taken by a single workstation which we called the *Measurement Client* (MClient). Several other workstations were used to impose

802.12 high and normal priority cross traffic on the network. We called these workstations *High-* and *Normal Priority Traffic Clients* (Traffic Clients) according to the priority level of the traffic generated. In each test, a single machine operated as the measurement *Controller*. The Controller synchronized the actions of all High- and Normal Priority Traffic Clients, and of the MClient. It further collected statistics from the hubs and the LAN switches in the test network such as the number of data packets discarded due to a buffer overflow, or the amount of data forwarded. The Controller enabled us to automate the experiments and to control the parameter settings on all Traffic Clients and the MClient from a single machine.

3.2.2 Traffic Trace Driven Measurements

There are two basic experimental approaches which are typically used to confirm theoretical results: simulations and measurements. Simulations allow a wide range of experiments, but require a realistic model of the medium access and the data transmission process. Furthermore, traffic characteristics need to be known and mapped onto accurate and tractable source models. In contrast, measurements in real networks are not based on a model and thus avoid potential mistakes made in the design of such a model. However, they typically only provide results for the specific environment in which the experiments were carried out e.g. a university campus or a corporate intranet with certain traffic characteristics. It is usually not possible to study all interesting cases such as the network behaviour under overload since this heavily affects the service quality or might even make the network unusable for the duration of the measurement.

We chose the experimental approach in favour of simulations due to the rather complicated signalling and timing constraints built into the Demand Priority medium access protocol, especially when multi-hub 802.12 topologies are managed. Measurements were also valuable in further investigating and understanding the network behaviour and allowed us to verify our network packet transmission model and the results derived in the theoretical part of this thesis.

Our test network was completely isolated from the site LAN. To generate realistic traffic patterns within the network, we originally intended to run a number of applications on each Traffic Client. Not all of our workstations however had the audio or video hardware support required for the test applications. We further observed performance constraints when many applications run simultaneously on the same machine. This was caused by: (1) the high number of context switches, and (2) the two copy operations required for passing data packets from the user space to the LAN adapter card.

To overcome both constraints, we used a traffic trace driven approach. For each application, we first recorded a 2 hour test trace using our LAN Traffic Monitor. In the experiments, the traces were then passed to the kernel based Traffic Generator which generated an almost identical data stream to the original trace monitored on the network. This was for example used to simulate the case in which each workstation on the LAN takes part in a video conference. To generate N homogeneous data sources from the same Traffic Client, we multiplexed N copies of the original trace into a single

trace file, where each of the *N* copies had a different, randomly chosen, start offset into the original trace. On reaching the end of the trace, a source wrapped around to the beginning. In the experiments the trace file describing the aggregated traffic of *N* data sources was then passed to the Traffic Generator.

Random start offsets were further applied at the beginning of each experiment. This was carried out on all Traffic Clients and on the Measurement Client to avoid traffic synchronisations between different workstations. The independence was further increased by exploiting source traces of 2 hour length for the trace multiplexing. The measurement interval however was typically only in the order of 30 minutes for all trace driven measurements in the test network. Two traces used by different data sources might thus differ completely over the entire measurement interval even though they originate from the same source trace.

The above method is basically identical to the one used by Garrett in [GaWi94] and Jamin in [Jami96]. Garrett exploits it in a trace driven simulation to simulate data from different data sources, based on a single 2 hour variable bit rate, JPEG [Wall91] encoded video stream. Jamin simulates a number of different *Fractional Autoregressive Integrated Moving Average* (FARIMA) sources using a single pre-computed data set for all sources. This is then passed into a simulation to investigate the behaviour of a measurement based admission control scheme.

The main advantages of the traffic trace driven approach are its performance and its flexibility. A high performance can be achieved by multiplexing data sources before the actual measurement. Data packets are allocated in the kernel and do not have to be copied from the user- into the kernel space. Flexibility is given by avoiding hardware dependencies and application specific informations in the trace files. The latter permits experiments based on trace-files generated from arbitrary traffic models. Chapter 4 and Chapter 7 for example also report results based on Pareto source models.

3.3 A Kernel Based Traffic Monitor

3.3.1 Design and Implementation Issues

The Traffic Monitor is implemented on a standard HP C100 workstation. It has a single 802.12 LAN adapter card which connects the workstation to the shared test network. The adapter card operates in promiscuous mode and looks at each packet on the network[1]. The monitor consists of two parts: (1) the *Data Collector* which is embedded into the device driver of the LAN adapter card within the kernel, and (2) a *Data Storage Process* implemented as a user space UNIX demon.

When a data packet is received on the LAN adapter card then the packet is instantly DMA-ed into kernel memory. A high priority hardware interrupt informs the kernel about the new packet. At

1. Beside broadcast and multicast traffic, network nodes in shared 802.12 networks typically only receive unicast data packets addressed to them. This is due to a filter function performed by 802.12 hubs as outlined in Section 5.1.1 in Chapter 5. The promiscuous mode is enabled using link level signalling between the node and the connecting hub.

interrupt context[1], the Data Collector records a pre-defined set of packet information for later analysis. This information may for example include a time-stamp and the link level header of the packet. Afterwards, the data packet is instantly discarded if not addressed to the monitor itself.

To store the packet information, the Data Collector manages two large continuous buffers which are allocated within the kernel prior to the monitoring. We called them *Packet Information Buffers*. The Data Collector always only writes into one of these buffers. If a buffer is filled then packet information is placed into the other buffer, provided this buffer is empty. The Data Collector further sends a UNIX signal to the Data Storage Process which then copies the contents of the full buffer from the kernel to the disk of the workstation. This is performed at a lower priority than the data recording. After copying the packet information, the kernel buffer is marked empty by the Data Storage Process and may then be re-used by the Data Collector. The communication between the Data Collector and the Data Storage Process is based on UNIX signals and *ioctl* system calls.

The traffic traces used in this thesis were obtained by recording the parameter pair: <*packet arrival time*; *packet length*> for each data packet of a selected flow. This required only 10 bytes storage space for each packet monitored (8 bytes for the time-stamp and 2 bytes for the packet length)[2]. The arrival time was measured using the Interval Timer (Control Register *CR16)* [HP92b - Chapter 2] of the PA-RISC 7200 processor. Since all our time measurements are based on this control register, its function is described in more detail in the following.

Internally, CR16 actually consists of two registers. The first contains a counter which is basically incremented at instruction rate. This provides a clock with a resolution of 10 ns on the C100 workstation. Reading CR16 returns the value of this counter. In contrast, writing on CR16 always modifies the second internal register. This register holds a comparison value. Whenever the values on both registers are identical then a hardware timer interrupt is triggered.

To measure the arrival times of data packets, the Traffic Monitor reads CR16 instantly after receiving a data packet. This is performed at the beginning of the interrupt service routine and only causes a minimum overhead. Our function to do this consists of just five instructions and is coded in PA-RISC assembler. All time-stamps itself thus have a granularity of 10 ns. The Traffic Monitor however only records times with a granularity of 1 μs because this seemed to be sufficient to us.

It remains to remark that register CR16 is also used by the operating system timer. Even though it is incremented every 10 ns, standard HP-UX 9.05 only updates the system time every 10 ms. By using the nanosecond counter in CR16 directly, we not only obtain time-stamps with a high granularity, but also avoid time jumps such as reported in [Paxs97]. Time jumps are the result of clock adjustments. These are required to set a new system time e.g. to correct long term drift and skew effects. If not considered or avoided, they may lead to invalid measurement results. The nanosecond counter in CR16 however is not adjustable and thus cannot be altered when a new system time is set.

1. The data recording is performed within the interrupt service routine at processor level 6.
2. The format of the resulting traffic trace is identical to the one used in the LAN traffic traces *BC-pAug89.TL* or *BC-pOct89.TL* in the Internet Traffic Archive: *http://ita.ee.lbl.gov/html/traces.html*.

3.3.2 Performance and Measurement Accuracy

The Traffic Monitor cannot capture data packets at the maximum 802.12 link data rate. It however can handle network loads far in excess of the traffic generated by the audio and video data sources monitored. There are three possibilities why the monitor can fail to record a packet: (1) there is no packet buffer on the LAN adapter card so that incoming data packets are dropped, (2) both Packet Information Buffers are filled up such that no further information can be stored by the Data Collector, or (3) the disk runs out of space. The latter error was not an issue because we never recorded traffic traces for longer than 2 hours.

To estimate the monitor's performance, we measured the maximum load that can be captured without a single packet loss. This was done for different packet sizes used for the data transmission. The results are shown in Figure 3.1. They were also of general interest in respect to the Measurement Client because it run on the same workstation type. The performance was measured by using four Traffic Clients generating constant bit rate data traffic with a pre-defined data rate and packet size. All Traffic Clients, the Controller and the Traffic Monitor were connected to a single 802.12 hub using 5 m UTP cables. For a set of packet sizes ranging from 64 bytes to 1500 bytes, we then increased the network load until a packet loss occurred. The incremental step of the load was 1 Mbit/s, the measurement interval for each individual measurement was 10 minutes. 8 Mbytes were allocated for each of the two Packet Information Buffers within the kernel.

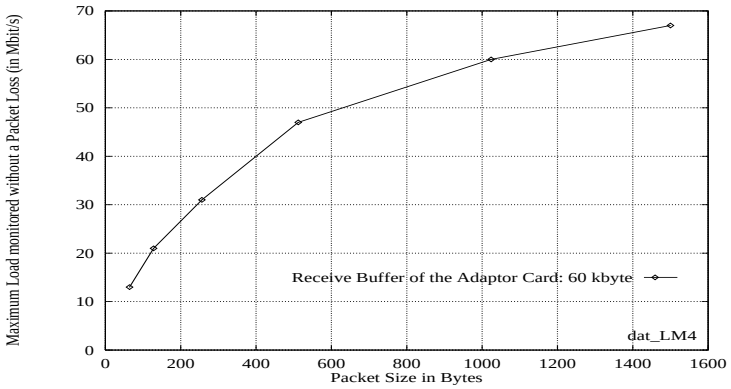


Figure 3.1: Maximum Data Rate monitored without a Packet Loss in Dependence of the Packet Size used for the Data Transmission.

Figure 3.1 shows that the Traffic Monitor can capture flows with a maximum data rate of about 13 Mbit/s without packet loss when all data is transmitted with 64 byte packets. As the packet size increases, the maximum data rate also grows. For 1500 byte packets, the monitor recorded a data rate of 67 Mbit/s without a single packet loss. In the experiment, packet drops on the adapter card were detected by the Data Collector which read the *Dropped Packet Counter* on the Cascade 802.12 MAC chip [HP94]. This counter is incremented by the hardware whenever a data packet is dropped due to insufficient buffer space on the adapter card which itself can run at maximum link rate. Another counter was held in kernel memory to count the number of packet drops caused by a buffer

overflow in the kernel. We called this counter the *Buffer Overflow Counter.* All results in Figure 3.1 were caused by a buffer overflow in the kernel. The system bottleneck is the Data Storage Process which could not save the data as fast as the Data Collector was storing them. Once both Packet Information Buffers were full, packet information thus went missing which then led to an increase of the Buffer Overflow Counter.

The performance further depends on the size of the Packet Information Buffers[1]. Larger buffers require fewer copy operation and reduce the total processing overhead. In the extreme case, the buffers are allocated such that the entire trace can be stored in kernel memory and is only copied to disk after the test. This is what we finally did when we recorded the test traces. The 2 hour, 3 Mbit/s JPEG encoded MMC1 trace analysed in Section 4.2.1 for example only required a single Packet Information Buffer of about 21 Mbytes to store the entire trace.

The measurement accuracy of the Traffic Monitor is determined by: (1) the time it takes to DMA the arriving data packet from the adapter card into kernel memory, (2) the latency caused by the interrupt processing in hardware, (3) the time to interrupt the running software process and to invoke the interrupt service routine, and (4) the accuracy of the time stamp assigned to each data packet. Since all data packets are passed through the same receive path, only the maximum *delay variations* of these operations actually need to be considered e.g. the time difference in DMA-ing a minimum or maximum sized data packet. This is because we are interested in time differences and do not rely on the clock accuracy.

As part of the experiments reported in Section 6.5.2, we found that all hardware related operations for sending and receiving a single data packet to and from the LAN adapter card require about 145 μs. The context switch takes about 25 μs. The time variation of a pure receive operation will however be much lower than this.

Other factors to be taken into account are: (1) a possible queuing delay on the LAN adapter card, (2) interrupts from the hardware timer, and (3) the interference on the system bus of the workstation. The queuing delay can be neglected because all monitored flows generated data rates of less than a few Mbit/s. While recording data packets, we further never observed more than one DMA packet receive descriptor in use. This indicates that packets were never queued on the adapter card.

The system timer interrupt service routine may cause inaccuracies because it is the only function invoked that has a higher priority than the Data Collector. We however modified this routine such that it only updates the system time on the workstation and schedules a new timer interrupt. Any additional work is performed by a lower prioritized routine. Since the code path of the interrupt service routine only consists of a few hundred instructions, the resulting error cannot be larger than a few microseconds.

1. There are several ways of how the overall performance of the Traffic Monitor could be improved. The simplest solution is to use a faster workstation. Alternatively the contents of the Packet Information Buffer could be DMA-ed from the kernel memory directly to the workstation disk. This would save the copy operation to and from the user space. The latter approach however requires additional kernel modifications.

Interference on the system bus is mainly caused by the Data Storage Process copying/mapping data from the kernel memory to the user space and from there DMA-ing them to the disk. It competes with the network DMA operation copying packets from the adapter card to the kernel memory. The system bus on a C100 is called *Runway*. It is an HP proprietary bus interconnecting the PA-RISC 7200 processor, the main memory and several bus converters [HP92b - Chapter 1]. The 802.12 LAN adapter card is connected via an EISA bus to an EISA/Runway Bus Converter[1]. The disk is connected via a Fast-Wide SCSI/Runway Bus Converter. The Runway system bus is 64 bit wide and multiplexes addresses and data. The overhead consists of one address cycle for every four data cycles, which results in a sustainable bus bandwidth of 5.12 Gbit/s considering a clock rate of 100 MHz. This is sufficiently high to ensure no interference between the network DMA and the Data Storage Process.

The Traffic Monitor has thus a measurement accuracy below 100 μs which is in the same order of magnitude as the accuracy of the high resolution monitor described in [LeWi91].

3.4 A Trace Driven Traffic Generator

3.4.1 Design and Implementation Issues

The Traffic Generator runs on all Traffic Clients and on the Measurement Client. Its design is similar to the design of the Traffic Monitor. The core is a *Packet Generator* which generates data packets according to a trace file. To achieve high performance and accuracy, the Packet Generator is implemented in the 802.12 device driver in the kernel. The trace file is read from the workstation disk by a user space UNIX demon which copies the data from the disk into a Packet Information Buffer in the kernel. Similar to the Traffic Monitor, two of these buffers are managed. For each data packet to be generated, the trace file must have an entry with the format: <*packet arrival time; packet size*>.

The Packet Generator attempts to generate data packets with the same interpacket time[2] as specified in the trace file. This is based on the operating system timer. Every time the Packet Generator is invoked, it updates the virtual clock managed for the traffic trace and generates the data packets that have become eligible in the last timer interval. By default, eligible packets are instantly passed to the network for transmission. In Chapter 6 and Chapter 7 however, we use the Traffic Generator in combination with a Link Level Rate Regulator which allows to further regulate the output of the Traffic Generator. Once the information in the first Packet Information Buffer has been used, the Packet Generator continues with the second one. The buffer management and the communication between the Packet Generator and the UNIX demon are basically identical to the mechanisms used in the Traffic Monitor. Packet information is however moved into the kernel. We further recorded the error-case when a Packet Information Buffer was not updated fast enough by the UNIX demon such that the Packet Generator was blocked in its operation due to missing packet information.

1. The theoretical maximum transfer rate of the EISA bus is 264 Mbit/s.
2. The interpacket time for any two data packets in the trace file is the difference between their packet arrival times.

3.4.2 The Accuracy of the Approach

In contrast to the Traffic Monitor whose time-stamping operation is only driven by packet arrivals, the Traffic Generator requires a local timer interrupt to trigger the packet generation process. Since the accuracy of the Traffic Generator mainly depends on the resolution of the operating system timer which has however only a default granularity of 10 ms, we implemented a fast timer in the HP-UX kernel. The implementation is based on CR16 and reported in Section 6.4.3 in Chapter 6. For the Traffic Generator we used a timer granularity of 1 ms since this seemed to be a good compromise between the processing overhead and the measurement accuracy that can be achieved.

To measure the accuracy of the trace driven Traffic Generator, we used two workstations in a shared, single hub test network: one was running as Traffic Generator, the other as Traffic Monitor. For several test traces, we then monitored the data packets sent by the Traffic Generator into the test network. Afterwards we compared the original trace passed to the Traffic Generator with the trace measured by the Traffic Monitor. The results are shown in Figure 3.2 which contains the cumulative distribution function $F(diff < t)$ for the *interpacket arrival time differences* for all packets *i* of the original trace and the trace measured, where $diff^i = d^i_{orig} - d^i_{meas}$ for all $i > 1$, and $d^i_{orig} = t^i_{orig\_arrival} - t^{(i-1)}_{orig\_arrival}$. For the measured trace, we have the equivalent: $d^i_{meas} = t^i_{meas\_arrival} - t^{(i-1)}_{meas\_arrival}$. The parameters $t^i_{orig\_arrival}$ and $t^i_{meas\_arrival}$ are the packet arrival times of the *i*‘st data packet in the original and the measured trace, respectively.

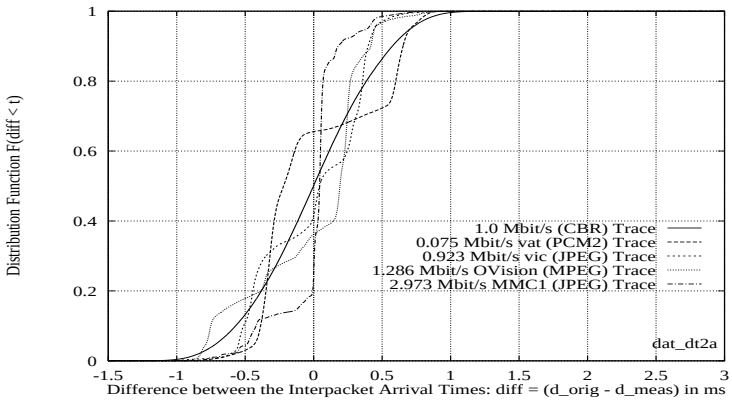


Figure 3.2: Difference of the Interpacket Arrival Times between sent and measured Audio and Video Data Traces.

Figure 3.2 contains the results for five data traces of 2 hour length. The first is a 1 Mbit/s Constant Bit Rate (CBR) trace with randomly chosen packet sizes between 64 and 1500 bytes. The measurement results for this trace are identical to our expectations: a symmetrical distribution with a mean of 0 ms. All samples are basically within the time interval: *[-1 ms, 1 ms]* caused by the timer granularity of 1 ms. The other four traces are traces from variable bit rate audio and video applications using variable packet sizes. These traces are identical to traces used later in this thesis. Since the details of these traces are not relevant for the main result of this test, we refer to Section 4.2.1 for a description of the applications and the configurations used in recording them. It can be observed,

that the results for the latter four traces differ significantly from the CBR trace. The reason for this is undetermined, but might be caused by regular traffic patterns in respect to the interpacket arrival times and the packet sizes within these traces. The main result is that the difference between the original and the measured trace is small and basically determined by the timer resolution (granularity) of the Traffic Generator. For the 1.286 Mbit/s, MPEG [LeGa91] encoded traffic trace for example, 99 percent of all packet interarrival times differed by an absolute value of less than 0.85 ms.

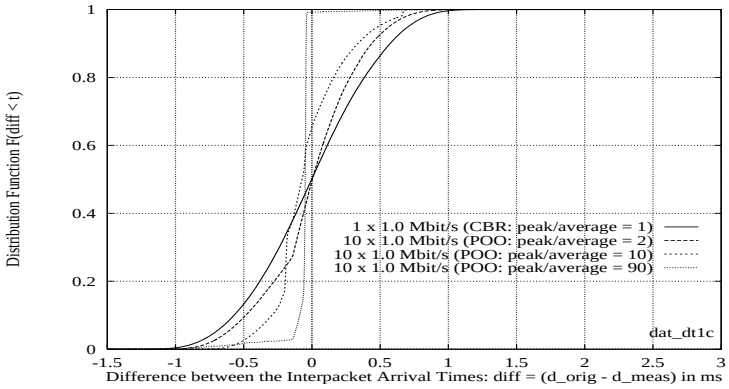


Figure 3.3: Difference of the Interpacket Arrival Times between sent and measured Pareto Test Traces.

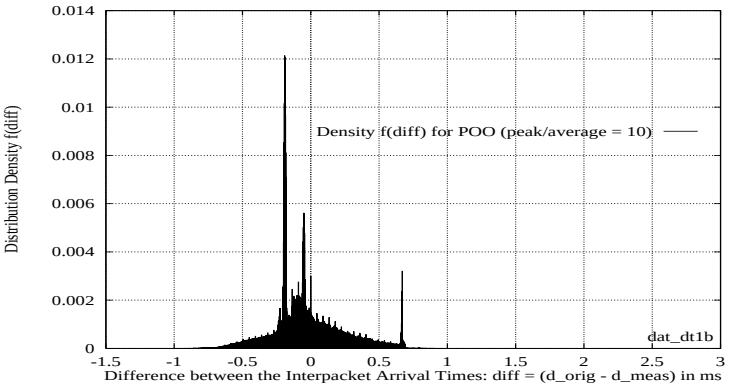


Figure 3.4: Delay Distribution (Density) for Curve 3 (*POO: peak/average = 10*) in Figure 3.3.

The most accurate results were however measured for traces generated with an ON/OFF traffic source model and a peak data rate close to the link bandwidth. Figure 3.3 shows the results for 3 traces which we computed according to a Pareto (POO) source model. These were measured using the same setup as described for Figure 3.2. For comparison, we further added the measurement result for the 1 Mbit/s CBR trace from Figure 3.2. An example for the corresponding distribution density is shown in Figure 3.4.

All three Pareto traces consist of 10 multiplexed flows with homogeneous source model parameters and only differed by the ratio of the peak to average data rate used in the source model. This, we

varied from 2 to 90. Each flow was computed using an ON/OFF source model with pareto distributed ON times and pareto distributed OFF times. During each ON time, an average of $N = 10$ data packets was generated. The average data rate was 1 Mbit/s resulting in an average of 10 Mbit/s for each trace. The pareto shape parameter for the ON interval was 1.9, the equivalent parameter for the OFF time was 1.1. This is identical to the parameters selected for source *POO3* in Section 4.2.2 in Chapter 4. For a discussion of the Pareto source model, the parameter selection and the method used to compute Pareto distributed traces with a certain peak to average rate ratio, we also refer to Section 4.2.2. It remains to remark that all three Pareto traces in Figure 3.3 further contained data packets with the fixed length of 1024 bytes.

The 99 percentile of the results for trace four (peak/average = 90) in Figure 3.3 is 0.35 ms which is far below the 1 ms timer resolution. This accuracy is caused by the network whenever the interpacket time between subsequent data packets in the trace is close to the link speed. This was the case in this setup. The measurement results in Section 4.3.1 show that for a single hub network and a data transmission using 1024 byte packets, the maximum data throughput on the 802.12 network is just about 89.5 Mbit/s . The peak data rate of the POO sources was 90 Mbit/s. Even though the Packet Generator sends packet bursts at intervals of 1 ms, the network spaces them out during the transmission such that data packets arrive at the Traffic Monitor with an interpacket gap equivalent to 90 Mbit/s. A similar effect can also be observed for the JPEG encoded MMC1 trace in Figure 3.2.

3.5 Measuring the Throughput in Shared and Switched LANs

The method we used for measuring the throughput is based on the Management Information Base (MIB) counters [Flic96], [McCR91] maintained in hardware on the managed hubs and switches in the test network. These counters were periodically read by our Measurement Controller using SNMP *Get-Request* control messages [CFSD90]. An alternative was to use the Traffic Monitor which however would have had difficulties to accurately measure data rates close to the network capacity. External traffic monitors are further less suitable for measurements in switched networks because they cannot easily be connected to point-to-point links between switches[1].

Using a MIB based approach avoided any performance and connectivity problems that might have occurred with the Traffic Monitor. Our hubs and switches however only support the standard MIB and do not maintain counters on a per-flow basis. Any MIB based scheme can thus only measure the aggregate load on a test link. This was sufficient for our experiments because we typically simultaneously measured the end-to-end delay for the tested flows in order to confirm the quality of service provided. The delay measurements however recorded the delay of every single data packet belonging to the flow.

1. A possible solution is to connect the Traffic Monitor to a promiscuous switch port and set appropriate filter entries in the switch such that a copy of all data packets from and to the test link is also forwarded through the promiscuous port.

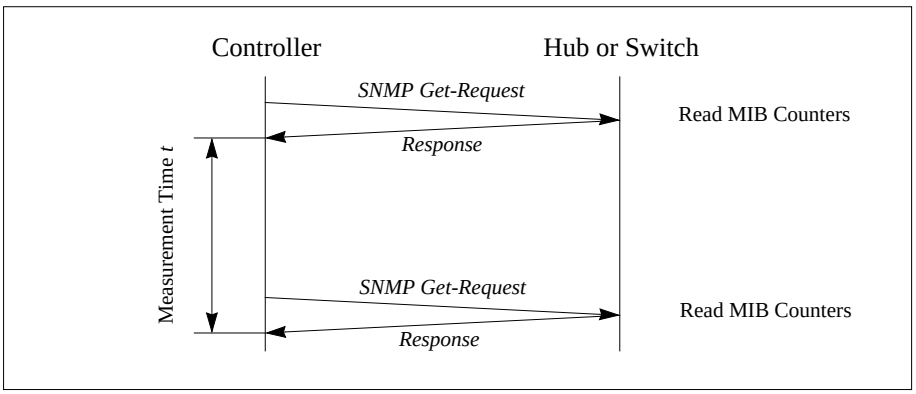


Figure 3.5: Control Message Sequence for Measuring the Network Throughput.

Figure 3.5 shows the control message sequence which is used by the Controller to retrieve the MIB counters from a hub or switch in the test network. Table 3.1 contains the object identifiers for the MIB counters used. It also lists the packet drop counter which enabled us to determine the packet loss rate in switches on a per-port basis. Once the Controller has received the start and finish values for the counters, it computes the average throughput over the measurement time *t*. If we assume that the counters did not wrap around during the measurement interval, then we have for example for a half-duplex switch port *x*:

$$r^x = ((ifOutOctets^x_{finish} - ifOutOctets^x_{start}) + (ifInOctets^x_{finish} - ifInOctets^x_{start})) \;/\; t \qquad (3.1)$$

where r^x is the data throughput. $ifOutOctets^x_{start}$ and $ifOutOctets^x_{finish}$ are the start and finish counters, respectively, specifying the number of data bytes sent through port *x*. The parameters $ifInOctets^x_{start}$ and $ifInOctets^x_{finish}$ denote the equivalent receive counters. The byte counters in Table 3.1 wrap around about every 5 minutes, when the network load is close to the link capacity. The Controller must thus read these counters at smaller intervals. Whenever all data were transmitted with fixed sized packets, we used the packet counters instead of the byte counters for the computation of the throughput. This avoided intermediate Get-Requests to switches.

Object	Identifier for Port *x*	Description
ifInOctets	1.3.6.1.2.1.2.2.1.10.*x*	Number of data bytes received on port *x*.
ifInUcastPkts	1.3.6.1.2.1.2.2.1.11.*x*	Number of unicast pkts received on port *x*.
ifInNUcastPkts	1.3.6.1.2.1.2.2.1.12.*x*	Number of multicast, broadcast pkts received on *x*.
ifOutOctets	1.3.6.1.2.1.2.2.1.16.*x*	Number of data bytes sent through port *x*.
ifOutUcastPkts	1.3.6.1.2.1.2.2.1.17.*x*	Number of unicast pkts sent through port *x*.
ifOutNUcastPkts	1.3.6.1.2.1.2.2.1.18.*x*	Number of multicast, broadcast pkts sent on *x*.
ifOutDiscards	1.3.6.1.2.1.2.2.1.19.*x*	Number of packet drops on port *x*.

Table 3.1: MIB Counters used for Throughput Measurements.

To ensure a high measurement accuracy, the SNMP Get-Request / Response time should be small in comparison to the measurement interval *t*, because errors are introduced when start and finish control messages are exchanged at significantly different network loads. In a simple test recording 50 requests to a managed hub, the request / response time was in the order of a few 100 microseconds. The measurement interval was at least 30 seconds in all experiments.

To check the accuracy of the measurement approach, two experiments were carried out. In the first, we used a single Traffic Client connected to a single hub 802.12 network. It generated constant bit rate data traffic using fixed sized data packets of 1024 bytes. The data rate was controlled by the Controller which at the same time measured the load on the test network. The measurement results are shown in Figure 3.6. The Traffic Client was an HP C100 workstation. In the test, the data rate was increased from zero up to the maximum network capacity using an incremental step of 1 Mbit/s . The measurement interval was 30 seconds for each data rate.

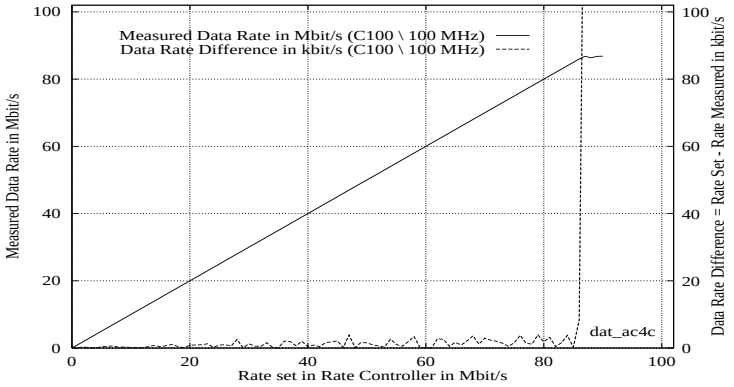


Figure 3.6: Traffic Generator Performance on a HP C100 / 100 MHz.

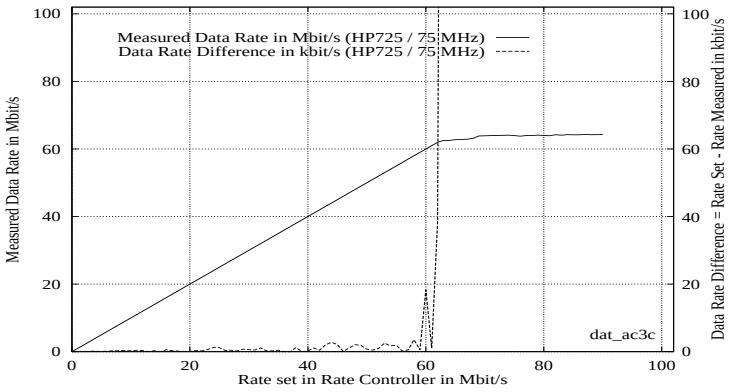


Figure 3.7: Traffic Generator Performance on a HP 725 / 75 MHz.

Figure 3.6 contains two graphs. The first shows the data rate measured by the Controller on the test network. Up to a maximum rate of about 86 Mbit/s, this increases linearly with the data rate configured at the Traffic Client. Data rates above 86 Mbit/s can not be generated with a single C100 in this

setup. The second graph in Figure 3.6 shows the difference between the data rate configured in the Traffic Client and the data rate measured by the Controller. Note that this is given in kbit/s. We can observe that the measurement accuracy is in the order of a few kbit/s until the data rate reaches the performance limit. This range can be viewed as the operational space of the Traffic Client.

Figure 3.7 shows the results for an HP 725 / 75 MHz workstation in the same experiment. The HP 725 was the second workstation type frequently used as Traffic Client in our test network. The basic results are the same as received for the C100. A Traffic Client on a HP 725 workstation however has a smaller operational space. We observed a performance limit of about 62 Mbit/s.

3.6 Measuring End-to-End Delay

3.6.1 A Centralistic Measurement Approach

The link level end-to-end delay can be measured for data packets using the 802.12 high- or normal priority medium access mechanism. Figure 3.8 illustrates our approach for a shared cascaded network whose topology we classify later in Section 4.1. A similar setup was used in switched networks. All delay measurements were taken by the Measurement Client. It had two 802.12 LAN adapter cards, each of them was connected via a separate UTP cable to the corresponding hub. One interface was exclusively used for sending test data packets, the second one was used for receiving. All packets generated by the Measurement Client were addressed to a pre-defined multicast group which was joined with the receive interface. By using the same workstation for sending and receiving test packets, we could use the same clock for determining the start and finish time of each measurement. This used the high resolution counter in CR16.

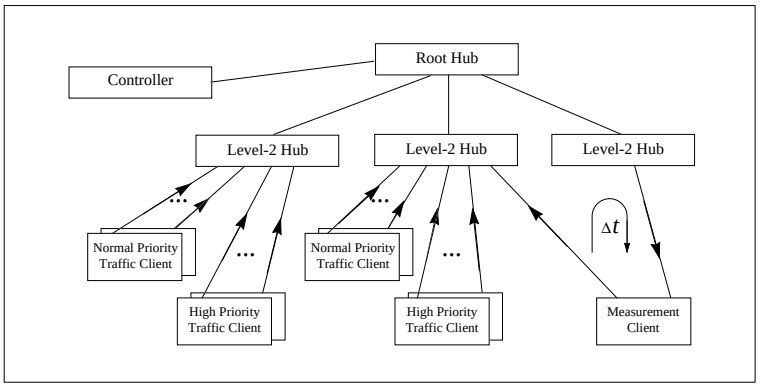


Figure 3.8: Setup for Measuring End-to-End Delay in a shared Network.

Outgoing data packets are time-stamped in the device driver, just before the packet is DMA-ed onto the LAN adapter card. This can not be interrupted. The time-stamp (measurement start time) is carried in the lower four bytes of the source address field in the link level header[1] of the data packet. The arrival time (measurement end time) is also taken in the interrupt service routine. This mechanism is identical to the one carried out by the Traffic Monitor. The measured delay Δt shown in

Figure 3.8 is the link layer end-to-end delay. It includes: (1) the time for transferring the data packet from kernel memory to the sending LAN adapter card, (2) the queueing and propagation delay within the network, (3) the time for transferring the packet from the receiving LAN adapter card back into kernel memory, and (4) the time introduced by the interrupt processing and the context switch. For each packet, Δt is the difference between the packet's finish and start time.

All delay results measured by the Measurement Client are first stored in kernel memory. This is based on a Delay Bucket Table consisting of a number of buckets each of which corresponds to a certain end-to-end delay. Each bucket is used to count the number of data packets received with the delay represented by the bucket. The granularity of the table is 5 microseconds. After the measurement is finished, the table is copied into the user space. The results can then for example be used to compute the distribution density and function.

3.6.2 Accuracy Issues and Alternative Approaches

The strengths of the centralistic measurement approach are its accuracy and its independency of the network load. Furthermore, user processes on the Measurement Client do not impair the measurement results. The same accuracy issues as discussed for the Traffic Monitor apply because the delay measurement approach uses the same mechanism for time stamping. The latency through the relevant send and receive data path can be viewed as a deterministic upper bound for the measurement accuracy. This bound is about: (145 + 25) = 170 μs as reported in Section 6.5.2 in Chapter 6. The average measurement accuracy and the true maximum value is probably however much lower than 170 μs because a large part of the latency will be constant for all data packets. In the experiments in Section 6.5.2, we can observe a maximum variation of about 40 μs in the measurement results, which we believe are caused by overhead variations on the Measurement Client and in the network. The results further show that the packet transmission time for a single maximum sized data packet, which is equivalent to 120 can μs, can clearly be distinguished (see for example the discussion for Figure 6.10).

The main disadvantages of our approach are its costs and its portability. To ensure a high measurement accuracy, source code modifications were required at many places in the kernel. Most of them were specific to the operating system, the 802.12 LAN device driver or the timing register CR16. The solution can thus not easily be ported onto other platforms. Our approach further benefited from the fact that the network entrance and exit points were located close to each other and could be connected to a single workstation. This can typically not be applied in wide area networks.

An alternative is to use two workstations: one for sending test packets, and one for receiving them. Such a distributed approach however always implies timing discrepancies which are typically solved by synchronizing the clocks of the two workstations. This could be based on Global Posi-

1. The most significant byte of the source address field carried the value *0x01* which ensured that our test switches considered the time-stamp as multicast address and thus did not learn every time-stamps as new MAC source address.

tioning System (GPS) receivers connected to the workstations. [Mill94] reports a solution with a time offset of just about 20 - 30 µs between the GPS receiver and the local clock of a Sun machine. The scheme however also requires device driver modifications. A simple approach is to synchronize different workstations by using the NTP protocol which is available on many computer platforms. The results in [Mill94] show that a reliable synchronization with an average of a few hundred microseconds can be achieved on a moderately loaded Ethernet or FDDI network. Note that this is an average value. Temporary clock differences in the order of several millisecond are also reported.

There are two reasons why we decided not to use an NTP based approach: first, NTP's accuracy depends on the properties of the network path, in particular the delay variation, because the synchronization is based on UDP control messages exchanged between the computers to be synchronized. The network load in our test network however often varied substantially. Frequently it was also close to the capacity limit. We believe that the variable packet delay and potential control message losses would have had a negative impact on the accuracy of this approach[1].

Secondly, the synchronization that can be achieved with the standard NTP did not seem to be sufficiently reliable for our purposes. Some of our measurements were performed to test deterministic service guarantees which cover every single data packet within a flow. It would have been difficult to determine whether a particular measurement result was caused by a high queuing delay or just loosely-synchronized workstations. Measurements were also used in this thesis for confirming network performance parameters such as the 802.12 high priority medium access time. These are in the order of 100 microseconds which would have been difficult to measure using the standard NTP. Finally, for a discussion of other synchronization algorithms we refer to [Mill92].

3.7 Measuring the Packet Loss Rate

We used two different approaches to measure the packet loss rate in the network. The first is based on the MIB counters and basically identical to the approach described earlier in Section 3.5, but applied to the packet drop counters of hubs and switches. The counters were retrieved from hubs and switches with the same SNMP Get-Request message as used for the other counters in Table 3.1. This exploited the fact that SNMP permits requests for several MIB objects in a single control message. For half duplex switched links however, two SNMP messages had nevertheless to be sent in order to retrieve the counters from both switches connected to the tested link. Considering this example, we have for the total packet loss rate $loss^l$ on link l:

$$loss^l = (pkt\_drops^l \cdot 100) / (pkt\_drops^l + pkts\_forwarded^l) \quad (3.2)$$

where pkt_drops^l and $pkts\_forwarded^l$ are the total number of packets lost and forwarded on link l, respectively. These parameters can easily be computed using the start and finish values for the

1. In [Mill94], several UNIX kernel modifications are proposed to improve the accuracy of NTP.

counters: *ifOutDiscards*, *ifOutUcastPkts* and *ifOutNUcastPkts* of the relevant switch ports. The same basic accuracy issues as discussed for the throughput measurements in Section 3.5 apply.

The Measurement Client was used whenever the packet loss rate had to be measured for a single flow. In contrast to the MIB based approach used by the Controller, this was based on packet sequence numbers. These were carried by all data packets generated by the Measurement Client. During the experiments, the Measurement Client then recorded the number of data packets discarded in the network as well as the total number of packets successfully sent and received. The packet loss rate then follows directly from the results for these counters.

Chapter 4
Quality of Service under Network Overload

The QoS in packet switching networks which do not reserve resources is hard to predict under conditions of load. In this chapter we study the performance of 802.12 networks in respect to the bandwidth, the packet delay and the packet loss rate encountered by data flows in our test network. This aims at gaining an understanding of the network's link level service capabilities under selected test conditions. The results are further used as reference in later sections of this thesis.

We first introduce a taxonomy for classifying cascaded network topologies. In Section 4.2, we then discuss the traffic traces used throughput the thesis. These include: (1) traces obtained by recording data packets generated by multimedia applications in the test network, and (2) traces computed according to a traffic source model. The characteristics of the application traces are investigated first. This is followed by a description of the source model used to generate the model traffic traces. In Section 4.3, we discuss the 802.12 network behaviour based on measurement results received in test networks with different topologies. Section 4.4 briefly looks at approaches for maintaining QoS in the network. Finally, in Section 4.5, we summarize the important results of this chapter.

4.1 Classifying 802.12 Networks

The support for multi-hub network topologies was introduced into the 802.12 standard to allow enlargements of network size and extension. Figure 4.1 shows potential topologies. Each hub is assigned a *Cascading Level* which marks its position in the shared network hierarchy. The *Root-*, or *Level-1* hub is located at the top of the topology tree. All hubs directly connected to the Root hub are called *Level-2* hubs. These may themselves have many links to network nodes or lower level hubs, which are then denoted *Level-3* hubs, and so on for larger hierarchies. A network node in this context either denotes a host, a bridge or a router. All hubs, except the Root hub, have a single link which connects them to the next upper hub in the hierarchy. This link is called the *Up Link* of the hub. Links connecting lower level hubs or network nodes are called *Down Links*. Each hub may thus have many Down Links but has never more than one Up Link.

The Cascading Level can be used to classify the resulting multi-hub topologies. A *Level-N Cascaded Topology* consists of at least *N* hubs. It always includes one Level-1- and at least one Level-N hub, but never a *Level-(N+1)* hub. The single hub network shown in Figure 4.1 can thus be classified as Level-1 cascaded topology. With a UTP physical layer, cascaded networks with topologies

of up to Level-5 are supported by the standard. The maximum cable length between network nodes and hubs is 200 m in these topologies. Networks with a high cascading level, e.g. Level-4 and Level-5 topologies, are however only required in cases when the physical extension of the network need to be enhanced[1]. Realistic network sizes can already be achieved using Level-2 or Level-3 topologies. A Level-2 topology consisting of 32 x 32 port hubs (1 Root-, and 31 Level-2 hubs) for example could incorporate a maximum of 31 x 31 = 961 nodes. The 32nd port of all Level-2 hubs is the Up-link. This should be sufficient to satisfy any requirement for a single shared network.

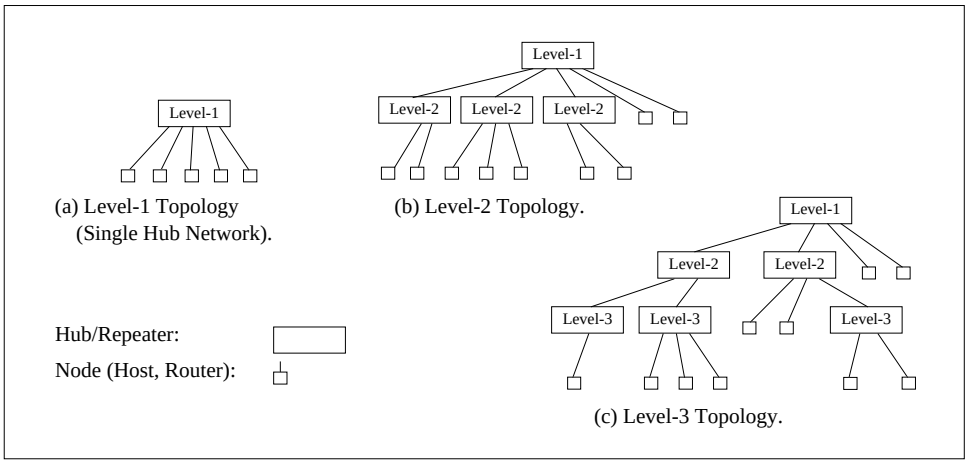


Figure 4.1: Cascaded 802.12 Network Topologies.

Beside cascaded networks, half-duplex switched links are also used in existing 802.12 networks. They also use the Demand Priority protocol to access the physical medium. In contrast to this, full-duplex links work independent of the Demand Priority protocol because the outgoing link is controlled by a sole sender. No contention between different nodes on the network need to be resolved. We thus do not specifically consider full-duplex links in this thesis.

In general however, any switched link can be viewed as a special case of a shared one. Service disciplines which can control performance parameters such as the packet delay in shared networks can typically also be applied to half-duplex and full-duplex switched links. Switched links simplify the network analysis and often exhibit a better performance than shared ones. This is due to the reduced contention when the physical medium is only accessed by two network nodes (half-duplex case), or entirely controlled by a single node (full-duplex case).

In general, we assume a LAN that consists of shared and switched links. Switched links are mainly used in the backbone, between switches, or to connect nodes with large performance requirements such as servers, routers and gateways. Shared segments can typically be found at the workgroup or desktop level to interconnect hosts. The investigations in this thesis mainly focus on shared 802.12 networks as the more general but also the more interesting case in respect to quality of service.

1. The operation of the Demand Priority protocol is also specified across Fiber-Optic links. These allow to bridge distances of up to 2 km between two hubs, or between a host and a hub.

4.2 Traffic Traces and Traffic Models

4.2.1 Application Test Traces

Applications which are most likely to request QoS in an Integrated Services networks are multimedia applications. These were thus of particular interest to us to obtain test traffic traces for our experiments. We recorded data traces for the applications: *vat*[1], *vic*, *Optivision* and *MMC* [McCJ95], [OV96], [Leym96]. All of them used UDP as transport protocol, did not include a congestion control and thus generated traffic patterns which were independent of the network load. This ensured realistic traffic at all data rates in our test network. In contrast, a TCP trace is typically only accurate if used in test networks with similar load conditions that existed when the trace was recorded. In an overloaded network for example, a TCP trace recorded on a lightly loaded network will behave differently to a real TCP flow whose congestion control reacts to the network load.

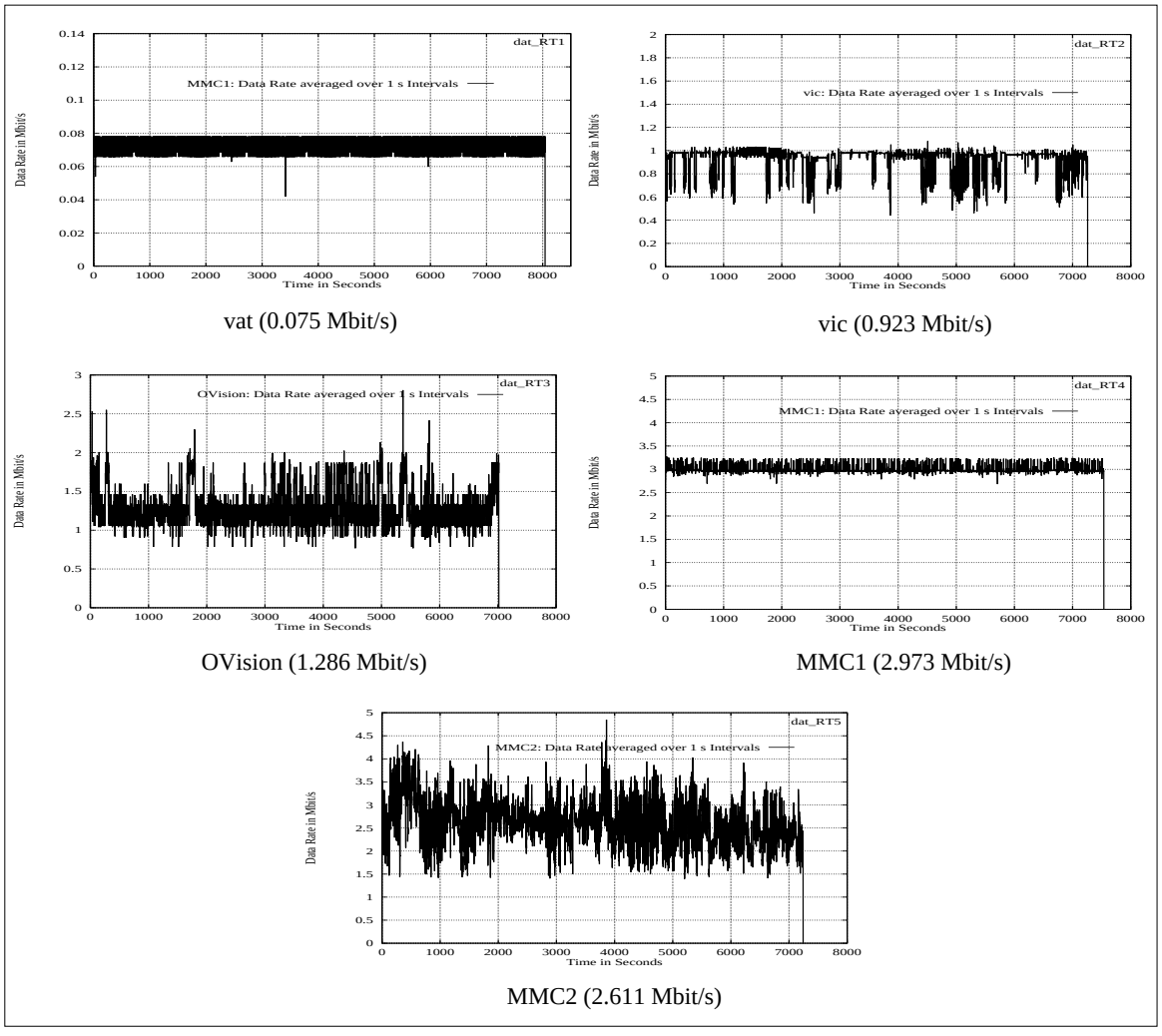


Figure 4.2: The Rate Characteristics of the Application Test Traces.

1. For a description of vat, see: http://www.nrg.ee.lbl.gov/vat/.

All application traces were recorded by the Traffic Monitor on an otherwise empty, single hub 802.12 network. The configurations used for this are described in the following:

1. *vat* is a public domain audio conferencing tool[1]. We used it (version v3.2) on an HP 725 workstation to generate a single audio data stream on the test network. The data source was a TV News audio signal. It was passed to vat via the built-in audio device of the workstation which was connected to the TV audio output. We used vat's default configuration for PCM2 audio encoding. This resulted in an average data rate of about 75 kbit/s at the link layer.

2. *vic* is a public domain video conferencing tool. It was used (version v2.7b2) to generated a JPEG compressed video stream with a data rate of about 1 Mbit/s. Hardware support was given by a Parallax[2] compression card on the HP 725 workstation. The data source was a video camera. We used the following vic specific parameter setting which can be adjusted by the user: *normal picture size* (resolution 368 x 276 pixel), *ordered*, *jpeg*, *22 frames/s*.

3. *OptiVision* is a commercially available communication system supporting audio and MPEG video. It can be used for conferencing or Video-on-Demand within LANs. We recorded a single MPEG encoded video stream with an average data rate of about 1.3 Mbit/s. The video source was a video player playing the adventure movie *Jurassic Park*. The picture resolution was 704 x 480 pixel. 25 frames per second were generated by the system.

4. *MMC* is a high quality conferencing system supporting voice, video and application sharing. We used version v4.0 to generate JPEG compressed video data streams of about 3 Mbit/s on the test network. This was based on the same hardware as used for *vic*. The size of the video was 720 x 540 pixel. About 11 frames per second were generated. We recorded two different MMC traces which we called *MMC1* and *MMC2*. These differed by the nature of the video signal passed into MMC. For trace MMC1, we connected a video camera to the workstation's Parallax card. It was directed into the Lab capturing busy people at some distance. The data source for trace MMC2 was a TV Sportshow. For this we connected the TV video output to the Parallax card.

In all experiments, we recorded the application output for about 2 hours. Figure 4.2 shows the complete traces. To characterize them, two important traffic descriptors can be identified: the average data rate and the burstiness. The average data rate and other basic trace characteristics are given in Table 4.1. To estimate the burstiness, we used two methods: (1) the maximum peak to average bandwidth ratio over different time scales, and (2) the Variance-Time plot (see for example [GaWi94], [LTWW94]). The results for both are discussed in the following.

Figure 4.3 shows the results for the peak to average bandwidth ratio. These are computed over time intervals I ranging from 5 µs to 5 s. For each interval, we determined the maximum data rate over any interval I within the trace by applying a sliding window. The final result was then normalized

1. *Vat* and *vic* are publicly available as part of the *Mbone* Tools from: http://www-nrg.ee.lbl.gov/.
2. For informations, see Parallax Graphics, *PowerVideo700 Board*, (http://www.parallax.com/products/hp/xvideo700.html).

using the average data rate in Table 4.1. For comparison, we further added the result for a trace of the adventure movie *Star Wars*[1] because the characteristics of this trace were analysed in detail by Garret and Willinger in [GaWi94].

Source Number	Source Name	Encoding Scheme	Total Number of Bytes	Total Number of Packets	Total Trace Length in Minutes	Average Data Rate in Mbit/s
1	vat	PCM2 Audio	75299414	202451	134.090	0.075
2	vic	JPEG Video	837984422	893857	121.027	0.923
3	OVision	MPEG-1 Video	1128571004	844438	116.989	1.286
4	MMC1	JPEG Video	2802731665	2078674	125.678	2.973
5	MMC2	JPEG Video	2365988081	1722701	120.813	2.611

Table 4.1: Basic Application Trace Characteristics.

The basic characteristics in Figure 4.3 are similar for all traces. We find high peak rates over short time intervals. For our own traces we can observe maximum bandwidth ratios of 24 to 29 over time intervals of 10 ms. We believe that these are mainly caused by the traffic control mechanism used in the applications: MMC for example grabs an entire video frame from the JPEG compression card and passes it to the network as one unit. Since the workstation can send data at line speed (100 Mbit/s), the video frame fragmented into several data packets, appears almost as a single traffic burst on the network. For time intervals smaller than 50 ms, the results for the Star-Wars trace are significantly lower. The trace is however a computed coding result and not a measurement result from a real application.

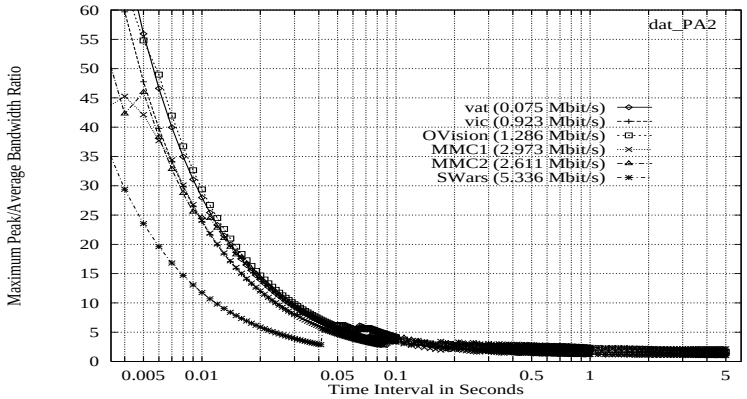


Figure 4.3: Peak to Average Bandwidth Ratio's for the Application Traces in Table 4.1.

For longer time intervals the peak to average ratios then decrease quickly. For all traces, we find a result of less than 5 for time intervals longer than 100 ms. The instantaneous steps in the graphs are caused by the ON/OFF behaviour of the data sources. Since our Pareto sources do also exhibit this behaviour, but more significantly, this is discussed in the next section.

1. The *Star Wars* movie was encoded using the JPEG compression standard and has an average data rate of 5.336 Mbit/s. The entire 2 hour trace is publicly available at: *ftp://ftp.bellcore.com/pub/vbr.video.trace/*

Even though the graphs in Figure 4.3 are useful to quickly estimate the peak data rate over different time scales, the results are determined by a single value: the maximum peak rate observed for a particular time interval over the entire trace. To explore other properties such as the variance within each time interval, we computed the Variance-Time plot for all our own traces and the Star-Wars trace. More specifically, this aimed at: (1) a comparison of all traces based on their variance in different time intervals, and (2) an estimation of the degree of self-similarity of each trace. The latter was motivated by research results on traffic analysis which showed that network traffic may exhibit self-similar or fractal-like characteristics [LTWW94], [GaWi94], [WTSW95], [PaFl95]. This was based on the observation that correlations between packet arrivals are extremely long-lived, with the implication that burstiness occurs over much longer time intervals than previously considered.

To describe self-similarity more precisely, we follow [LTWW94]: let $X = (X_t: t = 0, 1, 2, ...)$ be a stationary process (e.g. an application data trace without rate shifts) with the autocorrelation function $r(k)$, where $k \geq 0$. Further, let $X^{(m)}$, $m = 1, 2, 3, ...,$ be the stationary time series obtained by averaging the original series X over non-overlapping time blocks of size m. The autocorrelation function corresponding to $X^{(m)}$ is denoted by $r^{(m)}(k)$, where $k \geq 0$. Process X is exactly or asymptotically second-order self-similar if the corresponding aggregated processes $X^{(m)}$ are the same as X or have the same autocorrelation function as X [LTWW94]. More formally: $r^{(m)}(k) \rightarrow r(k)$, as $m \rightarrow \infty$. Two important characteristics are exhibited [LTWW94]: (1) the autocorrelations $r(k)$ decay hyperbolically fast (i.e. as $r(k) \sim k^{-\beta}$, as $k \rightarrow \infty$ and with $0 < \beta < 1$) rather than negative exponentially fast (i.e. as $r(k) \sim a^k$, as $k \rightarrow \infty$ with $0 < a < 1$) implying a non-summable autocorrelation function $\sum_k r(k) = \infty$. Secondly (2), the variances of the sample mean $X^{(m)}$ decrease proportional to $Var(X^{(m)}) \sim m^{-\beta}$, as $m \rightarrow \infty$, and with $0 < \beta < 1$.

The degree of self-similarity is quantified using the *Hurst* parameter *H* which is related to the decay β of the autocorrelation coefficients by: $H = 1 - \beta/2$. The Variance-Time plot is a graphical method for estimating *H*. It is obtained by plotting the variances $Var(X^{(m)})$ versus the block size *m* ("the time") in log-log coordinates. The slope β of the resulting graph, as $m \rightarrow \infty$, is estimated using a least squares regression, which should ignore the results for small *m*. Estimations between -1 and 0 suggest self-similarity. This corresponds to: $0.5 < H < 1$ where the degree of self-similarity and thus the degree of the burstiness (long range dependence) increases for larger *H*-values. A slope of -1 ($H = 0.5$) or smaller values than this, indicate burstiness occurring only over short time intervals (short range dependence).

The Variance-Time plots for all traces are shown in Figure 4.4, Figure 4.5 and Figure 4.6. We used block sizes *m* ranging from 1 to 20000 with an incremental step of 1. A single block corresponds to 100 ms. The computation of each plot was thus based on at least 70000 - 100 ms samples.

The results for the traces: StarWars, MMC2, OVision, vic and MMC1 are shown in Figure 4.4. We ordered them according to their maximum variances. Note that both coordinates are logarithmic, but provide absolute values. For StarWars, we found an estimate for *H* of about 0.74 in the interval $[400, 10000]$. This is close to the result of 0.78 reported in [GaWi94] for this trace. The authors

unfortunately however neither specified the estimation interval nor the time corresponding to a single time block. For the MMC2 trace, we estimated a Hurst parameter of about 0.84 over the interval $[100, 10000]$, which suggests that this trace is: (1) self-similar, and (2) burstier than the StarWars trace whose slope decays faster.

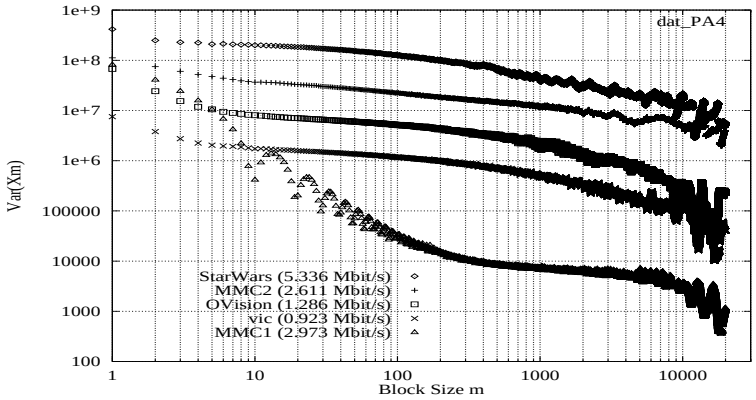


Figure 4.4: Variance-Time Plot for Application Traces (a).

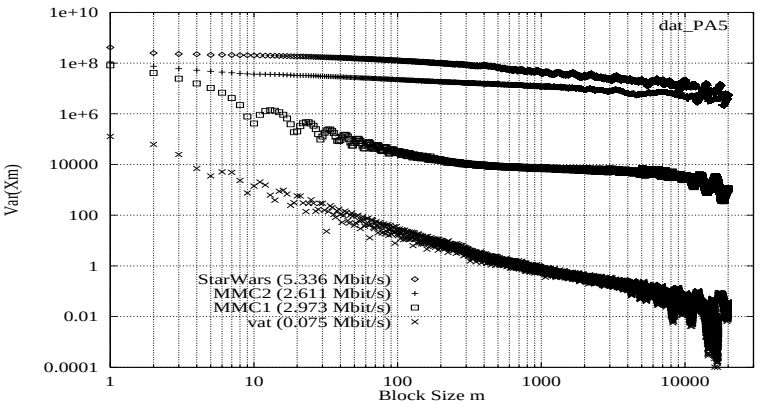


Figure 4.5: Variance-Time Plot for Application Traces (b).

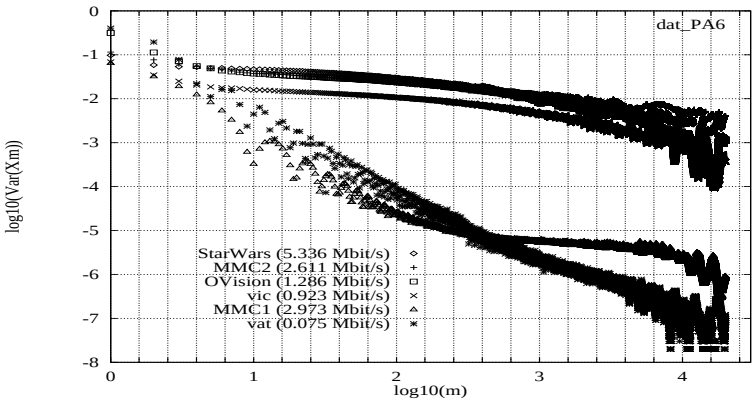


Figure 4.6: Variance-Time Plot for Application Traces (c).

Finding a reliable estimation for the OVision and the vic trace is difficult because the slope of both graphs does not become stable. The results in Figure 4.4 however suggest that both traces are less bursty than the StarWars and the MMC2 traces. In all graphs, we can observe that the computed variances become more and more unreliable for large block sizes ($m > 10000$). This is caused by the smaller number of data blocks that are used in the statistical analysis. The result for $m = 10000$ (1000 seconds) for example is only based on 7 samples due to the trace length of just 2 hours.

Rather unexpected for us was the shape of the curve received for the MMC1 trace. For small block sizes ($m < 200$) the variance decreases rapidly, but then remains almost constant with a slope of -0.256 ($H \approx 0.87$) over the interval $[200, 7000]$. Based on this result, one might assume self similarity and high burstiness, but a look at the MMC1 trace in Figure 4.2 shows that this is not the case. Instead, we believe that this behaviour is due to noise, because the absolute values for the variance are extremely small (< 10000) and correspond to an average data variation of only about 100 bytes between different samples over time scales of more that 20 seconds.

Figure 4.5 shows the result for the vat trace (75 kbit/s). For comparison we added the graphs for StarWars, MMC2 and MMC1. It can instantly be observed that vat did not generate traffic bursts over long time scales. We estimated a slope of about -1.09 ($H \approx 0.45$) within the interval $[300, 5000]$ which confirms the short range burst behaviour expected for this trace.

Finally we plotted the normalized results for all traces in Figure 4.6. They were computed by normalising the data in each 100 ms time block with the average over all blocks in the trace, creating a data-rate independent result for each trace. We find that the variances computed for the StarWars, MMC2, OVision traces are in the same order of magnitude, although the slope for the OVision trace decreases faster. The vic trace is less bursty which is however not surprising considering the corresponding graph in Figure 4.2. The MMC1 and vat traces exhibit a similar behaviour for block sizes smaller than 2.3 in the logarithmic scale. This occurred despite that the average data rates of these traces differ significantly.

In general, we found that estimating the Hurst parameter H is difficult. Estimates depend significantly on the time interval used for the least squares regression. A stable slope can further not always be clearly identified. Longer traces might provide more samples and thus increase the accuracy, but often also contain rate shifts which may distort the results. We thus found a visual inspection and the relative comparison of all traces with a well known reference such as the Start Wars trace essential.

4.2.2 Source Model Traces and Parameter Selection

Modelling data traffic is a hard problem because LAN traffic is complex and may depend on the user's behaviour, the application, and the network. The goal of a traffic analysis is a model which accurately reflects the traffic characteristics but is also mathematically tractable. Traditionally traffic models based on exponential or geometric distributions typically only exhibit burstiness over short time intervals. When applied to modelling real network traffic implying self-similar characteristics,

their use may cause an over-optimistic estimation of the network's performance. The results of the analysis in [LTWW94], [WTSW95] strongly suggest that LAN traffic is more accurately modelled using heavy-tailed distributions with infinite variance. This is because these distributions generate events over a wide range of time scales.

For comparison, we thus used two "artificial" traffic sources with infinite variance distributions for generating test traces. Following [WTSW95] this was based on an ON/OFF source model with Pareto distributed ON times and Pareto distributed OFF times. We used the name POO model for this. In [WTSW95], it is shown that the superposition of many POO sources whose ON and OFF periods exhibit infinitive variance, produces, on large time scales, network traffic that is self-similar. The cumulative probability function of the Pareto distribution (see for example [WTSW95] or [PaFl95 - Appendix B] and the references therein) is given by:

$$F(x) = 1 - \left(\frac{a}{x}\right)^{\beta}, \qquad a, \beta > 0;\ x \geq a \tag{4.1}$$

where β is the shape parameter and a is the location parameter describing the characteristics of the distribution. A shape parameter of $\beta < 2$ results in a heavy-tailed distribution that has infinite variance, a shape parameter of $\beta \leq 1$ provides a distribution with infinite mean. The location parameter a is given by: $a = r \cdot (\beta - 1)/\beta$, where r denotes the mean of the distribution[1]. The relation between the shape parameter β and the Hurst parameter H of the aggregate traffic is [WTSW95]: $H = (3 - \beta)/2$. Furthermore, traditional traffic models can be viewed as special cases of the self-similar approach when these are used with a shape parameter bigger than 2.0 [WTSW95].

The β-estimates for LAN traffic in [WTSW95] suggest different β values for the ON and OFF-periods in the POO model, where higher results were found for ON than for OFF. For data traffic, the authors observed values of about 2.0 (on the borderline between finite and infinite variance) for ON-periods, and values around 1.0 and 1.5 for the OFF-period. Values exceeding 2.0 for ON-, and close to 1.0 for the OFF-periods are suggested for Mbone [Erik94] traffic.

Source Number	Source Name	Packet Size (fixed) in bytes	Average Per-Flow Data Rate in Mbit/s	Model Parameters					
				Peak Rate in Packets/s	I in ms	N in Packets	Peak-to-Average Rate Ratio	β ON-Period	β OFF-Period
6	POO1	1280	0.321	64	325	20	2	1.2	1.1
7	POO3	1280	0.262	256	360	10	10	1.9	1.1

Table 4.2: Pareto Source Characteristics.

1. In our implementation, we generate Pareto distributed random variables x from a uniform distribution. This is based on the inverse of Equation 4.1, for which we have: $x = a/((1 - F(x))^{(1/\beta)})$. Now, if $F(x)$ is uniformly distributed in the interval *[0.0 , 1.0)* , then we receive Pareto distributed values for x. A uniformly distributed variable however can easily be generated, for example by exploiting the UNIX function *erand48()*.

Table 4.2 shows the characteristics of the two Pareto sources used in this thesis. The source model parameters are identical to those of the sources *POO1* and *POO3* in [JSD97] (see Table 1 therein). We however used fixed sized packets of 1280 byte length. Following [JSD97], the average packet generation rate r_{ave} is computed by:

$$\frac{1}{r_{ave}} = \frac{I}{N} + \frac{1}{r_{peak}} \tag{4.2}$$

where *N* is the average of the random, Pareto distributed number of data packets generated at fixed peak rate r_{peak} during each ON-period. The parameter *I* denotes the average of the Pareto distributed OFF-period. Note that multiplexed Pareto traces were computed using different instantiations of the particular source model. This differs from the method used for applications traces in which multiple-flow traces were generated from a single source trace.

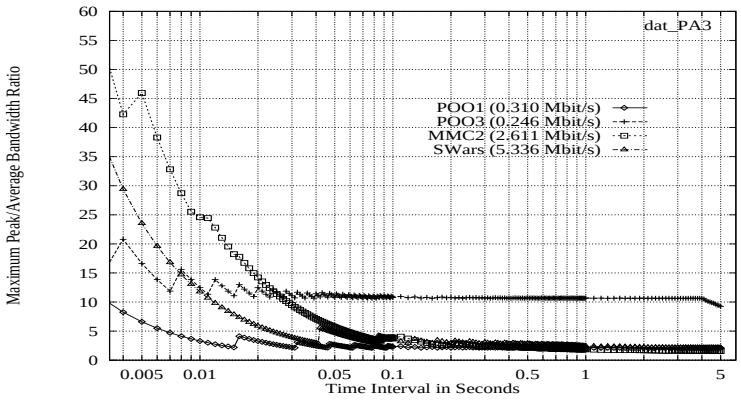
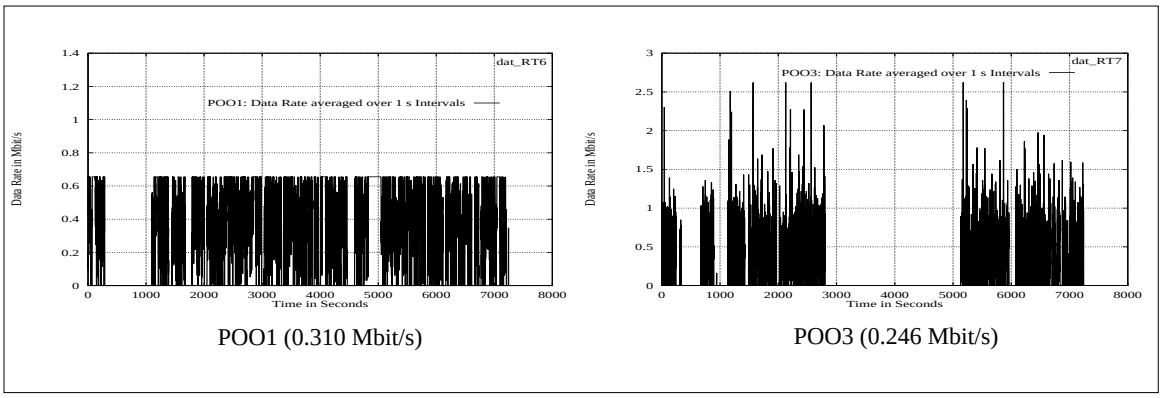


Figure 4.7: Peak to Average Bandwidth Ratio for the Pareto Sources in Table 4.2.

To illustrate the difference to the application traces, Figure 4.7 shows the equivalent results to Figure 4.3 for the two Pareto flows. The graphs for the MMC2 and the StarWars traces were further added. We found that both POO flows maintain the peak to average ratio over significantly longer time scales. The result for the POO3 trace only decreases for time intervals larger than 4.5 seconds. The ratios for both flows are however slightly higher than specified in Table 4.2. The reason for this is the infinite variance of the distribution and the rather short trace length of 2 hours over which we averaged the data rate. This resulted in a slightly lower average data rate for both POO instantiations and thus a higher peak-to-average ratio.

The instantaneous steps in the graphs are caused by the ON/OFF behaviour of the (single) sources and the sliding window technique applied in the computation. Sometimes a burst just fitted into the averaging time interval. For larger intervals, the following OFF period then decreased the peak to average ratio. As soon as the interval however accommodated the following burst, the ratio increased again.

Finally, Figure 4.8 shows examples for the two Pareto source models. We deliberately chose an instantiation with a long silence period (POO3: [2808, 5133]) to illustrate the impact that the low shape parameter of $\beta = 1.1$ may have on the OFF time even for a relatively short trace of 2 hours.



POO1 (0.310 Mbit/s) POO3 (0.246 Mbit/s)

Figure 4.8: Rate Characteristics of two Test Flows generated according to the Pareto Source Models POO1 and POO3.

4.3 802.12 Network Overload Behaviour

4.3.1 Available Bandwidth in Cascaded Network Topologies

The network parameter that is typically most important for the user, is the network bandwidth. In LAN's, this parameter is however often not constant, but may depend on: (1) the network topology, and (2) the packet-size used for the data transmission. To investigate these dependencies in 802.12 networks, we measured the maximum throughput for different packet sizes across different cascaded test networks. The experimental setup for this and the results are outlined in the following.

All experiments were based on the worst-case network setup identified for the particular test topology. The performance of the single hub network as shown in Figure 4.1 (a) was investigated first. For this we used seven Traffic Clients to generate data traffic with a packet size ranging from 64 bytes to 1500 bytes. All traffic was multicast in conformance with the worst-case packet transmission model which we describe in detail later in Section 5.2.1 in Chapter 5. The Controller measured the throughput and controlled the packet sizes used by the Traffic Clients. The former was based on the method introduced in Section 3.5. The link between each Traffic Client and the hub consisted of a 100 m Category 3 UTP cable. The Controller was connected via a 5 m cable of the same type. For each packet size, we measured the throughput for 30 seconds. The incremental step of the packet size was 4 bytes.

After determining the maximum throughput for the single hub test network, the experiment was repeated in a Level-2, Level-3 and Level-4 cascaded test network. This used the same measurement setup and the same UTP cabling, but only three Traffic Clients[1]. The Level-2 topology consisted of

one Root hub and three Level-2 hubs. Each Traffic Client was connected to one of the three Level-2 hubs, which themselves were then linked to the Root hub. The Level-3 and Level-4 cascaded topologies differed from the Level-2 network by three additional Level-3 and Level-4 hubs, respectively. These were inserted between the Traffic Clients and the hubs with the so far highest cascading level. The Level-4 topology thus consisted of 10 hubs: one Root hub and three hubs for each higher level. All hubs, apart from the Root hub, had only one Up-Link and one Down-Link, creating a symmetric topology tree with the Root hub as the only branch point. Each of the three Traffic Clients was always connected to a different hub located at the leaves of the hierarchy. To determine the throughput on the LAN, the Controller only read the MIB counters from the Root hub. This was sufficient because data packets are forwarded to all hubs in the cascaded networks. The measurement results for all four topologies are shown in Figure 4.9. Repeating the measurements showed throughput differences in the order of a few kbit/s. We thus omitted error bars since these could not have been identified in Figure 4.9.

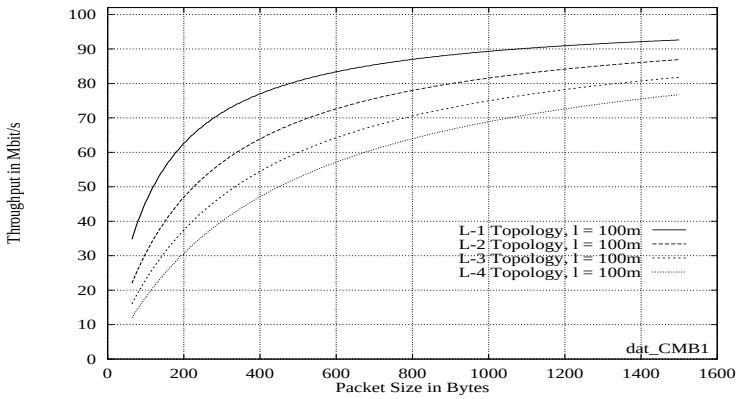


Figure 4.9: Measured Worst-Case Throughput in Cascaded 802.12 Networks using a UTP Physical Layer.

Let us first look at the graph measured for the single hub network (L-1 topology): the achievable data throughput varies for different packet sizes and becomes significantly smaller for data transmissions that only use small sized packets. We measured a maximum of 92.76 Mbit/s for 1500 byte data packets and just 35.13 Mbit/s when 64 byte packets were used: a performance loss of over 60%.

This dependency is caused by the nature of the packet transmission in Demand Priority networks. To transmit data packets across cascaded topologies, network nodes and hubs communicate with each other and synchronize their actions by exchanging 802.12 link control signals. These are used to signal the local MAC status and to control the physical medium access in the shared network. Both consumes time. Each packet transmission is therefore associated with a Demand Priority protocol and signalling overhead, which itself is however independent of the actual packet size. We

1. The same measurement results could actually be achieved with just two Traffic Clients provided that these are sufficiently powerful to overload the network.

thus find that a data transmission using large packets achieves a significantly higher throughput than one that uses small packets because the latter implies a larger total transmission overhead.

The network topology can have an impact on the performance due to the extensive signalling required in large multi-hub topologies to synchronize the medium access. The worst-case results in Figure 4.9 show that the network throughput may significantly decrease in higher cascaded topologies. One can observe a maximum performance difference of over 30 Mbit/s in the graphs for the Level-1 and the Level-4 test topology. These are the costs for having: (1) a larger network size, (2) a wider physical network extension, and (3) a controlled medium access for all network nodes.

As in the single hub network, the throughput further decreases in all test topologies when only small sized packets become used for the data transmission. The maximum throughput measured for example in the Level-4 network for data packets of 100 bytes is as low as 17.93 Mbit/s. For 1500 bytes, we measured 76.75 Mbit/s.

Note that all results in Figure 4.9 were achieved in a worst-case setup that included: (1) Traffic Clients located only at the leaves of the topology tree, and (2) data packets transmitted using multicast. Both maximized the signalling overhead which we will analyse later in Chapter 5. In realistic networks however, unicast and multicast are used. Servers and bridges are typically directly connected to the Root hub. This reduces the overhead. Hubs can further serve requests from several hosts before passing on the network control, which further decreases the signalling requirements. In real networks, we will therefore on average observe a much higher network performance than shown in Figure 4.9 for example for the Level-4 topology.

The importance of multicast traffic in todays LAN's is hard to evaluate and seems to depend much on the special case. For example, only a few percent of the total traffic currently (1998) forwarded within Hewlett-Packards corporate Intranet is multicast. In contrast to this, the analysis in [WTSW95] reports over 50% Mbone traffic for traffic traces taken at Bellcore in 1994. This makes an evaluation of the difference between the worst case and the reality more difficult.

4.3.2 Available Bandwidth in Switched Networks

We next investigated the available bandwidth in half-duplex switched networks. For this, we measured the throughput across a single UTP link between two standard 802.12 switches. The test was based on the same fundamental measurement method as described for the cascaded topologies. We however only used two Traffic Clients, each of which sent multicast traffic across the test link. Both Traffic Clients were connected to one of the two switches via a 5 m UTP cable. The test link had a length of 100 m. To measure the throughput, our Measurement Controller read the number of sent and received packets from the switch it was connected to. The setup and the results are shown in Figure 4.10. For comparison, we also added the result for the single hub network from Figure 4.9.

As expected, the half-duplex link exhibits similar characteristics as observed in cascaded networks. This is because the physical medium is still shared between the two switches. Even though the max-

imum throughput is slightly higher than the result measured for the single hub network, it also degrades when small sized data packets become used.

The higher throughput across a switched link can be explained with the two different 802.12 operational modes built into the MAC chips of the switches in the test network. As other 802 style standards, 802.12 differentiates between hosts and hubs. The functional control requirements for hosts are defined in the MAC protocol. The equivalent requirements for hubs are specified in the Repeater-MAC (RMAC) protocol. Switch ports can typically operate in “host-mode” (802.12 MAC) when connected to a hub, or in “hub-mode” (802.12 RMAC) when connected to a host. In the case that two switches are connected to each other, one of them operates in MAC, the other in RMAC mode. This leads to a short data path which only includes one MAC, one RMAC and one UTP link. In the single hub network however, the data path consists of the elements: MAC-RMAC-MAC and two UTP links, which results in the lower throughput observed in Figure 4.10.

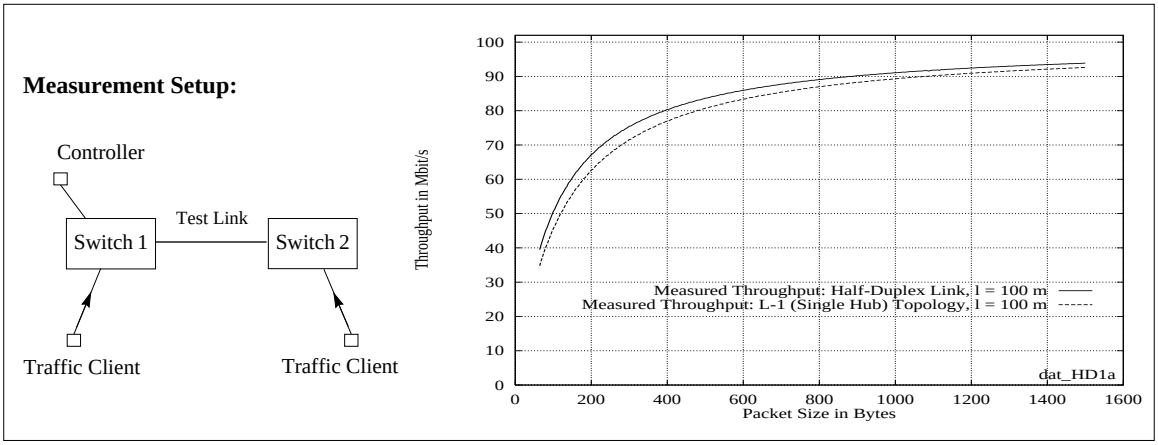


Figure 4.10: Measured Worst-Case Throughput for a half-duplex switched Link using a UTP Physical Layer.

It remains to remark that a dependency between throughput and packet size can also be observed for other 802 style LANs, although not to the same extent as in 802.12. At the moment, the most widely deployed LAN technology is 802.3 Ethernet. For the shared medium 802.3 version, no clear worst case can be given because the medium access is unbounded. For a full-duplex, 100 Mbit/s Ethernet link however, we measured a maximum throughput of 76.19 Mbit/s for 64 byte packets, 83.33 Mbit/s for 100 bytes and 98.68 Mbit/s for 1500 byte packets in a single direction. This was performed in a similar experiment as described for 802.12.

4.3.3 Network Delay and Loss Characteristics

The per-packet delay observed in packet switching networks can be split into two basic components: a fixed part and a variable part. The fixed part is caused by the constant delay through the sublayers of the 802.12 transport-stack such as the MAC or the PMD (see Section 5.1 for them). Our analysis in Section 5.2 and Section 5.3 show that for 802.12 devices the delay introduced by each of

these sub-layers is only in the order of a few nano- or microseconds. In addition to the fixed network delay, there is a variable amount of delay encountered by data packets in the network. This is mainly caused by the queuing delay within network devices. A maximum delay of 120 microseconds, which is the transmission time of a maximum sized data packet over a 100 Mbit/s link, can additionally be introduced by 802.12 switches or routers operating according to the store-and-forward approach. Devices using this technique first wait for the entire data packet to arrive before executing any further packet processing. In contrast, *Cut-Through* switching devices such as 802.12 hubs are typically able to avoid this delay by starting the packet transmission before the data packet has actually been fully received.

A major part of the end-to-end delay that can actively be controlled by using admission control is the queuing delay. For network devices such as LAN switches, the queuing delay depends on the burstiness of the arriving data traffic, the buffer capacity of the switch, the arrival- and the service data rate. To investigate the basic characteristics, we first measured the packet delay and the packet loss rate versus the network load across a half-duplex switched link.

The test network was similar to the one shown in Figure 4.10. We however used eight Traffic Clients connected to *Switch 1* to generate multicast cross traffic. The experiment was based on the traffic trace driven approach described in Chapter 3. For measuring the end-to-end delay, we linked a Measurement Client to the test network such that it could send data packets to Switch 1 and received them from *Switch 2* after their transmission across the network. Using static filter entries in both switches ensured that cross traffic: (1) was only forwarded onto the test link, and (2) left Switch 2 through a different output port than the one connected to the Measurement Client. This avoided any interference between cross and measurement traffic other than on the output port of Switch 1 to Switch 2; but required a different multicast address for measurement traffic sent by the Measurement Client. The test link had a length of 100 m. For all other links in the test network we used 5 m UTP cables. Note that all Traffic Clients and the Measurement Client only used the 802.12 normal priority medium access.

Using this setup, we performed four different measurements. These were based on traces generated from (1) the application traces MMC2 and OVision, and (2) the traffic source models POO1 and POO3. In each measurement we only used *homogeneous* flows produced from the same application trace or the same source model. The Measurement Client always injected a single data flow into the test network and measured the delay and loss rate for the corresponding data packets. For each measurement point within a test, the MClient further used the same start-offset into the trace to ensure the same measurement conditions. The cross traffic varied from zero up to a total load of about 90 Mbit/s. It was increased with incremental steps of about 10 Mbit/s. For this, each Traffic Client sent packets equivalent to a number of homogeneous flows into the network. The required trace files were pre-computed. The measurement interval for a single measurement point was 30 minutes with an additional warm-up time of 2 minutes. The Controller additionally recorded the average network load on the test link and the total packet loss rate at the output of Switch 1.

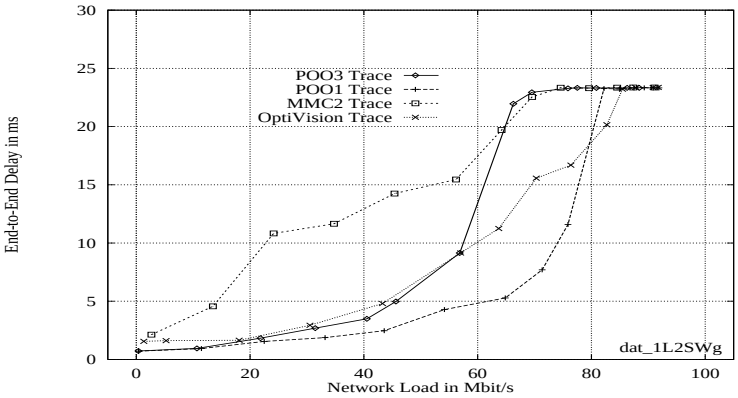


Figure 4.11: Maximum Packet Delay for different Flow Types in Dependence of the Network Load.

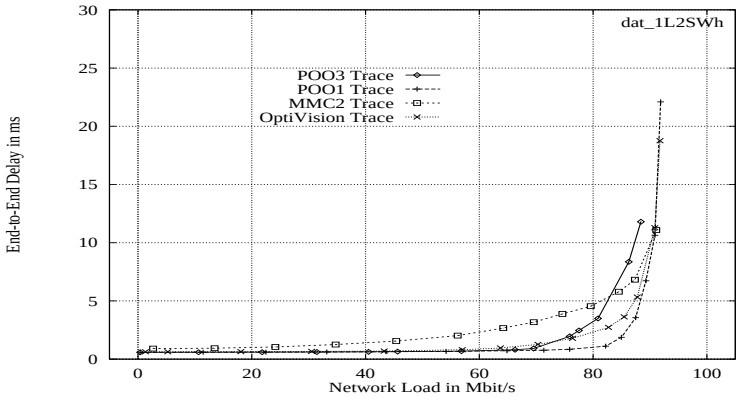


Figure 4.12: Average Packet Delay for different Flow Types in Dependence of the Network Load.

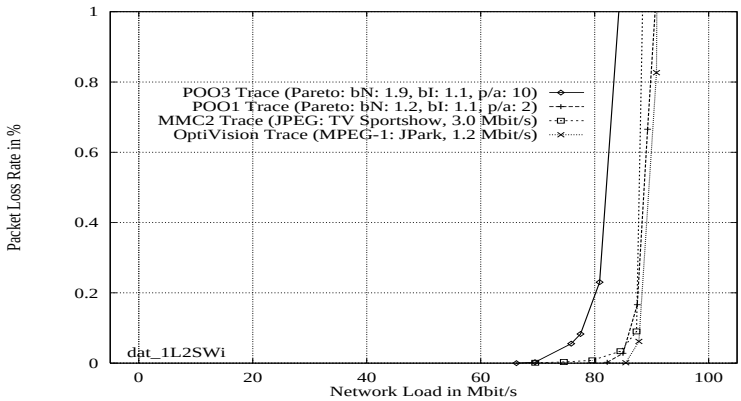


Figure 4.13: Packet Loss Rate for different Flow Types in Dependence of the Network Load.

Figure 4.11, Figure 4.12 and Figure 4.13 show the measurement results measured by the Measurement Client for the four different data sources. These include: (1) the maximum end-to-end packet delay, (2) the average end-to-end delay, and (3) the packet loss rate. For all test sources, we can more or less observe a certain threshold in the load-delay and load-loss curves: the results for the delay and the loss rate are low as long as the network load stays below the threshold. As soon as the network utilization however exceeds the threshold, delay and loss increase significantly faster. This is a typical behaviour and could be expected (see for example [Shen95]). For the average delay, the threshold is basically in the same range for all test traces. This is close to the maximum link capacity.

The maximum delay is determined by the burstiness of the traffic and the network load. Large maximum delays can thus be observed much earlier, but are limited by a bound of about 23 ms which corresponds to 256 kbytes of output buffer space used in Switch 1. As soon as the maximum delay reaches this bound, the output queue is full and packet loss occurs as can be observed in Figure 4.11 and Figure 4.13.

From all four test sources, POO3 and MMC2 exhibited the worst behaviour in respect to packet delay and loss rate. This could be expected considering their traffic characteristics discussed in the previous section. For the POO3 source, the first loss ($0.772 \cdot 10^{-4}$ %) occurred at a network load of 66.27 Mbit/s, which corresponds to 196 active POO3 sources. For MMC2, we measured a loss rate of 0.0011 % for 69.57 Mbit/s, or 26 active MMC2 sources. The packet loss rate for POO3 further increases significantly earlier than observed for any of the other test source. This can be explained by the extreme burstiness of this source.

More unexpected for us were the results for the average load because these almost stay constant over a load range of over 60 Mbit/s. Even for low loss rates smaller than 0.1%, the average delay for all test sources remains in the order of a few milliseconds, typically below 10 ms. From this, two simple conclusions can be drawn:

1. If the network administrator can ensure that the network is always operating below the load-threshold, then resource reservation is probably not required unless an application has guaranteed service constraints and cannot adapt.
   The appropriate maximum network load for an application is however difficult to determine because it depends on the QoS requirements of the particular application but also on the characteristics of the cross traffic on the network. In our specific test setup, the network for example could support 24 MMC2 JPEG video flows without packet loss and with a low average delay. This corresponds to a network load of about 63 Mbit/s.

2. Since the average delay does not significantly increase with the network load, there seem to be little gain in supporting several higher priority levels to differentiate service classes with a different average delay bound within 802.12 switches. Even when several classes were implemented, these would provide an average delay which would be hard to distinguish for existing real-time applications. This assumes that advanced LAN services will be operated at a maximum network

load far below the point at which the first packet loss occurs. The network might still be temporarily overloaded due to the best effort traffic which is not regulated in any way and typically forwarded at the lowest priority.

4.3.4 Impact of the Amount of Buffer Space within Switches

For switch designers, it is very desirable to reduce the amount of memory required for packet buffering within switches. This is because the costs for memory, even though much reduced in the past few years, are still significant given the total costs of LAN switches and the price competitive market. For the switches used in our test network for example, almost half of the costs for electronic parts were required for the port memory. Given the desire to reduce costs, we investigated the impact of the amount of buffer space within switches on the delay and loss characteristics. This used the same test network and the same setup as described in the previous section.

We performed six experiments based on MMC2 and POO3 test sources. In the first, we loaded the network with 24 MMC2 video flows: 23 were generated by the Traffic Clients and 1 was generated by the Measurement Client. This again only used the normal priority medium access. Switch 1 had an output buffer of just 16 kbytes for each of its ports. The Controller measured the packet loss rate of the a*ggregated* traffic at the output port from Switch 1 to Switch 2. At the same time, the Measurement Client recorded the delay and loss characteristics for all data packets of the single MMC2 flow it generated. The measurement interval for this was 30 minutes with an additional 2 minute warm-up before the data recording. After the measurement, we increased the buffer size in Switch 1 and repeated the experiment using the same setup but a larger buffer space[1].

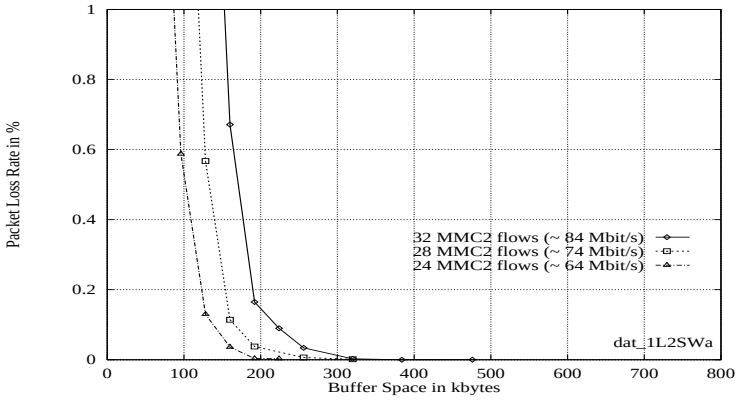


Figure 4.14: The Packet Loss Rate for different Sets of MMC2 Flows in Dependence of the Buffer Space in the Switch.

1. The output buffer space of the 802.12 port modules in the Switch 2000 is statically defined, but can be reprogrammed. Any changes however require modifications to the switch kernel. We thus built a number of kernels, each of them supporting a different output buffer size ranging from 16 kbytes to a maximum of 768 kbytes. Each of these kernels was then used to measure a different point in the loss - buffer space curve. Our prototype port module itself had a physical memory of 1 Mbyte.

After measuring the delay and loss characteristics for the entire range, we performed the same experiment with 28 and then with 32 MMC2 flows on the test network. To ensure the same traffic conditions, all Traffic Clients and the Measurement Client used a fixed start-offset into the trace. The value of the offset itself however varied for all of them. Figure 4.14 shows the results for the total packet loss rate observed by the Controller for different buffer sizes in Switch 1. For all three sets, a minimum buffer space of far less than 200 kbytes prevents packet loss rates larger than 1%. As expected, the slope of the loss-curves however becomes flatter such that significantly more memory is required to completely eliminate the packet loss in Switch 1. For 32 MMC2 flows (the upper curve in Figure 4.14), we still observed a loss rate of $0.196 \cdot 10^{-3}$ % when using a buffer space which was more than twice as large: 476 bytes. For 28 flows, we measured a loss rate of $1.52 \cdot 10^{-3}$ % for 320 bytes.

Figure 4.15 contains the results for the POO3 test sources measured using the same setup. Each test on average generated the same network load as the corresponding MMC2 test (≈ 64, ≈ 74, ≈ 84 Mbit/s). The results however differ significantly from the ones shown in Figure 4.14. In general, a much larger amount of buffer space is required to completely eliminate packet loss in the switch. This is not surprising considering the infinite variance of the Pareto distribution.

For a network load of about 84 Mbit/s (the upper curve in Figure 4.15) and a buffer space of 768 kbytes, the packet loss rate is still 0.428 %. Furthermore, the slope of the loss-curve only decays slowly. The loss rate however significantly decreases when the network load falls below a certain utilization which occurs in Figure 4.15 between 84 and 74 Mbit/s. This is caused by the limited peak rate in the Pareto source model. Unlike the results in Figure 4.14, we can also observe a longer tail in all loss-curves in Figure 4.15. We for example still measured a loss rate of $1.43 \cdot 10^{-3}$ % for 221 test flows and 640 kbytes buffer space (the second curve in Figure 4.15). However, in order to achieve rates of under 1%, only buffer space of far less than 100 kbytes is required in the switch.

Figure 4.16 and Figure 4.17 contain results recorded by the Measurement Client for a single POO3 flow. Figure 4.16 shows the negative effect that a large buffer space can have on the maximum packet delay. As long as the switch is overloaded, the results increase linearly with the buffer space which may lead to large delays introduced by a single switch. The average delay is not significantly affected in our tests and only increases as a result of having a few large samples in the total set.

The optimum amount of buffer space to be used in LAN switches is hard to determine. In this section, we could observe that increasing the buffer space decreased or even eliminated the packet loss in the test switch, provided traffic bursts were temporary and moderate. This however required large buffer sizes in Switch 1 because the loss rate and the buffer space are not linearly related. The results have also shown that more buffer space does not always help. In case traffic characteristics exceeded a certain threshold in respect to network load and burstiness, even a large amount of buffer space could only insignificantly reduce the loss rate. Commercial LAN switches known to us have a buffer space between 128 kbytes and 512 kbytes available per-port. Parameters which likely had an impact in the selection process are: (1) the deployment location of the switch: e.g. desktop, work-

group or backbone level, (2) the switch configuration: e.g. the number of ports or the link speeds supported, and (3) the cost - performance trade-offs made by the engineers.

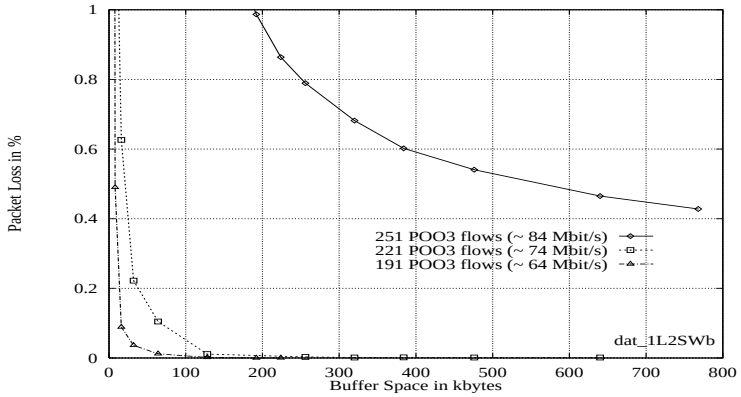


Figure 4.15: The Packet Loss Rate for the different Sets of POO3 Flows in Dependence of the Buffer Space in the Switch.

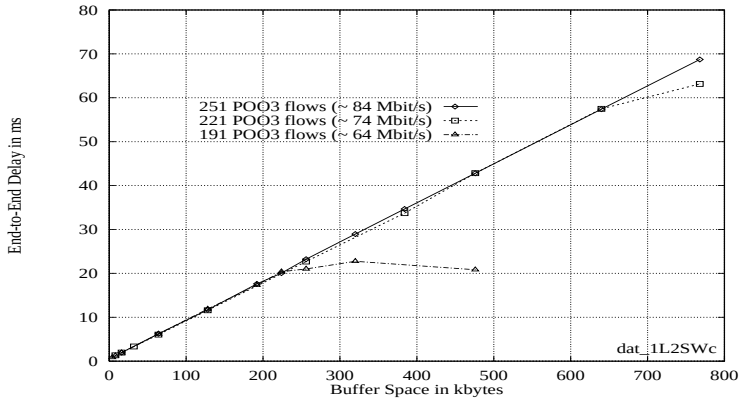


Figure 4.16: The Impact of the Buffer Space in *Switch 1* on the Maximum End-to-End Packet Delay.

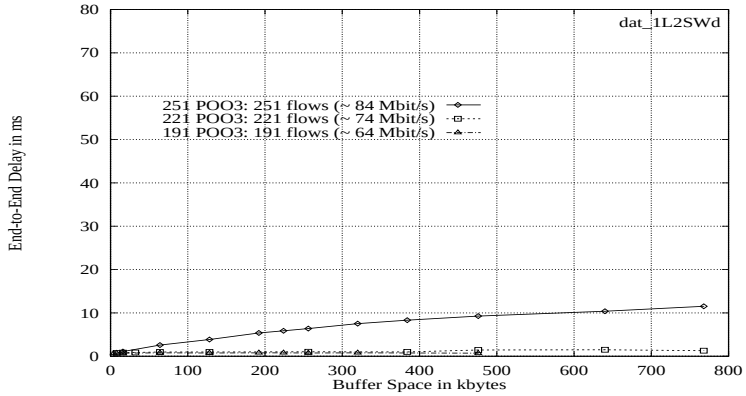


Figure 4.17: The Impact of the Buffer Space in *Switch 1* on the Average End-to-End Packet Delay.

4.4 Approaches to maintain QoS under Overload Conditions

A dynamic resource allocation based on admission control as proposed in the ISPN architecture is only one method that could be used to provide quality of service in the network. In general, two fundamentally different strategies could be pursued. First a network administrator might attempt to avoid any congestion in the network. This could be done based on (1) *Bandwidth Overprovisioning* or (2) *Usage Based Billing*. Secondly, he might setup the network such that it differentiates selected flows and provides a better service for them. This exploits the fact that some applications have stringent QoS requirements whereas others can tolerate service degradations well. (3) *Static Priorities* is the simplest and probably most cost-effective mechanism to differentiate flows. Since it will be available in many next generation LAN switches, we consider it in this section as a separate mechanism. To control the service quality, the network administrator might further use resource reservation with admission control. Two approaches for this can be identified: (4) a *Static Resource Allocation*, and (5) a *Dynamic Resource Allocation*. Each of these mechanisms is more or less appropriate under certain conditions and is briefly discussed in the following.

Bandwidth Overprovisioning

Installing more bandwidth in the network is the simplest way to improve the quality of service when the network shows signs of congestion. It further seems to be the only appropriate solution in the case that the network is continuously overloaded. In the wide area, bandwidth is still expensive. LAN technology however has become affordable and is easy to install. This especially applies to 10 Mbit/s and 100 Mbit/s interface cards, hubs and switches. Using Gbit/s technology is still associated with higher costs which might however be justified for example in the backbone of a large LAN. Furthermore, adding bandwidth can typically be performed gradually at selected locations in the network where bottleneck links were identified.

Typical questions in this context are [Shen95]: (1) How much more bandwidth is required in the network considering the bursty nature of the traffic, and (2) who pays for the cost of overprovisioning ? Traffic analysis within LANs has shown that the network utilization exhibits a cyclical behaviour with a cycle time of one day [LeWi91]. It can often be observed that each day has a few busy periods in which the network load is high. This is for example the case at about 10.30, 13.30 and 16.30 in the results[1] in [LeWi91]. Further analysis showed that busy periods include a few very bursty sub-periods[2]. In contrast, throughout the night, LANs are typically idle or only lightly loaded. Further, the ratio of the peak to average utilization over the day is high. If congestion occurs, then the network must be overprovisioned with a multitude of the bandwidth used on average. The appropriate ratio is case specific and depends on the network topology and applications used.

The main disadvantages of the approach are the costs for the new LAN equipment and the in general inefficient use of network resources. Service guarantees can still not be given, but the measure-

1. For details see Figure 3.1.1 in [LeWi91].
2. For details see Figure 3.2.1a and Figure 3.2.1b in [LeWi91].

ment results in this section have shown that the network can, when only moderately loaded, provide a sufficient service for existing time critical applications. Overprovisioning might thus not always be the desired solution, especially for LAN service providers or when the congestion occurs only temporarily.

Usage Based Billing

Usage based billing attempts to reduce the network utilization by charging users for the network resources they consume. To distribute the network load over the day, a LAN service provider might offer lower charges for data transfers during off-peak hours. Provided the resource demand decreases significantly, network users are likely to receive a better quality of service: even though this will not provide strict service guarantees.

We however do not believe that Usage Based Billing will be used within LANs, mainly due to the low costs for additional LAN bandwidth and the complexity of the accounting system required. Any such system would have to monitor data packets in hubs and LAN switches to accurately account for the total network traffic. Instead, a provider might much rather overprovision the network using the same investment. Usage based billing further assumes that network users take rational decisions: a fact which might not always be true. Utilization independent fees for outsourced network services are thus more likely to be negotiated with customers.

Static Priorities

Giving priority to delay sensitive flows within the network is a mechanism to improve the quality of service for these flows. As long as the network administrator can somehow ensure that the resource utilization in higher priority levels is always low, then static priorities is a cost effective solution to sustain temporarily network overload. It implies that at least part of the best effort traffic forwarded at lower priority is able to adapt to the available network capacity and backs off when its service rate decreases. Since existing LANs typically include a significant amount of TCP traffic, this assumption seems to be valid.

The main drawback is the starvation problem: a switch might cease to serve lower priority data packets due to excessive traffic to be forwarded at higher priority. Further, there are no control mechanisms which makes it difficult to maintain service guarantees. Furthermore, if everybody in the network is using the highest available priority then a LAN supporting priorities is no better than one forwarding all data packets with best effort.

Static Resource Allocation

Static resource allocation prevents starvation based on admission control. Resources are set up statically e.g. based on a manual switch configuration, and often remain allocated over long time scales such as weeks or month. Modifications are typically performed in response to topology changes or adjustments of the service level agreement. In contrast to all three previous methods, this can pro-

vide service guarantees over all time scales due to the advanced packet scheduling and the admission control applied.

As for the Static Priorities, additional costs in LAN switches are caused by the packet classifier and the scheduler. Even though the latter might imply any suitable scheme such as e.g. WFQ, we believe that in the near future, this will predominantly be based on static priorities or rate regulated static priorities.

A static resource allocation trades-off a simple resource management with a less efficient use of network resources. Simplicity is achieved by cutting out a potentially complex signalling protocol. The drawback is that resources might be allocated for inactive users or held longer than actually required. In spite of this, a static allocation seems to be a good compromise when performed e.g. for aggregated, delay sensitive traffic whose average data rate does not significantly change over time. It is thus likely to be used in LAN backbones or as part of service level agreements.

Dynamic Resource Allocation

A dynamic resource allocation scheme provides the most flexible and efficient mechanism to manage resources in the network. At the same time it typically also implies a higher complexity and costs. The details of this approach were discussed in Chapter 2.

Design Implications

There have been long debates between experts whether resource reservation and admission control in the network is needed [Clar95], [Ferr95], [Shen95] or not needed [Deer95]. For shared and switched LANs, a stringent requirement for this is even harder to justify because of the different cost and performance conditions in these networks. Since additional bandwidth, to a certain extend, is cheap, any resource reservation approach must be extremely cost effective to be able to compete with this solution. We believe that it actually has to be far less expensive than pure bandwidth to become a serious competitor. Bridged LANs are further almost self-configuring and easy to manage. Resource allocation systems should attempt to match this behaviour and dynamically discover network properties such as the network topology or intermediate link speeds.

The designers of LAN resource allocation schemes should thus aim at solutions with extremely low costs. Compromises in respect to the flexibility and the efficiency of the scheme however seem to be acceptable.

4.5 Summary

In this chapter, we discussed several experimental results showing basic performance characteristics of shared and switched 802.12 networks. First, we found that 802.12 networks actually do not provide a data throughput of 100 Mbit/s as envisaged by the standard. The throughput is further not constant, but may vary over a substantial performance range. Even though a degradation to some extent could be expected due to the Demand Priority signalling required to enforce a controlled

medium access, the actual degree of the performance loss was quite surprising. We further observed that the data throughput thus depends on the network topology and the size of the data packets used for the transmission. This suggests that: (1) cascaded networks should be built in rich, flat topologies with a low cascading level, and (2) large data packets should be used when possible. These dependencies further have a strong implication for the design and the complexity of resource allocation schemes which attempt to provide deterministic service guarantees, because it requires the Demand Priority protocol overhead to be considered in the admission control conditions.

The delay and loss characteristics basically confirmed our expectations. As long as the network operated at a low or moderate utilization, we observed a low average delay and no packet loss for all test sources. The behaviour suggested that several average delay classes can probably not be differentiated by existing applications and should thus not be implemented. The performance parameters to be controlled in LANs are: (1) the packet loss, and (2) the maximum delay. Packet loss may even occur when the average delay is still in the order of a few milliseconds. Further, increasing the amount of buffer space within LAN switches improves the loss behaviour in the network but may be expensive. The actual gain depends on the characteristics of the traffic in the network. To completely eliminate the packet loss in LAN switches may thus be impossible or require a substantial amount of memory. Beside its costs this has also a negative impact on the maximum delay. Furthermore, we looked at several mechanisms to provide quality of service within LANs and identified low implementation costs as a design goal for LAN resource reservation schemes.

Chapter 5
802.12 Network Analysis

Building an accurate resource allocation system on top of the 802.12 high priority access mechanism first requires the computation of the available bandwidth in the network. The result of this computation then defines the bandwidth limit up to which a resource allocator may allocate resources. This is essential not just to ensure that allocated resources are actually available on the network, and thus that delay bounds and buffer space requirements are met according to the service specification. More importantly, it enables the resource allocator to guarantee that a certain minimum bandwidth is always free for the best-effort service by sufficiently restricting the access to the high priority service.

In this chapter, we analyse the Demand Priority medium access mechanism in detail and derive upper bounds for the signalling overhead. These results enable the admission control conditions defined in Chapter 6 to accurately determine the minimum available bandwidth in 802.12 networks. We start with an outline of the access protocol operation and its theoretical performance constraints. Section 5.2 then investigates the protocol overhead in 802.12 networks using a UTP physical layer. For this we define parameter specific worst-case packet transmission models in order to comply with the requirements for a deterministic network service. Section 5.3 derives the equivalent parameters for networks with a Fiber-Optic physical layer. The impact of 802.5 packet frame formats is discussed in Section 5.4, before we summarize the chapter in Section 5.5.

5.1 802.12 and Demand Priority

As with other network technologies standardized within the IEEE, 802.12 is structured in a Media Access Control (MAC) sublayer, a Physical Medium Independent (PMI) sublayer, a Medium Independent Interface (MII), and a Physical Medium Dependent (PMD) sublayer. The MAC controls the access to the medium and carries out the link training. Both are based on the Demand Priority protocol. The PMI performs the quartet channelling, the 5B6B block data encoding, and adds the preamble pattern and the start and end delimiters. The PMD performs the NRZ encoding and controls the link status. We refer to the standard [ISO95] for the details of the functionality implemented in each sublayer.

The Demand Priority protocol has two characteristics which allowed us to built a Guaranteed service: (1) the support of two priority levels, and (2) a deterministic medium access and service order: data packets from all network nodes are served using a simple round-robin algorithm. Data are transmitted using either IEEE 802.3 or 802.5 frame formats. Several physical layers have been

defined. In particular the standard supports Category 3 UTP cable, which is the most widely deployed cabling within LANs. Also specified is the operation over Shielded Twisted Pair (STP) and over multimode fibre.

5.1.1 Network Operation

In a single hub network, the shared medium access is entirely controlled by the hub. 802.12 nodes wishing to transmit a data packet first signal a service request (or demand) to the hub. The request is labelled with either normal or high priority. The hub is continually scanning each of its attached ports and maintains two separate service lists: one for normal priority and one for high priority requests. All high priority requests are served first. For this, the hub acknowledges the request of the next node in its current round-robin cycle and grants the transmission of one packet. After receiving the corresponding control signal, the selected node starts sending its packet to the hub. As the hub receives the packet, it decodes the MAC address information in the packet header, selects the output port, and then only forwards the packet to its destination. This filtering is possible because the hub learned the MAC addresses of all nodes connected to it during a link training process, which is executed when the link to a network node is setup. Multicast and broadcast frames are however send to all nodes on the shared segment. The hub continues this process until the high priority list is empty and then carries on serving demands for the normal priority network service.

Whenever the hub receives a high priority request while its normal priority service list is being served, it completes the processing of the current request before it begins to serve high priority requests. The normal priority service is only resumed after all high priority requests have been served.

To control the shared medium access in cascaded topologies, the basic Demand Priority protocol was extended by the 802.12 working group. A mechanism was introduced to allow the distributed operation of the algorithm. As in the single hub topology, there is however always only one hub in control of the network. Using specific link level signalling, the network control is then passed from hub to hub in the network, such that all network nodes are collectively served in a single shared round-robin domain.

The following basic algorithm is carried out: whenever the cascaded network is idle then the network control is at the Root hub. Nodes wishing to transmit a packet first signal their service request to the hub to whom they are connected to (their local hub), just as described for the single hub case. To serve the request, the local hub must however first acquire the network control. If the hub is not the Root hub, then the request is passed on through the Up-link to the next upper hub, and so on until it reaches the Root hub. Following the basic Demand Priority protocol, the Root hub serves all requests in round-robin order. It can distinguish whether a request was received from a directly connected network node, or from a lower Level-1 hub. Whenever the service request from a lower Level-1 hub is granted then the Root hub passes the network control down to that hub. Having the network control enables the Level-1 hub to serve one request from all nodes connected to it. If

required, then the network control is passed further down to a lower Level-2 hub, and so on, so that requests from nodes at the leaves of the topology tree can be served. The network control is returned after a hub has once served a request from all downstream nodes and hubs. Note that the control is only passed down on request. It is never given to a lower level hub that does not have a pending service request.

The two priority levels are also supported in cascaded topologies. If the Root hub receives a high-priority request while a lower level hub is in the process of servicing normal-priority requests, then the Root hub can effectively interrupt the lower level hub in order to serve the high priority request first. This is based on the use of a special 802.12 control signal. After the network has processed all high priority requests, it continues the normal priority service at the point in the network, at which it was interrupted. This ensures that fairness is maintained, even in large networks with many hubs.

The service policy is however unfair if different nodes use different packet sizes. This is because hubs do not consider the size of the packets transmitted. Further details about the 802.12 technology and a comparison with the 100BaseT standard (IEEE 802.3u) can be found in [WAG+95] and [MoWa96].

5.1.2 Performance Parameters and their Dependencies

To describe the Demand Priority overhead we identified two network parameters: (1) the worst-case per-packet overhead, and (2) the worst-case time it takes to pre-empt the normal priority service (the normal priority service interrupt time). Both parameters allow us to determine the maximum bandwidth that can be allocated while giving deterministic service guarantees. They depend on: (1) the network cascading level, (2) the physical layer technology, and (3) the cable length.

The network cascading level has a significant impact because of the increased signalling delay within large shared multi-hub topologies. The physical layer can introduce an additional delay when operating in half-duplex mode. This is the case for data transmissions over UTP links. Since data are transmitted on all four pairs across such cables, no 802.12 link control signals can be exchanged during that time. This leads to further transmission delays and increases the normal priority service interrupt time. The delay is not introduced across STP or fiber-optic links since these operate in dual-simplex mode and can exchange data and control signals at the same time. The dependency from the cable length is caused by the propagation delay introduced for control signals and data across the network. This will be significant for long fiber-optic links which may have a length of up to 2 km [ISO95].

To determine the worst-case per-packet overhead and the normal priority service interrupt time, the Demand Priority link control signals and the packet transmission on 802.12 networks must be analysed in great detail. This is performed in the following. We first focus on a non-bundled UTP physical layer due to its wide deployment and the half-duplex character of the data transmission. In the analysis, we further assume 802.3 frame formats for all data packets transmitted. The impact of 802.5 formats on our analytical results will be discussed afterwards in Section 5.4.

5.2 Performance Parameters for the UTP Physical Layer

5.2.1 The Per-Packet Overhead in Single Hub Networks

The communication between network nodes and the hub is based on the exchange of 802.12 link control signals. There are 6 primary control signals that are relevant for the packet transmission in single hub networks. The Idle signal (*Idle*) indicates that the sender e.g. a host currently has no request pending for the hub connected at the other end of the link. The Request signal (*Req_H*, *Req_N*) is used to demand the transmission of a normal (*Req_N*) or high priority (*Req_H*) data packet. The Grant signal (*Grant*) indicates that the node has been given permission to send a packet. *Incoming* will be signalled by the hub in order to inform nodes that a packet may soon be sent to them. This allows them to prepare themselves for the receipt.

To determine the overhead caused by: (1) the Demand Priority protocol itself and (2) by passing a data packet through the protocol stack, we defined a packet transmission model which describes the case when the lowest network throughput is achieved with a hub that never runs idle. This is based on worst-case assumptions. The worst case is reached in two configurations: (1) when two nodes are switching between sending and receiving unicast data packets, or (2) when two or more nodes send data packets using the multicast or broadcast addressing mechanism. In both cases the receiver of the last data packet is also the receiver of the next grant. This forces the hub to add an extra time offset, which is called *SEND_IDLE_BURST* (*I_BST*), before the grant is signalled to the node.

Figure 5.1 shows the *Time-Space* diagram for the transmission of three data packets using the high priority service. Further depicted is the example topology consisting of two network nodes e.g. hosts and one hub. Since Time-Space diagrams will be frequently used in this chapter, we describe them here in detail before discussing the data flow relevant for the per-packet overhead. The space between the upper two horizontal lines in the diagram represents the link *L2* in the example topology. Analogous to this, link *L1* is the space between the lower two horizontal lines. Link control signals are shown as arrows indicating the source and the destination of the signalling. The transmission of data packets is shown using large boxes carrying the label *DATA*. Control signals and data packets are further textured differently.

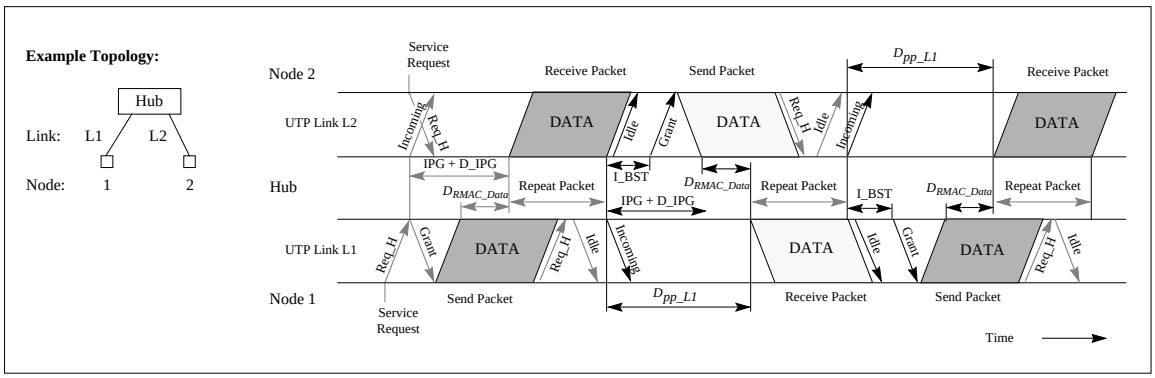


Figure 5.1: Worst-Case Signalling on a Single Hub Network using a UTP Physical Layer.

This is not specific to Time-Space diagrams, but is used to emphasize details of the data flow that are relevant for computing the per-packet overhead. The x-axis of the diagram provides the time consumed for each operation. The slope of control signals and data packets thus represents the propagation delay on the link. Further shown are delays introduced by the hub. The parameter D_{RMAC_Data} for example denotes the delay encountered by each data packet during the packet forwarding. It can instantly be observed that all data packets in Figure 5.1 are forwarded using cut-through switching because the hub starts the data transmission long before it has received the end of the data packet. Further examples for Time-Space diagrams can be found in [ISO95].

The data flow in Figure 5.1 starts when the upper layer of Node 1 passes a data packets to the 802.12 MAC layer. After receiving the packet, the MAC at Node 1 signals *Req_H* to the hub, demanding the transmission of the high priority data packet. If the hub is idle, as assumed at the beginning of the data flow in Figure 5.1, then the hub immediately acknowledges the request and returns a Grant signal to Node 1. At the same time, the hub signals Incoming to all other nodes on the network such as Node 2. After detecting the grant, Node 1 starts transmitting the data packet to the hub, which then forwards the packet to Node 2. The packet processing in the hub introduces a small delay (D_{RMAC_Data}). While the rest of the packet is repeated, the hub signals Idle to all nodes other than the destination e.g. to Node 1. This allows them to signal their next service request (Req_H, Req_N) or Idle to the hub. In Figure 5.1, Node 1 requests the transmission of another high priority packet by signalling Req_H. This assumes that another data packet was passed into the output queue at Node 1 while the first packet was transmitted to the hub.

In the meantime, the hub has also received a transmission request from Node 2. This request is granted after the packet from Node 1 has been fully repeated. The corresponding Grant signal is however not signalled before the SEND_IDLE_BURST (I_BST) timer has expired on the hub. This idle window allows Node 2 to potentially signal a service request to the hub. The transmission of the data packet from Node 2 requires the same signalling as described for the previous data packet. After the packet from Node 2 has been repeated, the hub continues and processes the next request from Node 1 and so on, until all requests have been served.

The medium access mechanism defines that the gap between two subsequent packet transmissions is always larger than a certain defined time interval called the *Inter-Packet Gap* (*IPG*). This is enforced by the IPG timer mechanism at the hub. If the packet was received from a node, then the interpacket gap is increased by an additional time offset of length *D_IPG*. It accounts for clock differences between different hubs in the shared network. The per-packet overhead denoted with D_{pp_L1} in a single hub (Level-1 cascaded) network is thus at least as big as IPG plus D_IPG[1].

The worst case however is determined by the maximum signalling-, packet-processing and propagation delay as illustrated in Figure 5.1. This includes the worst-case delay for: (1) signalling Grant from the hub to the node, (2) passing the data packet through the 802.12 protocol stack, (3) trans-

1. IPG + D_IPG correspond to a numerical value of 7.0 µs according to the 802.12 standard.

mitting the packet across the link, (4) receiving the packet at the hub and passing it to the MAC layer, and (5) decoding the address information and passing the data packet to the PMI of the outgoing port. The precise breakdowns for these operations are given in Table 5.3 and Table 5.4.

Sublayer	Comments	Worst Case Delay	Reference Section in [ISO95]
RMAC (Hub)		-	12.6.3.4 12.6.4.1
PMI	$D_{PMI\_Tx\_Ctrl}$ Control signal encoding, (control signals do not have a preamble).	4 BT	14.3.1
PMD	$D_{PMD\_Tx\_Ctrl}$ Propagation delay within the PMD.	20 BT	16.5.3.2
PHY (Link)	D_{PHY} Propagation delay on 100 m UTP, STP cable.	570 ns	16.9.1.3
PMD	$D_{PMD\_Rx\_Grant}$ Grant signal detection.	6 BT	16.6.5
PMI	$D_{PMI\_Rx\_Ctrl}$ Control signal mapping.	4 BT	14.3.2 14.3.3
MAC (Receiver)		-	

Table 5.3: Breakdown of the Grant-Signalling Delay for a UTP Physical Layer.

Sublayer	Comments	Worst Case Delay	Reference Section in [ISO95]
MAC (Source)		-	12.6.3.4 12.6.4.1
PMI	$D_{PMI\_Tx\_Data}$ Addition of the preamble pattern (48 BT): Addition of the Starting Delimiter (12 BT): Propagation delay for data (3 BT):	63 BT	 14.4.2.3.2 14.4.2.3.3 14.3.4
PMD	$D_{PMD\_Tx\_Data}$ Maximum propagation delay within PMD.	8 BT	16.5.2
PHY (Link)	D_{PHY} Propagation delay on 100 m UTP, STP cable.	570 ns	16.9.1.3
PMD	$D_{PMD\_Rx\_Data}$ Data recovery delay.	10 BT	16.6.4
PMI	$D_{PMI\_Rx\_Data}$ Synchronization, data decoding (8 BT): Propagation delay within the PMI (3 BT):	11 BT	 14.4.4 as 14.3.4
MII -> MII (Hub)	$D_{MII\_Rx\_Tx\_Data}$ Transmit delay from the receiving MII to the sending MII in the RMAC.	4.5 µs	12.9.7.2

Table 5.4: Breakdown of the Data Transmission Delay for a UTP Physical Layer.

All delays are worst-case delays and based on references in the standard. A Bit Time (BT) corresponds to $33.\bar{3}$ ns, e.g. $D_{PMD\_Rx\_Grant}$ in Table 5.3 is equal to 200 ns. The propagation delays on the physical medium are provided for 100 m Category 3 UTP cable. Further, we assume in our model, that the Medium Independent Interface (MII) itself does not introduce any significant delay.

Using the transmission model in Figure 5.1 and the results in Table 5.3 and Table 5.4, we are able to compute the worst-case per-packet overhead D_{pp_L1}. During the computation, we denote the overhead caused by the data transmission across a single link with D_{Tx_Data}. The parameter D_{Signal_Grant} is the worst-case time it takes to signal Grant across the link. Both parameters are computed later. Under idle network conditions, the Grant signal can travel much faster than the data signal due to a smaller overhead in the sending and receiving 802.12 PMDs and PMIs. We can however observe in Figure 5.1 that under worst-case conditions the Grant always travels behind a data packet. Node 2 can thus not detect the Grant signal in $I\_BST + D_{Signal\_Grant}$ time units after the hub has made its decision to serve this node. Instead, Node 2 first has to receive the data packet. We assume in our model that the Grant has been detected I_BST time units after the last bit of the data packet has been received at Node 2. The resulting delay is therefore: $D_{Tx\_Data} + I\_BST$. When detecting the Grant, Node 2 instantly sends the data packet. It takes not more than: $D_{Tx\_Data} + D_{RMAC\_Data} + P_{max}/C_l$ time units until the hub has fully repeated this packet, where D_{RMAC_Data} denotes the worst-case time, the packet is delayed in the RMAC of the hub. P_{max}/C_l is the transmission time for a data packet of maximum size. If we now consider that the per-packet overhead is always larger than the inter-packet gap: IPG + D_IPG then we have for the worst case per-packet overhead D_{pp_L1} in a single hub network:

$$D_{pp\_L1} \leq MAX((IPG + D\_IPG), (D_{Tx\_Data} + I\_BST + D_{Tx\_Data} + D_{RMAC\_Data})) \quad (5.1)$$

The timer values for the IPG- and D_IPG window, and the I_BST offset are defined in the standard (see Section 12.5.1). The numerical results for D_{Signal_Grant} and D_{Tx_Data} immediately follow from Table 5.3 and Table 5.4 by adding up the delay components introduced in each sublayer of the 802.12 protocol stack. We thus have:

$$D_{Signal\_Grant} = D_{PMI\_Tx\_Ctrl} + D_{PMD\_Tx\_Ctrl} + D_{PHY} + D_{PMD\_Rx\_Grant} + D_{PMI\_Rx\_Ctrl} \quad (5.2)$$

$$D_{Tx\_Data} = D_{PMI\_Tx\_Data} + D_{PMD\_Tx\_Data} + D_{PHY} + D_{PMD\_Rx\_Data} + D_{PMI\_Rx\_Data} \quad (5.3)$$

The delay in the RMAC sublayer (D_{RMAC_Data}) is computed based on the delay bounds given in Table 5.4. Since the standard provides the worst-case delay between the receiving and transmitting MII of the RMAC, we receive D_{RMAC_Data} by taking off the delays added by the PMIs:

$$D_{RMAC\_Data} = D_{MII\_Rx\_Tx\_Data} - D_{PMI\_Rx\_Data} - D_{PMI\_Tx\_Data} \quad (5.4)$$

This provides a delay of 2.033 μs. This value is fixed, the results for D_{Signal_Grant} and D_{Tx_Data} however depend on the cable length. Table 5.5 in the next section contains the numerical results for D_{pp_L1}. These were computed from Equations 5.1 - 5.4 and the values in Table 5.3 and Table 5.4.

5.2.2 The Per-Packet Overhead in Multi-Hub Networks

In this section, we derive the worst-case per-packet overhead for multi-hub 802.12 network topologies. The worst case occurs under exactly the same conditions as discussed for the single hub network.

Figure 5.2 shows a model for the packet transmission and the signalling that is required for transmitting four data packets across a Level-2 cascaded network. The model only shows the signalling details which are relevant for deriving the per-packet overhead in this topology. It also omits the normal and high priority service request signalling (Req_H, Req_N), the D_{RMAC_Data} delay, and the IPG, D_IPG and I_BST timer constraints discussed in the previous section. The worst-case per-packet overhead for this topology is denoted by D_{pp_L2}. The example topology consists of three hubs and two nodes. Each node is connected to a Level-2 hub creating a maximum data path between the two nodes. We further assume that both nodes have at least two data packet to send and request the same service priority. The data flow starts when Node 1 sends a data packet. This packet travels along the data path and traverses all three hubs in the network on its way towards Node 2. When the Root hub has finished repeating the packet, it hands the network control over to Hub 3. This uses the Grant signal. Having the network control enables Hub 3 to serve the request from Node 2. For this, Hub 3 carries out the same procedure as a hub in a single hub network: it sends a Grant to Node 2 and, when it receives the data packet, forwards the packet towards the destination e.g. towards Node 1.

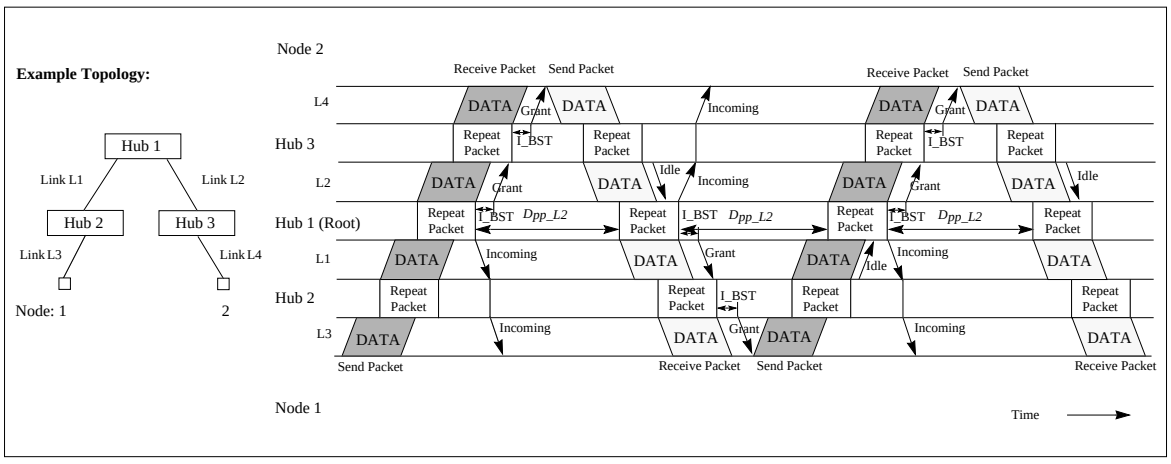


Figure 5.2: Worst-Case Signalling on a Level-2 Cascaded Network using a UTP Physical Layer.

After forwarding the last bit, Hub 3 passes the network control back to the Root hub by signalling Idle, as shown in Figure 5.2. The Demand Priority timing constraints ensure that the Root hub receives the network control before it has itself repeated the last bit of the data packet from Node 2 towards Hub 2. After the packet processing is finished, the Root hub hands the network control over to Hub 2 so that the next request from Node 1 can be served. When this request has been processed then the control is again given to Hub 3 and so on. The network control is thus passed between both Level-2 hubs for each service request in the network. This creates a maximum overhead without that the network runs idle.

As already observed for the single hub case, the Grant signalling in Figure 5.2 is always delayed by a preceding data packet. This increases the per-packet delay since: $D_{Tx\_Data} > D_{Signal\_Grant}$. The delay between the time when the Root hub decides to pass the network control to Hub 3 and the time when Node 2 detects the Grant signal is thus as long as: $D_{Tx\_Data} + D_{RMAC\_Data} + D_{Tx\_Data} + I\_BST$. This follows from Figure 5.2 and the considerations made for the single hub case. When Node 2 starts the packet transmission, it takes a maximum of: $D_{Tx\_Data} + D_{RMAC\_Data} + D_{Tx\_Data} + D_{MAC\_Data}$ time units until the MAC of the Root hub passes the first bit of the data packet to the PMI of link L1. If we again consider the constrain of the 802.12 standard that the gap between two subsequent data packets is at least as big as the interpacket gap: IPG + D_IPG, then we receive for the worst-case per-packet overhead D_{pp_L2} in a Level-2 topology:

$$\begin{aligned} D_{pp\_L2} \leq\ & MAX\ ((IPG + D\_IPG)\ , \\ & (D_{Tx\_Data} + D_{RMAC\_Data} + D_{Tx\_Data} + I\_BST + \\ & D_{Tx\_Data} + D_{RMAC\_Data} + D_{Tx\_Data} + D_{RMAC\_Data})) \end{aligned} \tag{5.5}$$

The same consideration as for the Level-2 topology can also be made for higher cascaded networks. This is omitted here because the results are straightforward when considering the results received for the Level-1 and the Level-2 cascaded network. If we rearrange Equation 5.5, then we have:

$$\begin{aligned} D_{pp\_L2} \leq\ & MAX\ ((IPG + D\_IPG)\ ; \\ & (D_{Tx\_Data} + I\_BST + D_{Tx\_Data} + D_{RMAC\_Data} + \\ & 2 \cdot (D_{Tx\_Data} + D_{RMAC\_Data}))) \end{aligned} \tag{5.6}$$

A comparison of Equation 5.6 with the result received for the single hub network shows that both results only differ by the term: $2 \cdot (D_{Tx\_Data} + D_{RMAC\_Data})$. This can be generalized since for each higher cascading level, the maximum network data path always increases by two hubs and two links, which causes an additional delay of: $2 \cdot (D_{Tx\_Data} + D_{RMAC\_Data})$ for data packets travelling along this path. This can for example be observed in Figure 5.2. The worst case per-packet overhead D_{pp_LN} in a Level-N cascaded topology is thus given by:

$$\begin{aligned} D_{pp\_LN} \leq\ & MAX\ ((IPG + D\_IPG)\ ; \\ & (D_{Tx\_Data} + I\_BST + D_{Tx\_Data} + D_{RMAC\_Data} + \\ & 2 \cdot (N-1)(D_{Tx\_Data} + D_{RMAC\_Data}))) \end{aligned} \tag{5.7}$$

where $1 \leq N \leq 5$. Equation 5.7 assumes that D_{Tx_Data} has a single upper bound for all UTP links in the multi-hub network. Such a bound can easily be found since the maximum UTP cable length may not exceed 200 m. It further ensures simplicity. The alternative would have been to use link specific values for D_{Tx_Data} based on a worst-case data path for high priority traffic. Identifying this worst-case data path may however be hard and requires a re-configuration of the path parameters used in the admission control whenever this path changes. Only a limited gain can further be achieved in following this strategy because of the rather small dependency between the per-packet overhead and the UTP cable length. This is shown by the numerical results in Table 5.5. We computed them using Equation 5.7 with the results received from Equation 5.3 and Equation 5.4.

UTP Cable Length	Network Cascading Level N				
	1	2	3	4	5
5 m	9.03 µs	19.29 µs	29.55 µs	39.81 µs	50.07 µs
100 m	10.11 µs	21.45 µs	32.79 µs	44.14 µs	55.48 µs
200 m	11.25 µs	23.73 µs	36.21 µs	48.70 µs	61.18 µs

Table 5.5: Per-Packet Overhead D_{pp_LN} for Cascaded Networks using a UTP Physical Layer.

A comparison shows that for a UTP sublayer, the cascading level has a much larger impact on the per-packet overhead than the cable length. This is particularly true for the results received for the Level-1 and Level-2 topologies which are likely to be the most widely used. The impact of these results on the computed minimum available bandwidth and a comparison with the measured worst-case throughput is performed in Section 6.5.1 in Chapter 6.

5.2.3 The Per-Packet Overhead in Half-Duplex Switched Links

Figure 5.3 shows the model which we used to determine the overhead for half-duplex switched links. It is simpler than the model for cascaded networks because the data path only includes a single link. Let us first consider the example topology. We assume that Switch 1 operates in RMAC mode, and Switch 2 in MAC mode.

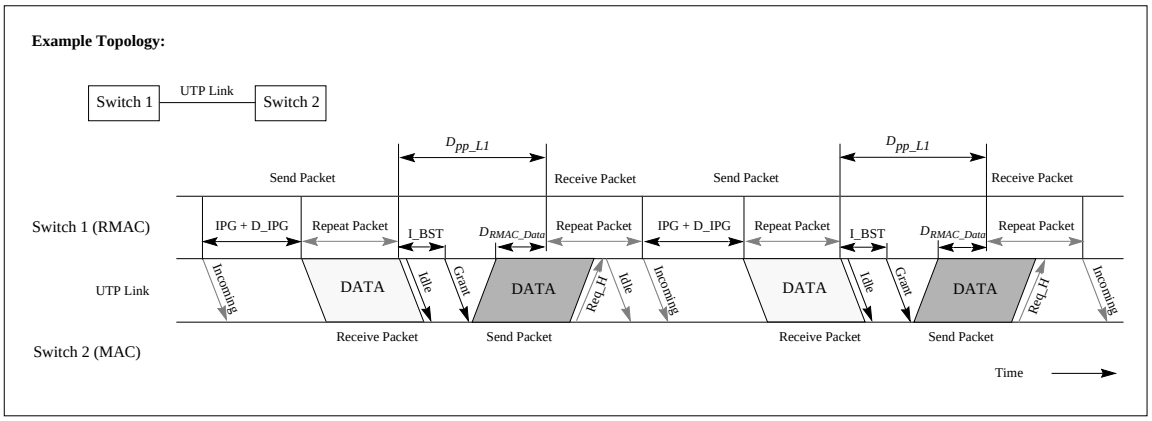


Figure 5.3: Worst-Case Signalling on a Half-Duplex Switched UTP Link.

The per-packet overhead for data packets send by Switch 1 is equal to the interpacket gap: $IPG + D\_IPG = 7.0\ \mu s$. This is because the RMAC in Switch 1 controls the link access and thus does not have to signal across the link to request the network service. If in contrast all data packets were *only* transmitted by Switch 2 then the maximum per-packet overhead would be: $D_{Signal\_Grant} + D_{Tx\_Data} + D_{RMAC\_Data} \leq 7.376\ \mu s$. This assumes a 100 m UTP cable between Switch 1 and Switch 2. The computation used the results from Section 5.2.1. The idle burst *I_BST* does not have to be considered in this case since Switch 2 does not receive any data packets.

The worst case is however again achieved when both switches toggle between sending and receiving data packets as depicted in Figure 5.3. How these packets are addressed is not significant for the result. The first data packet in the Time-Space diagram is sent by Switch 1. Assuming that Switch 2 has previously made a request for the network service, the RMAC on Switch 1 signals Grant after its data packet has been sent and the I_BST timer expired. As soon as Switch 2's MAC has detected the Grant, it starts transmitting its packet to Switch 1. As in cascaded networks, we assume that the RMAC in Switch 1 introduces a maximum delay of D_{RMAC_Data} required to decode the address information before it passes on the packet received from Switch 2 to another switch port. An RMAC implementation customized for a use within switches can however be expected to be much faster than that because only two RMAC ports need to be supported, one of which is the input port of the data packet. No address lookup is thus needed in this case. Since it is however not likely that all switches will use a custom-built RMAC chip, we do consider D_{RMAC_Data} in our computation. If we further take into account that in the worst case, each Grant signals becomes delayed by a data packet from Switch 1 - as previously discussed in Section 5.2.1, then we receive a maximum overhead of: $D_{Tx\_Data} + I\_BST + D_{Tx\_Data} + D_{RMAC\_Data}$ for the first data packet from Switch 2. This result is identical to the worst-case per-packet overhead D_{pp_L1} received for the single hub network.

All following data packets will have the same overhead as the first two packets provided the output queues of Switch 1 and Switch 2 remain occupied. For all four packets in Figure 5.3 we thus have: $D^{*}_{pp\_HD} \leq (IPG + D\_IPG) + D_{pp\_L1} + (IPG + D\_IPG) + D_{pp\_L1}$ for the worst case per-packet overhead.

In contrast to the results computed for cascaded networks, the overhead across half-duplex switched links depends on the direction in which the data path is crossed. This is caused by the non-symmetric medium access control which reduces the overhead for data packets from Switch 1 to the minimum. There is however no need to consider this dependency in the admission control conditions. The simplest upper bound for the worst-case overhead is given by: $D_{pp\_HD} = D_{pp\_L1}$. We however use a more accurate approach by taking the average of two packets: one from each direction. This provides:

$$D_{pp\_HD} \leq (IPG + D\_IPG + D_{pp\_L1})/2 \tag{5.8}$$

Numerical results are shown in Table 5.6. They were computed based on Equation 5.8 and the results for D_{pp_L1} in Table 5.5.

UTP Cable Length	D_{pp_HD}
5 m	8.01 µs
100 m	8.56 µs
200 m	9.13 µs

Table 5.6: Per-Packet Overhead D_{pp_HD} for Half-Duplex Switched UTP Links.

It remains to remark that Equation 5.8 provides worst-case results over a time interval of: $2 \cdot (D_{pp\_HD} + P_{max} / C_l) \approx 260\ \mu s$, which corresponds to the transmission time of two maximum sized data packets using the 802.3 packet format. The parameters P_{max} and C_l denote the maximum link packet size and the 802.12 link speed, respectively. Note further that averaging over such a time interval does not impair the deterministic service guarantees provided by our allocation system because the allocation is based on much longer time frames which are in the order of at least a few milliseconds.

The second network parameter required for determining the resource allocation limit is the normal priority service interrupt time. It is derived in the following for different network topologies. We start again with the single hub network.

5.2.4 The Interrupt Time in Single Hub Networks

The example topology used for the analysis is shown in Figure 5.4. The corresponding Time-Space diagram contains the worst-case signalling required for pre-empting the normal priority service and for transmitting a single high priority data packet. Unlike the diagrams discussed in the previous sections, the space between the upper two horizontal lines in Figure 5.4 represents two links: *L2* and *L3*. Further, only the signalling relevant for the computation is shown. Also omitted are the D_{RMAC_Data} delay and the IPG, D_IPG and I_BST timer constraints previously discussed.

The example network consists of a single hub and three nodes. We describe the interrupt time in respect to Node 1 which is requesting the transmission of a high priority data packet. The two other nodes in the setup, Node 2 and Node 3, only use the normal priority service. Similar to the packet transmission model discussed for the per-packet overhead, the worst-case delay occurs when Node 2 and Node 3 send data packets using multicast or broadcast, while Node 1 is requesting the high priority service. The worst-case normal priority service interrupt time is denoted by D_{it_L1}. It occurs when: (1) the signalling of the high priority request (Req_H) from Node 1 to the hub is delayed by the transmission of normal priority data packets on the network, and (2) these data packets are of maximum size. In a single hub topology, a maximum of *two* data packets can be served by the hub before the normal priority service is pre-empted. This is caused by the half-duplex operation of the UTP physical layer and will be outlined in the following.

The data flow in Figure 5.4 starts when Node 1 sends a multicast data packet. This is forwarded towards Node 2 and Node 3. At the same time, we assume that Node 2 has a pending normal priority service request. Instantly after the hub decided to serve this request, it also signals Incoming to Node 1 which is running idle at that time. Note that the hub forwards multicast data packets regardless of whether network nodes have joined the corresponding multicast group or not. Multicast data packets will thus always be forwarded to Node 1.

The worst case condition for D_{it_L1} occurs if a high priority request is made at Node 1 instantly after the Incoming signal was detected. In this case, the physical layer (PMD) at Node 1 does not signal Req_H to the hub because it must prepare itself for receiving the data packet from Node 2. As shown in Figure 5.4, the Req_H signal is not transmitted before the normal priority data packet from Node 2 has been fully received at Node 1.

After the hub repeated the packet from Node 2, it runs idle until it receives a demand for transmitting a normal priority data packet from Node 3. The worst case occurs when the high priority request from Node 1 arrives at the hub just after the normal priority request from Node 3 has been acknowledged. The hub then first grants the transmission of the packet from Node 3. After forwarding this packet, the normal priority service is pre-empted and the hub starts to serve the high priority packet from Node 1. Note that even though the normal priority request arrives later at the MAC of Node 3, it is served earlier by the hub than the high priority data packet from Node 1.

Assuming that both nodes, Node 2 and Node 3, send a maximum size data packet, we find in Figure 5.4 that the worst case interrupt time D_{it_L1} is given by:

$$D_{it\_L1} \leq 2 \cdot \frac{P_{max}}{C_l} + d_2 + d_1 \tag{5.9}$$

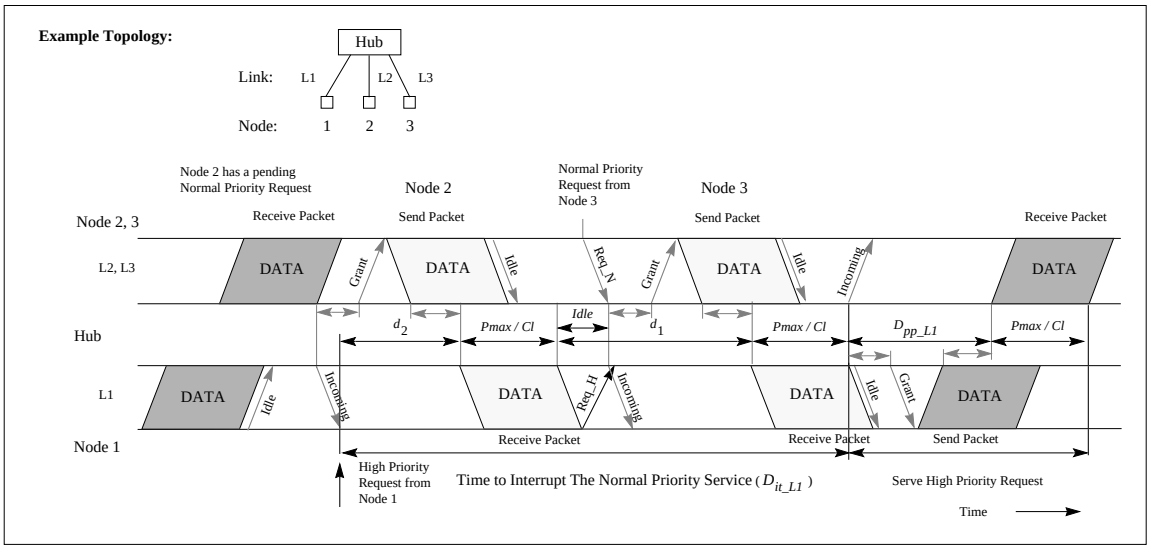


Figure 5.4: The Model for Computing the Worst-Case Interrupt Time in a Single Hub Network using a UTP Physical Layer.

where P_{max}/C_l is the time it takes to transmit one data packet of maximum size. The two constants d_2 and d_1 contain the overhead for the two normal priority data packets. The overhead for the packet from Node 2 is the worst-case overhead D_{pp_L1} computed for the single hub network. This can be seen when comparing the signalling and data transmission with the worst-case model in Figure 5.1. For the interrupt time, we however only have to consider:

$$d_2 = D_{pp\_L1} - D_{Incom} \tag{5.10}$$

where D_{Incom} is the time it takes to signal Incoming across the UTP link. This can be observed in Figure 5.4. The overhead for the normal priority packet from Node 3 also follows from Figure 5.4:

$$d_1 = D_{Tx\_Data} + D_{Req\_H} + I\_BST + D_{Signal\_Grant} + D_{Tx\_Data} + D_{RMAC\_Data} \tag{5.11}$$

where D_{Tx_Data}, D_{Signal_Grant}, D_{RMAC_Data} and I_BST are the parameters discussed and computed in Section 5.2.1. D_{Req_H} is the time it takes to signal Req_H across a link. Both parameters, D_{Req_H} and D_{Incom}, have the same numeric value which we denote with D_{Signal_Ctrl}:

$$D_{Incom} = D_{Req\_H} = D_{Signal\_Ctrl} \tag{5.12}$$

A precise breakdown for D_{Signal_Ctrl} is provided by Table 5.7.

Sublayer	Comments	Worst Case Delay	Reference Section in [ISO95]
RMAC (Hub)		-	12.6.3.4 12.6.4.1
PMI	$D_{PMI\_Tx\_Ctrl}$ Control signal encoding, (control signals do not have a preamble).	4 BT	14.3.1
PMD	$D_{PMD\_Tx\_Ctrl}$ Max. propagation delay within the PMD.	20 BT	16.5.3.2
PHY (Link)	D_{PHY} Prop. delay on 100 m UTP or STP cable.	570 ns	16.9.1.3
PMD	$D_{PMD\_Rx\_Ctrl}$ Control signal recovery and decoding.	48 BT	16.6.1
PMI	$D_{PMI\_Rx\_Ctrl}$ Control signal mapping.	4 BT	14.3.2
MAC (Receiver)		-	

Table 5.7: Breakdown of the Delay required for Signalling the Control Signals Req_H, Req_N and Incoming across a single UTP Link.

Using these components, we get:

$$D_{Signal\_Ctrl} = D_{PMI\_Tx\_Ctrl} + D_{PMD\_Tx\_Ctrl} + D_{PHY} + D_{PMD\_Rx\_Ctrl} + D_{PMI\_Rx\_Ctrl} \quad (5.13)$$

D_{Signal_Ctrl} is larger than D_{Signal_Grant} since the PMD can detect a *Grant* signal faster than any other link control signal. One can further observe, the maximum interrupt time D_{it_L1} is achieved when the network is not fully loaded since the hub in Figure 5.4 runs idle for a short time after serving the normal priority packet from Node 2. d_1 is thus larger than the worst-case overhead D_{pp_L1} determined in Section 5.2.1 since it also includes the time: $D_{Tx\_Data} + D_{Req\_H}$ in which the hub runs idle.

Equations 5.9 - 5.13 and the results received in Section 5.2.1, enable us to compute the numerical values for the interrupt time D_{it_L1} to be considered in the admission control for a single hub network. Example results are provided in Table 5.8 in the following section.

5.2.5 The Interrupt Time in Multi-Hub Networks

The results received for the single hub network can be generalized for higher cascaded 802.12 networks. To see this, we first describe the packet transmission model and derive the interrupt time for the Level-2 network. We then look at a generalization for higher cascaded topologies. At the end of this section we discuss measurement results achieved for the interrupt time in test networks with four different cascading levels.

Figure 5.5 shows the signalling that are required for pre-empting the normal priority service in a Level-2 cascaded network. The same worst case conditions as in the single hub network apply. We further omit the same signalling details as listed for Figure 5.4. The interrupt time is analysed in respect to Node 1 which requests the transmission of a high priority data packet. The two other nodes in the setup, Node 2 and Node 3, again only use the normal priority service. Note that the space between the upper two horizontal lines in Figure 5.5 again represents two links: *L4* and *L5*.

Comparing the model in Figure 5.5 with the model used for the single hub network then we can observe that the maximum interrupt time D_{it_L2} now includes the transmission times for *four* normal priority data packets. These are sent by Node 2 and Node 3. This occurs when the high priority request (Req_H) is only able to travel across a single UTP link before it is delayed by a normal priority data packet. At the same time, the network control toggles between the Root hub and Hub 3.

In the worst case, the network control is passed to Hub 3 just before the Req_H signal from Node 1 reaches the Root hub. The Root hub must then first regain the network control before the high priority request from Node 1 can be granted. For this, the Root hub sends a special link control signal to Hub 3. This signal is called *Enable-High-Only* (*Ena_HO*) and used to pre-empt the normal priority service. When Hub 3 detects the Ena_HO signal, it finishes the processing of the current normal priority packet and returns the network control. Afterwards, the Root hub passes the control to Hub 2, so that the high priority request from Node 1 can be served.

In higher cascaded topologies, a hub receiving Ena_HO from a higher level hub might have to pass the signal on when the network control is currently at a hub that is located further down in the topology tree. Furthermore, if a hub receives Ena_HO while serving high priority requests, it may first finish its current high priority service round before it returns the network control to the upper level hub.

The data flow in Figure 5.5 starts when the Root hub forwards a data packet towards Node 2 and Node 3. This might have come from Node 1 or another network node (not shown) directly connected to the root hub. At the same time, we assume that Node 2 has a pending normal priority service request. Both packets are served by the network in the same way as described for the first two data packets in Figure 5.2 in Section 5.2.2. The overhead associated with the normal priority packet from Node 2 is the worst-case delay D_{pp_L2} for this topology, as discussed in Section 5.2.2.

We then assume that the MAC of Node 1 in Figure 5.5 runs idle. As in the single hub case, the interrupt time becomes maximum when a new high priority request is made at Node 1 instantly after the Incoming signal was detected. Since the Incoming signal must travel across two links before it can arrive at Node 1, we receive for the overhead d_2 to be considered in D_{it_L2} for the first normal priority packet:

$$d_2 = D_{pp\_L2} - 2 \cdot D_{Incom} \tag{5.14}$$

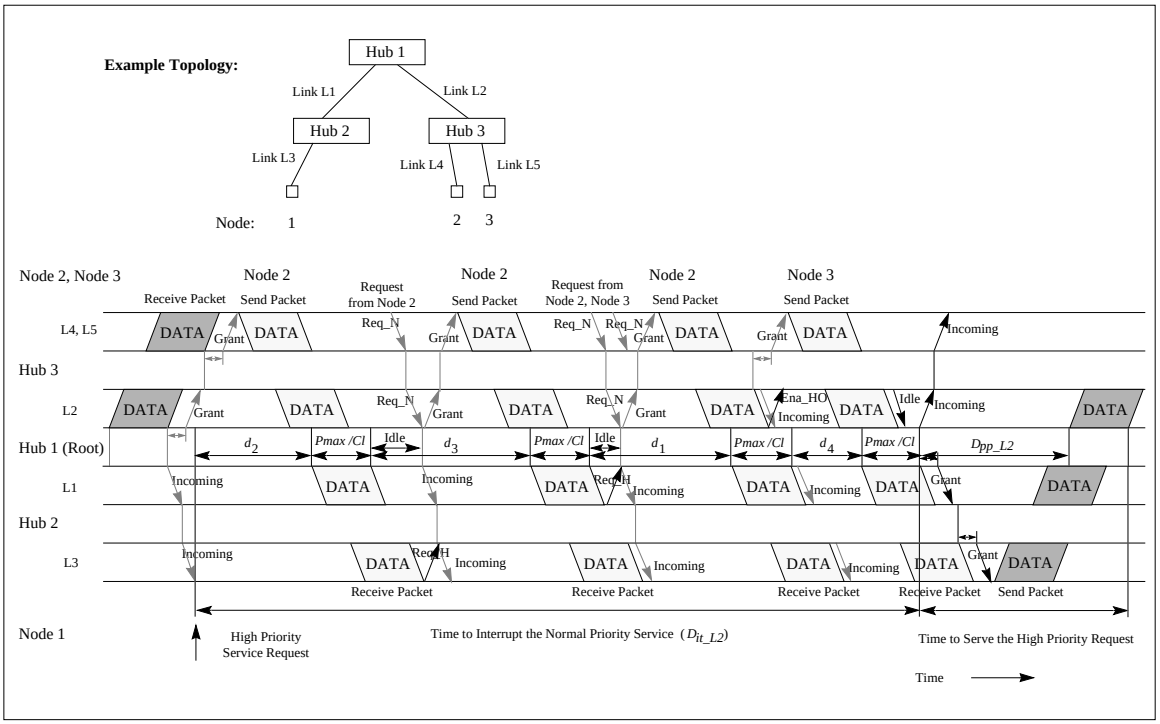


Figure 5.5: The Model for Computing the Worst-Case Interrupt Time in a Level-2 Cascaded Network using a UTP Physical Layer.

After the Root hub has forwarded the first packet from Node 2, it runs idle until it receives another normal priority service request from Node 2. This request could also be from another node connected to any Level-2 hub other than Hub 2. The request is instantly granted as shown in Figure 5.5. For this, the Root hub hands the network control to Hub 3 and, at the same time, sends Incoming to Hub 2. The worst case in respect to D_{it_L2} occurs when the Req_H from Node 1 arrives at Hub 2 at the same time as the Incoming signal from the Root Hub. In this case, the UTP PMD of Hub 2 does not pass the request on to the Root hub. If the Incoming had however arrived later at Hub 2, then the Req_H would have travelled further across link L1 to the Root hub. The overhead to be considered for the second data packet from Node 2 is denoted by d_3. It is larger than D_{pp_L2} since it also contains the time in which the Root hub runs idle. By using the delay components computed in Section 5.2.1 and Section 5.2.4 for a single UTP link, we receive for d_3:

$$\begin{aligned} d_3 \;=\; & D_{Tx\_Data} + D_{RMAC\_Data} + D_{Tx\_Data} + \\ & D_{Req\_H} - D_{Incom} + 2 \cdot D_{Signal\_Grant} + \\ & D_{Tx\_Data} + D_{RMAC\_Data} + D_{Tx\_Data} + D_{RMAC\_Data} \end{aligned} \tag{5.15}$$

When the Root hub has forwarded the data packet from Node 2, it again runs idle. The idle time is equal to the idle time observed for the single hub case. Node 2 and Node 3 then request the transmission of a normal priority packet by signalling Req_N to Hub 3. As in the single hub case, the worst case occurs when the Req_H signal from Hub 2 arrives at the Root hub just after the normal priority request from Node 2 has been granted. The Ena_HO signal is not transmitted across link L2 before the data packet from Node 2 has been fully received at the Root hub. From Figure 5.5 thus follows for the overhead d_1 to be considered for the third packet from Node 2:

$$d_1 = D_{Tx\_Data} + D_{Req\_H} + 2 \cdot D_{Signal\_Grant} + D_{Tx\_Data} + D_{RMAC\_Data} + D_{Tx\_Data} + D_{RMAC\_Data} \tag{5.16}$$

After Hub 3 has forwarded the data packet from Node 2, it keeps the network control and serves the normal priority request from Node 3. The Ena_HO signal from the Root hub always arrives at Hub 3 after this decision has been made. The network control is thus not returned until the normal priority data packet from Node 3 has been fully repeated. The corresponding packet overhead d_4 can be as long as the maximum delay in a single hub network, since Node 3 did also have to receive the preceding multicast data packet from Node 2 (which is however not illustrated in Figure 5.5). For the worst case, it thus follows from our considerations in Section 5.2.1:

$$d_4 \;=\; D_{pp\_L1} \tag{5.17}$$

The normal priority service is pre-empted when the Root hub has regained the network control from Hub 3. The network then serves the high priority request from Node 1. The signalling carried out

for this request is the same as discussed for the data packets in Figure 5.2. If we now assume a link speed of C_l, and that Node 2 and Node 3 sent normal priority data packets of maximum size P_{max} then we receive from Figure 5.5 for the worst-case interrupt time D_{it_L2} in a Level-2 cascaded network:

$$D_{it\_L2} \leq 4 \cdot \frac{P_{max}}{C_l} + d_2 + d_3 + d_1 + d_4 \tag{5.18}$$

where d_2, d_3, d_1 and d_4 are the results received with the Equations 5.14 to 5.17, respectively.

Generalization

We made the same considerations as in Figure 5.5 for the Level-3 and the Level-4 cascaded network. This is however omitted here since we can find the generalization without explicitly deriving these results in this thesis. If we consider the corresponding cascading level in the results received for the Level-1, Level-2 and Level-3 topology, then we have for the interrupt times:

$$D_{it\_L1} \leq 2 \cdot \frac{P_{max}}{C_l} + d_2(1) + d_1(1) \tag{5.19}$$

$$D_{it\_L2} \leq 4 \cdot \frac{P_{max}}{C_l} + d_2(2) + d_3(2) + d_1(2) + d_4(2) \tag{5.20}$$

$$D_{it\_L3} \leq 6 \cdot \frac{P_{max}}{C_l} + d_2(3) + d_5(3) + d_3(3) + d_1(3) + d_4(3) + d_6(3) \tag{5.21}$$

It can be observed that the maximum number of normal priority data packets which are served by the network before the normal priority service is pre-empted is equal to the number of UTP links in the data path. In a Level-5 cascaded topology, as many as *ten* normal priority data packets can thus be served by the Root hub before a high priority request is granted. The per-packet overheads in the Equations 5.19, 5.20 and 5.21 are computed using the functions $d_i(N)$, where N is the cascading level and i a packet index. These functions provide a generalized way to compute the per-packet overhead in all topologies. $d_2(2)$ and $d_4(2)$ for example provide the overhead of the first and fourth normal priority data packet in D_{it_L2}, and are thus identical with Equation 5.14 and Equation 5.17, respectively. If we generalize the Equations 5.19, 5.20 and 5.21 then we receive for the Level-N cascaded topology:

$$D_{it\_LN} \leq 2N \cdot \frac{P_{max}}{C_l} + \sum_{i=1}^{2N} d_i(N) \tag{5.22}$$

where $1 \le N \le 5$. The generalization of the per-packet overheads for packets with an even index *i* in Equation 5.22 is straightforward. Observing the results for the Level-1, Level-2, Level-3 and Level-4 topologies, we obtain for the corresponding functions $d_i(N)$:

$$d_2(N) = D_{pp\_LN} - N \cdot D_{Incom} \quad (5.23)$$
$$d_4(N) = D_{pp\_L\,(N-1)} \quad (5.24)$$
$$d_6(N) = D_{pp\_L\,(N-2)} \quad (5.25)$$
$$d_8(N) = D_{pp\_L\,(N-3)} \quad (5.26)$$

where D_{pp_LN}, $D_{pp\_L(N-1)}$, $D_{pp\_L\,(N-2)}$ and $D_{pp\_L\,(N-3)}$ denote the worst case per-packet overhead in the Level-N, Level-N-1, Level-N-2 and Level-N-3 cascaded network, respectively. The results for the functions $d_i(N)$ with an odd index *i* are more complicated since they also describe the idle times which we could for example observe for the Root hub in Figure 5.5. We further made two worst-case assumptions for all topologies. These are: (1) that each Grant signal is delayed by a preceding idle burst (I_BST), and (2) that all per-packet overheads are at least as big as D_{pp_LN}. The first condition assumes that the receiver of the next Grant was always also one of the receivers of the last data packet. Since this assumption is however not always true as can be observed in Figure 5.5, this insignificantly increases the computed upper bound. It however enables a simple generalization of the results for all cascading level. By adding these two assumptions to the results received for the Level-1, Level-2, Level-3 and Level-4 topologies, we have:

$$\begin{aligned} d_1(N) = \; & MAX\,((D_{pp\_LN}), \\ & (D_{Tx\_Data} + D_{Req\_H} + \\ & N \cdot (I\_BST + D_{Signal\_Grant} + D_{Tx\_Data} + D_{RMAC\_Data}))) \end{aligned} \quad (5.27)$$

$$\begin{aligned} d_3(N) = \; & MAX\,((D_{pp\_LN}), \\ & (2 \cdot D_{Tx\_Data} + D_{RMAC\_Data} + D_{Req\_H} - D_{Incom} + \\ & N \cdot (I\_BST + D_{Signal\_Grant} + D_{Tx\_Data} + D_{RMAC\_Data}))) \end{aligned} \quad (5.28)$$

$$\begin{aligned} d_5(N) = \; & MAX\,((D_{pp\_LN}), \\ & (3 \cdot D_{Tx\_Data} + 2 \cdot D_{RMAC\_Data} + D_{Req\_H} - 2 \cdot D_{Incom} + \\ & N \cdot (I\_BST + D_{Signal\_Grant} + D_{Tx\_Data} + D_{RMAC\_Data}))) \end{aligned} \quad (5.29)$$

$$\begin{aligned} d_7(N) = \; & MAX\,((D_{pp\_LN}), \\ & (4 \cdot D_{Tx\_Data} + 3 \cdot D_{RMAC\_Data} + D_{Req\_H} - 3 \cdot D_{Incom} + \\ & N \cdot (I\_BST + D_{Signal\_Grant} + D_{Tx\_Data} + D_{RMAC\_Data}))) \end{aligned} \quad (5.30)$$

for the functions with an odd index *i* in Equation 5.22. The two additional functions for the Level-5 topology, $d_9(N)$ and $d_{10}(N)$, are straightforward to derive from the results for the lower cascaded topologies. This is thus omitted here.

In Figure 5.5, we could observe an idle time between subsequent normal priority data packets. We found that this idle time further increases in higher cascaded networks. However, it does not lead to a significant increase of the interrupt time because the propagation delay across a 200 m UTP link is small. Considering the numerical results computed for the Grant-, Incoming- and the Data signalling delay in Section 5.2.1 and Section 5.2.4 the impact is only in the order of a few microseconds. The maximum overhead is further not always achieved with a maximum idle time. In some cases, the maximum interpacket gap occurs when the normal priority request is instantly granted and the Grant signal is delayed by a preceding multicast data packet. In this case the overhead for the normal priority data packet becomes equivalent to D_{pp_LN} as we considered in Equations 5.27 to 5.30.

Using Equation 5.22, the Equations 5.23 - 5.30, and the delay components derived in Section 5.2.1 and Section 5.2.4 we computed the worst-case interrupt time for all valid cascading level *N*. The results for different UTP cable length are shown in Table 5.8.

UTP-Cable Length	Network Cascading Level N				
	1	2	3	4	5
5 m	259.22 µs	545.45 µs	861.34 µs	1208.57 µs	1586.58 µs
100 m	261.92 µs	554.11 µs	878.07 µs	1236.06 µs	1628.23 µs
200 m	264.77 µs	563.23 µs	895.74 µs	1265.70 µs	1673.11 µs

Table 5.8: Normal Priority Service Interrupt Times in Cascaded Networks using UTP Cabling.

Measurement Results for the Interrupt Time in Cascaded Networks

We measured the interrupt time in test networks with a Level-1, Level-2, Level-3 and Level-4 topology. This was based on the delay measurement approach described in Section 3.6 in Chapter 3. The cascaded test topologies were identical to the ones used for the throughput measurements in Section 4.3.1 in Chapter 4. High priority traffic was generated by the Measurement Client. It generated data packets at a constant bit rate with a low mean - about 0.56 Mbit/s. The low data rate ensured that there was never more than a single high priority packet in transit through the network. This was additionally checked in each measurement. We further used 10 Normal Priority Traffic Clients which imposed multicast traffic at a total constant bit rate ranging from 0 to 100 Mbit/s. All data packets had a size of 1500 bytes to enforce worst-case results. The measurement interval for each sample was 1 minute which corresponds to about 3000 data packets transmitted by the Measurement Client. The incremental step of the normal priority network load was 500 kbit/s. In contrast to the setup in Section 3.6, we did not use High Priority Traffic Clients during these experiments.

The Measurement Client and the hubs were interconnected using 100 m UTP cabling. To link the Traffic Clients to the hubs, we however used 5 m cables of the same type, since we did not have a sufficient large number of 100 m cables available. This introduced a small difference between the measurement setup and the theoretical model. This is however not significant since the overhead plus propagation delay for a 5 m versus a 100 m UTP cable only differ by a maximum of 0.542 µs .

Figure 5.6 shows the maximum- and the minimum end-to-end delays observed by the Measurement Client. We only labelled the maximum delay curves. All results are bounded. For each topology, the time difference between the corresponding maximum- and minimum delay is the time it takes to interrupt the normal priority data transmission within that topology. This is illustrated in Figure 5.6.

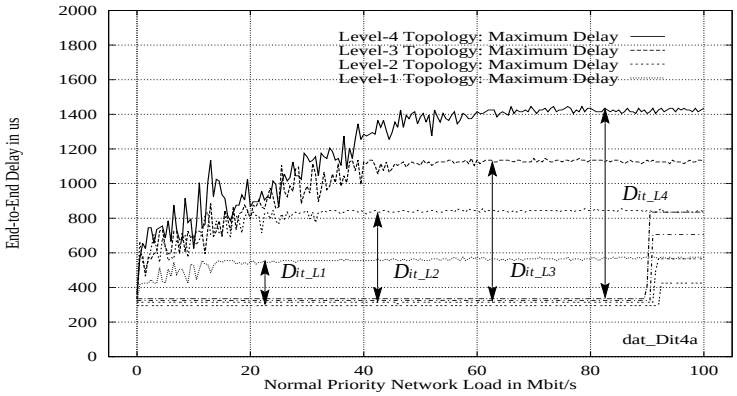


Figure 5.6: Measured Interrupt Times in Cascaded Networks using a UTP Physical Layer.

The minimum delay in a single hub network is about 300 µs. This slightly increases in higher cascaded topologies due to the data transmission and signalling across a longer data path, which for example included 7 repeating hubs and 8 links in the Level-4 cascaded test network. We measured a minimum delay of about 335 µs for the Level-4 topology. The maximum delay observed in the single hub network is 570 µs. This increases with each cascading level by about 240 µs plus overhead for the two normal priority data packets transmitted. We measured a maximum delay of 855 µs, 1135 µs and 1445 µs for the Level-2, Level-3 and Level-4 topology, respectively. The resulting normal priority service interrupt times (D_{it_L1}, D_{it_L2}, D_{it_L3}, D_{it_L4}) are: 275 µs, 540 µs, 810 µs and 1110 µs, respectively.

These results confirm the theoretical bounds shown in Table 5.8 and implicitly, the models used to computed them. The only measurement result that exceeds its corresponding bound is the result for the single hub network (275 µs versus 261.92 µs). We explain this with inaccuracies introduced by the measurement process. The theoretical bounds for higher cascaded networks are sufficiently large and conservative such that the measurement error is covered.

5.2.6 The Interrupt Time in Half-Duplex Switched Links

As in single hub networks, the worst-case interrupt time on half-duplex links is equivalent to the transmission time of *two* data packets across the physical medium plus the corresponding packet overheads. Figure 5.7 shows the signalling and packet transmission for this case. The worst case occurs when the switch operating in 802.12 MAC mode (Switch 2) requests the high priority service while the switch possessing the network control (Switch 1) is transmitting several normal priority data packets.

To compute an upper bound on the interrupt time, similar considerations as previously discussed for the single hub network can be made. The half-duplex case however differs in respect to the per-packet overhead to be considered for the normal priority data packets sent by Switch 1. We may thus use Equation 5.9 in Section 5.2.4 for the computation, but have to determine the half-duplex link specific results for the parameters d_2 and d_1 in this equation. The first data packet in Figure 5.7 is transmitted from Switch 1 to Switch 2. The associated overhead is the interpacket gap: IPG + D_IPG. This assumes that just before the transmission of this packet, Switch 1 had sent another data packet (not shown) to Switch 2. From Figure 5.7, we thus receive: $d_2 = IPG + D\_IPG - D_{Incom}$ for the overhead to be considered in the computation of the interrupt time.

As in the example for the single hub network, the RMAC of Switch 1 runs idle after it completed the transmission of the first normal priority data packet. On its receipt at Switch 2, the UTP sublayer instantly signals Req_H indicating the demand for the high priority service. The worst case occurs if the Req_H signal arrives at Switch1 just after another normal priority service request has been granted. This case is shown in Figure 5.7. The high priority request from Switch 2 is thus not served before the normal priority packet transmission from Switch 1 has been completed. The maximum idle time for the RMAC at Switch 1 is: $D_{Tx\_Data} + D_{Req\_H}$. Considering additionally the RMAC delay D_{RMAC_Data}, we receive a maximum overhead of: $d_1 = D_{Tx\_Data} + D_{Req\_H} + D_{RMAC\_Data}$ for the second data packet from Switch 1.

It remains to remark that for all cable lengths supported by the standard, the results for d_1 will always be larger than the interpacket gap: IPG + D_IPG. Furthermore, the per-packet overhead for the following high priority packet from Switch 2 can be as large as the worst case in a single hub network.

By using the results for both parameters, d_2 and d_1 in Equation 5.9, we receive for the worst-case interrupt time D_{it_HD} on a half-duplex UTP link:

$$D_{it\_HD} \leq 2 \cdot \frac{P_{max}}{C_l} + IPG + D\_IPG - D_{Incom} + D_{Tx\_Data} + D_{Req\_H} + D_{RMAC\_Data} \tag{5.31}$$

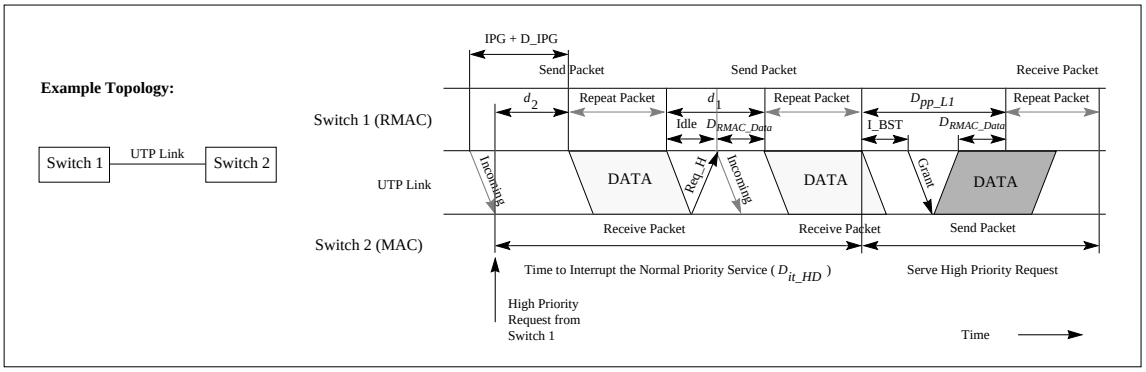


Figure 5.7: The Model for Computing the Worst-Case Interrupt Time on a Half-Duplex Link using a UTP Physical Layer.

Numerical results are shown in Table 5.9. As expected these are lower than the bounds computed for the single hub network. The impact of the UTP cable length on the results also decreases due to the reduced signalling overhead required for controlling the medium access. Furthermore, the results for D_{it_HD} are independent of the packet addressing mechanism used. The data packets sent by both switches in Figure 5.7 may thus carry a unicast, multicast or broadcast destination address.

UTP Cable Length	D_{it_HD}
5 m	252.13 µs
100 m	252.67 µs
200 m	253.24 µs

Table 5.9: Normal Priority Service Interrupt Times on Half-Duplex Switched UTP Links.

Measurement Results for the Interupt Time in Half-Duplex Switched Links

To confirm the theoretical analysis, we also measured the normal priority interrupt time on a half-duplex switched link. This was based on the same fundamental measurement setup and methodology as used in the previous section to achieve the equivalent results in cascaded network topologies.

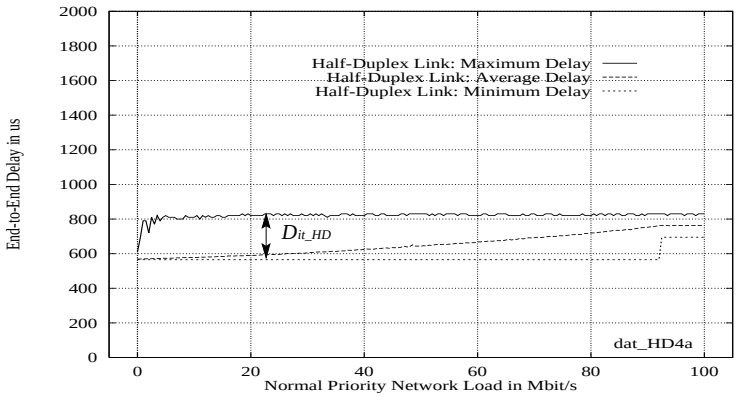


Figure 5.8: Measured Interrupt Time on a Half-Duplex Switched UTP Link.

The two LAN interfaces of the Measurement Client were connected to Switch 1 and Switch 2 in the example topology shown in Figure 5.7. All high priority packets generated by the Measurement Client entered the test link at Switch 2 and were returned to it from Switch 1 after their transmission on that link. The setup further included 4 Normal Priority Traffic Clients which we used for generating the normal priority traffic. Each of these Clients was connected to Switch 1 via a separate 5 m UTP cable. All normal priority data packets were thus transmitted from Switch 1 to Switch 2. Filter entries in both switches ensured that cross traffic was not forwarded through the ports connecting the Measurement Client. The details of the measurement process such as the measurement time, load range, incremental load step, etc. were identical to the parameters described for cascaded networks. Finally, the test link between Switch 1 and Switch 2 consisted of 100 m UTP cable.

Figure 5.8 shows the maximum-, the average- and the minimum end-to-end delay measured by the Measurement Client. In comparison to the single hub network, the results for the minimum delay increased by about 270 µs. We measured an absolute value of 565 µs. This offset is mainly caused by: (1) the store-and forward approach used within Switch 1 and Switch 2 - resulting in 120 µs delay in each switch, and (2) the time it takes to transfer a data packet across the internal switch bus. The latter consumes about 12 µs due to the bus speed of 1 Gbit/s. For the maximum delay, we measured a maximum of 830 µs. This provides 265 µs for the worst case normal priority interrupt time which closely matches the theoretical result for 100 m UTP cable in Table 5.9.

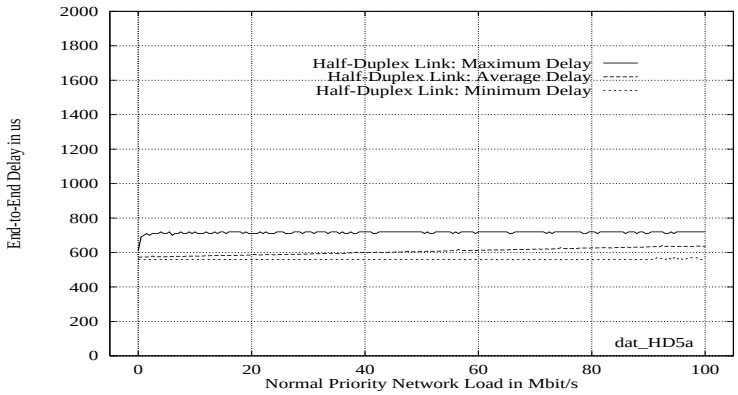


Figure 5.9: End-to-End Delay in a Setup with *Switch 2* operating in RMAC Mode.

After measuring the interrupt time, we repeated the experiment using the same setup but with Switch 2 possessing the network control over the test link. Switch 2 thus operated in 802.12 RMAC mode, Switch 1 in MAC mode. Since in this scenario, all high priority packets enter the test link at a switch (Switch 2) that controls the link access, the interrupt time should theoretically never be larger than one packet transmission time plus overhead. This is confirmed by the results of this experiment shown in Figure 5.9. We measured a maximum difference of 150 µs between the results for the maximum- and minimum end-to-end delay.

5.3 Performance Parameters for the Fibre-Optic Physical Layer

To compute the per-packet overhead and the interrupt time in 802.12 networks with a fibre-optic physical layer, we can re-use some of the packet transmission models introduced in Section 5.2 for UTP. Fibre optic technology however implies two properties which differ significantly from the features provided by the UTP sublayer. These are the support for: (1) longer link distances between hosts, hubs and switches in the network, and (2) a *dual simplex* operation across fibre-optic links. Both properties need to be considered in the computation of the performance parameters. The first may substantially increase the signal propagation delay and thus the worst-case per-packet overhead. The latter reduces the impact of normal priority cross traffic on the interrupt time. It remains to remark that we were not able to take measurement results in fibre-optic networks. This was due to a lack of sufficient access to isolated networks using this technology.

5.3.1 The Per-Packet Overhead in Cascaded Networks

The per-packet overhead in cascaded networks can be determined based on the same worst-case model as used for the UTP physical layer. If we assume fibre-optic links with a maximum length of 2 km, multi-hub topologies with a cascading level of up to $N = 2$ are supported by the standard. This enables a network administrator to build a shared network with a maximum distance of 4 km between each network node and the Root hub. Networks with a higher cascading level can be formed when links of smaller length are used. We however focus on the $1 \leq N \leq 2$ case because we believe that this implies network topologies of sufficient size and physical extension. In real 802.12 LANs, fibre-optic links are more likely to be employed to interconnect switches. Shared workgroup segments might then be linked to these switches using the more cost-effective UTP physical layer.

The maximum per-packet overhead in fibre-optic cascaded networks occurs under the same conditions as in the equivalent networks using UTP cabling. This is due to identical signalling characteristics exhibited by both physical layer technologies in this case. We may thus use Equation 5.7 defined in Section 5.2.2 for the computation of the overhead, but must consider fibre-optic specific results for the data transmission delay: D_{Tx_Data} in this equation. The other parameters such as the idle burst time (I_BST) or the decoding delay (D_{RMAC_Data}) are RMAC specific and thus valid for any physical layer. The precise breakdown of D_{Tx_Data} for a fibre-optic link is given in Table 5.10. The delay components for the MAC and PMI sublayers are identical to the ones in Table 5.4. The propagation delay on the physical medium is given for 2 km multi-mode fibre with a typical refractive index of n = 1.5.

Sublayer	Comments	Worst Case Delay	Reference Section in [ISO95]
MAC (Source)		-	12.6.3.4 12.6.4.1
PMI	$D_{PMI\_Tx\_Data}$ Addition of the preamble pattern (48 BT): Addition of the Starting Delimiter (12 BT): Propagation delay for data (3 BT):	63 BT	 14.4.2.3.2 14.4.2.3.3 14.3.4
PMD	$D_{PMD\_Tx\_Data}$ Maximum propagation delay within PMD.	12 BT	18.5.3
PHY (Link)	D_{PHY} Propagation delay on l = 2 km fibre, n = 1.5, $c = 2.998 \cdot 10^8 m/s$.	10.0 µs	$D_{PHY} = (l \cdot n)/c$
PMD	$D_{PMD\_Rx\_Data}$ Data recovery delay.	12 BT	18.6.5
PMI	$D_{PMI\_Rx\_Data}$ Synchronization, data decoding (8 BT): Propagation delay within the PMI (3 BT):	11 BT	 14.4.4 as 14.3.4
MII -> MII (Hub)	$D_{MII\_Rx\_Tx\_Data}$ Transmit delay from the receiving MII to the sending MII in the RMAC:	4.5 µs	12.9.7.2

Table 5.10: Breakdown of the Data Transmission Delay for a Fibre-Optic Physical Layer.

Numerical results for D_{Tx_Data} are computed using Equation 5.3 from Section 5.2.1 with the delay components in Table 5.10. Example results for the per-packet overhead are shown in Table 5.11. They were computed using Equation 5.7, where $1 \le N \le 2$.

Length of the Fibre-Optic Cable	Network Cascading Level N	
	1	2
100 m	10.37 µs	21.97 µs
1000 m	19.37 µs	39.97 µs
2000 m	29.37 µs	59.97 µs

Table 5.11: Per-Packet Overhead D_{pp_LN} for Fibre-Optic Cascaded Networks.

We can observe that for short fibre-optic cables, the impact of the propagation delay on the per-packet overhead is small. The results for 100 m in Table 5.11 for example, almost match the equivalent results computed for a UTP physical layer. For long distances however the large values for D_{PHY} dominate the per-packet overhead and substantially increases the numerical results received.

5.3.2 The Per-Packet Overhead in Half-Duplex Switched Links

To compute the maximum per-packet overhead for fibre-optic half-duplex switched links, we identified two specific cases which we discuss in the following. First, if the link length l is below a threshold L, then the same worst-case conditions apply as discussed for UTP in Section 5.2.3. In this case, we may use Equation 5.8 and the result for D_{pp_L1} in Table 5.11 for the computation of the per-packet overhead.

If however $l \ge L$, then the worst case is achieved when the node operating in MAC mode (Switch 2) is continuously transmitting data packets to the node possessing the network control (Switch 1). This case is illustrated in the Time-Space diagram in Figure 5.10. The data throughput decreases because the RMAC at Switch 1 must send a Grant signal for every single data packet to be transmitted by Switch 2. For short links, this does not have a significant impact on the per-packet overhead and explains why Equation 5.8 is valid for $l < L$. The Grant signalling delay may however increase substantially on long distance fibre-optic links due to the large propagation delay.

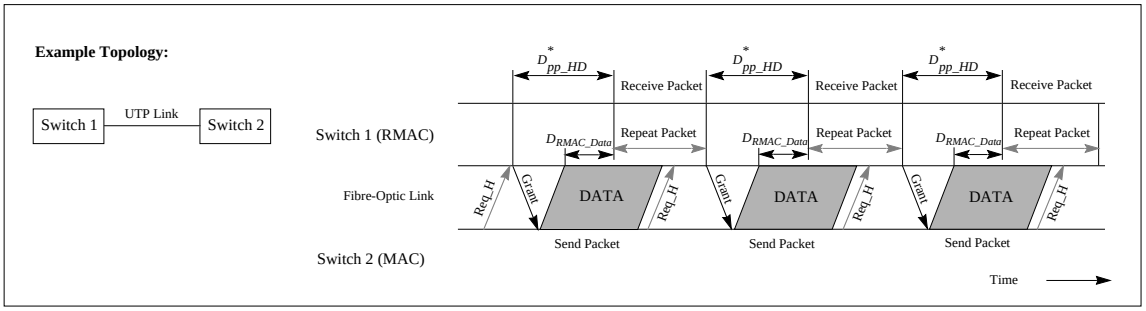


Figure 5.10: Worst-Case Signalling on a Fibre-Optic Half-Duplex Switched Link for $l \ge L$.

It can easily be seen that for each data packet in Figure 5.10, we have a maximum per-packet overhead of: $D^{*}_{pp\_HD} \leq D_{Signal\_Grant} + D_{Tx\_Data} + D_{RMAC\_Data}$. We are however interested in the worst case for all cable lengths. Using Equation 5.8 for the case $l < L$, we receive for this:

$$D_{pp\_HD} = MAX(((IPG + D\_IPG + D_{pp\_L1})/2)\ ,\ (D_{Signal\_Grant} + D_{Tx\_Data} + D_{RMAC\_Data})) \tag{5.32}$$

Alternatively, we can consider both cases separately. We then have:

$$D_{pp\_HD} = \left\{ \begin{array}{ll} (IPG + D\_IPG + D_{pp\_L1})/2 & \text{if} \quad l < L \\ D_{Signal\_Grant} + D_{Tx\_Data} + D_{RMAC\_Data} & \text{if} \quad l \geq L \end{array} \right\} \tag{5.33}$$

The Grant signalling delay D_{Signal_Grant} in these equations is computed using the parameters in Table 5.12 and Equation 5.2 in Section 5.2.1. To determine the length L in Equation 5.33, we set:

$$(IPG + D\_IPG + D_{pp\_L1})/2 = D_{Signal\_Grant} + D_{Tx\_Data} + D_{RMAC\_Data} \tag{5.34}$$

Using Equation 5.1 from Section 5.2.1 in Equation 5.34 then provides:

$$D_{Signal\_Grant} = (I\_BST + IPG + D\_IPG - D_{RMAC\_Data})/2 \tag{5.35}$$

This uses the fact that we always have: $IPG + D\_IPG \leq D_{Tx\_Data} + I\_BST + D_{Tx\_Data} + D_{RMAC\_Data}$ for the case $l \geq L$. If we then substitute D_{Signal_Grant} in Equation 5.35 with Equation 5.2 and use the term $D_{PHY} = (L \cdot n)/c$ for the physical layer propagation delay, where n and c are the refractive index and the speed of light, respectively, then after reordering, we receive for the length L in Equation 5.33:

$$\begin{aligned} L = \frac{c}{n} ((I\_BST + IPG + D\_IPG - D_{RMAC\_Data})/2 \ - \\ (D_{PMI\_Tx\_Ctrl} + D_{PMD\_Tx\_Ctrl} + D_{PMD\_Rx\_Grant} + D_{PMI\_Rx\_Ctrl})) \end{aligned} \tag{5.36}$$

Using the delay components in Table 5.12, we receive a numerical result of L = 342.87 m.

Sublayer	Comments	Worst Case Delay	Reference Section in [ISO95]
RMAC (Hub)		-	12.6.3.4 12.6.4.1
PMI	$D_{PMI\_Tx\_Ctrl}$ Control signal encoding, (control signals do not have a preamble).	4 BT	14.3.1
PMD	$D_{PMD\_Tx\_Ctrl}$ Propagation delay within the PMD.	12 BT	18.5.3
PHY (Link)	D_{PHY} Propagation delay on l = 2 km fibre, n = 1.5, $c = 2.998 \cdot 10^8 m/s$.	10.0 µs	$D_{PHY} = (l \cdot n)/c$
PMD	$D_{PMD\_Rx\_Grant}$ Grant signal detection.	12 BT	18.6.5
PMI	$D_{PMI\_Rx\_Ctrl}$ Control signal mapping.	4 BT	14.3.2 14.3.3
MAC (Receiver)		-	

Table 5.12: Breakdown of the Grant-Signalling Delay for a Fibre-Optic Physical Layer.

Finally, Table 5.13 provides selected numerical results for the per-packet overhead on a half-duplex switched link. These were computed using Equation 5.33 with the results shown in Table 5.10 and Table 5.12.

Length of the Fibre-Optic Cable	D_{pp_HD}
100 m	8.68 µs
1000 m	16.47 µs
2000 m	26.47 µs

Table 5.13: Per-Packet Overhead D_{pp_HD} for Fibre-Optic Half-Duplex Switched Links.

5.3.3 The Interrupt Time in Cascaded Networks

The dual simplex operation of the fibre-optic physical layer significantly simplifies the computation of the interrupt time for cascaded networks. This is because the sublayer is able to transmit control informations across a link while it is receiving a data packet. The signalling of a high priority service request can therefore not be blocked by incoming normal priority data packets as we could observe for UTP in Section 5.2.5.

Figure 5.11 shows the worst-case conditions in a Level-2 cascaded network. Illustrated is the transmission of three data packets: the first two have normal priority, the last packet has high priority. All three packets are assumed to be multicast. The first normal priority packet is transmitted from the

Root hub to Node 2. It could for example have come from another network node (not shown) connected directly to the Root hub or be sent by Node 1. After this packet is forwarded to Hub 3, the Root hub passes the network control on to Hub 3 so that the normal priority data packet from Node 2 can be served. This assumes that Node 2 has a pending normal priority service request. The Req_N signal corresponding to this request has however been omitted in Figure 5.11.

The high priority service is requested by Node 1. We obtain worst-case conditions when: (1) the corresponding service request (Req_H) arrives at the Root hub just after this hub made the decision to serve the normal priority request from Node 2, and (2) the normal priority data packet from Node 2 is of maximum size and forwarded with a maximum per-packet overhead. In Figure 5.11, we find that these conditions cause a maximum interrupt time of: $D_{it\_L2} = 2 \cdot D_{Req\_H} + D_{pp\_L2} + P_{max}/C_l$. In contrast to cascaded networks with a UTP physical layer, only one maximum sized normal priority data packet can be served by the network before the high priority request from Node 1 is guaranteed to be granted.

Similar considerations can be made for the single hub network. These are left out here because the result can implicitly be derived from Figure 5.11. For the *Level-N* cascaded network with fibre-optic links, we thus have for the worst-case normal priority interrupt time:

$$D_{it\_LN} = N \cdot D_{Req\_H} + D_{pp\_LN} + P_{max}/C_l \tag{5.37}$$

where $1 \leq N \leq 2$. The service request signalling delay D_{Req_H} can be computed using Equation 5.12 and 5.13 from Section 5.2.4 combined with the results in Table 5.14. Table 5.14 contains the breakdown of the signalling delay for control signals across a fibre-optic physical layer. The per-packet overhead D_{pp_LN} is determined using Equation 5.7 and the results in Table 5.10.

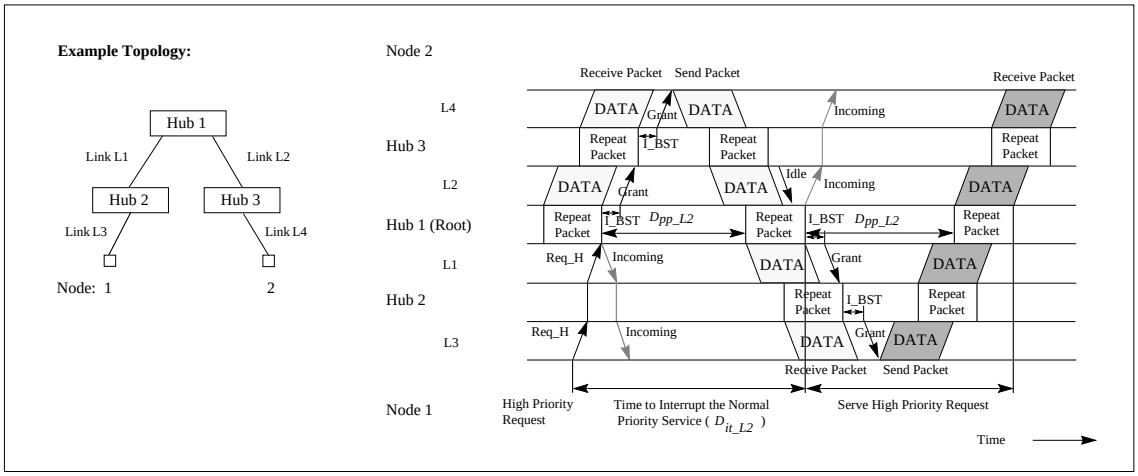


Figure 5.11: The Model for Computing the Worst-Case Interrupt Time in a Level-2 Cascaded Network using a Fibre-Optic Physical Layer.

Sublayer	Comments	Worst Case Delay	Reference Section in [ISO95]
RMAC (Hub)		-	12.6.3.4 12.6.4.1
PMI	$D_{PMI\_Tx\_Ctrl}$ Control signal encoding, (control signals do not have a preamble).	4 BT	14.3.1
PMD	$D_{PMD\_Tx\_Ctrl}$ Max. propagation delay within the PMD.	12 BT	18.5.4.2
PHY (Link)	D_{PHY} Propagation delay on l = 2 km fibre, n = 1.5, $c = 2.998 \cdot 10^8 m/s$.	10 µs	$D_{PHY} = (l \cdot n)/c$
PMD	$D_{PMD\_Rx\_Ctrl}$ Control signal recovery and decoding.	24 BT	18.6.1
PMI	$D_{PMI\_Rx\_Ctrl}$ Control signal mapping.	4 BT	14.3.2
MAC (Receiver)		-	

Table 5.14: Breakdown of the Delay required for Signalling the Control Signals Req_H, Req_N and Incoming across a single Fibre-Optic Link.

Numerical results for D_{it_LN} in cascaded fibre-optic networks with $1 \le N \le 2$ are shown in Table 5.15. These are based on Equation 5.37 and the results for D_{pp_LN} in Table 5.11.

Length of the Fibre-Optic Cable	Network Cascading Level N	
	1	2
100 m	132.34 µs	145.90 µs
1000 m	145.84 µs	172.90 µs
2000 m	160.84 µs	202.90 µs

Table 5.15: Normal Priority Service Interrupt Times in Fibre-Optic Cascaded Networks.

5.3.4 The Interrupt Time in Half-Duplex Switched Links

To determine the maximum interrupt time for fibre-optic half-duplex switched links, we use the scenario in Figure 5.3 in Section 5.2.3. Shown are two switches, Switch 1 and Switch 2, connected via a half-duplex switched link. In the following discussion, we assume this link to be a fibre-optic link. Both switches in Figure 5.3 toggle between sending and receiving data packets. As in fibre-optic cascaded networks, it takes one packet transmission time plus signalling overhead to pre-empt the normal priority service on the half-duplex link. This is due to the dual simplex operation of the physical layer. All data packets sent by Switch 1 (RMAC) have a per-packet overhead of: $IPG + D\_IPG$. This can be also be observed in Figure 5.3. Assuming now the case that these data

packets are sent with normal priority and Switch 2 then requests the 802.12 high priority service, we find a maximum interrupt time of: $D_{Req\_H} + IPG + D\_IPG + P_{max}/C_l$ for Switch 2. The worst-case condition is achieved when the high priority request from Switch 2 arrives at Switch 1 instantly after the RMAC at Switch 2 decided to serve the next normal priority packet.

For all data packets from Switch 2 (MAC) in Figure 5.3, we have a per-packet overhead of D_{pp_L1}. This is based on the same considerations as made for UTP in Section 5.2.3. If we now assume the case that Switch 2 sends normal priority data packets and Switch 1 is requesting the high priority service then the interrupt time is bounded by: $D_{pp\_L1} + P_{max}/C_l$. Note that in this case, we do not have to consider a delay for the service request signalling at Switch 2 (D_{Req_H}), because Switch 2 contains the RMAC. The worst-case interrupt time D_{it_HD} is the maximum of both cases:

$$D_{it\_HD} \leq MAX((D_{Req\_H} + IPG + D\_IPG + P_{max}/C_l), (D_{pp\_L1} + P_{max}/C_l)) \quad (5.38)$$

where D_{Req_H} is computed using the Equations 5.12 and 5.13 combined with the results in Table 5.14. The per-packet overhead D_{pp_L1} is determined based on Equation 5.1 and 5.3 and the results in Table 5.10. If we use the numerical results for these parameters, then we find for all valid cable lengths that condition: $D_{Req\_H} + IPG + D\_IPG + P_{max}/C_l < D_{pp\_L1} + P_{max}/C_l$ holds. This then provides:

$$D_{it\_HD} \leq D_{pp\_L1} + P_{max}/C_l \quad (5.39)$$

Table 5.16 finally provides numerical results for the interrupt time computed from Equation 5.39.

Length of the Fibre-Optic Cable	D_{it_HD}
100 m	130.37 µs
1000 m	139.37 µs
2000 m	149.37 µs

Table 5.16: Normal Priority Service Interrupt Times on a Fibre-Optic Half-Duplex Switched Link.

5.4 The Impact of the 802.5 Frame Format on the Performance Parameters

Since the 802.12 MAC signalling is independent of the size of the data packet to be transmitted, the per-packet overhead is the same for 802.3 and 802.5 frame formats. The equations 5.7, 5.8 and 5.32 may thus be applied in both cases. The results for the interrupt time however depend on the size of the normal priority data packets transmitted while the service is being interrupted. Valid results can

be computed by using the format specific maximum packet size P_{max} within the Equations 5.22, 5.31, 5.37 and 5.38.

Alternatively, the numerical results determined for the 802.3 frame format can be used to compute the corresponding upper bounds for the 802.5 frame format. For the results in Table 5.8 and Table 5.9 the mapping is performed using the formula:

$$D_{it}^{802.5,\,UTP} = D_{it}^{802.3,\,UTP} + \left(2N \cdot \left(\frac{P_{max}^{802.5}}{C_l} - \frac{P_{max}^{802.3}}{C_l}\right)\right) \tag{5.40}$$

where $D_{it}^{802.5,\,UTP}$ and $D_{it}^{802.3,\,UTP}$ denote the interrupt times for the UTP physical layer and the 802.5 and 802.3 frame format, respectively. $P_{max}^{802.5}$ and $P_{max}^{802.3}$ are the maximum data packet sizes for the two formats. The parameter N is the cascading level. Equation 5.40 follows from observations in Equation 5.22 and Equation 5.31. The mapping is performed by adding twice the difference between the link propagation times of a maximum size 802.5 and 802.3 data packet to the interrupt time for each cascading level. To map the results in Table 5.15 and Table 5.16 computed for networks with a fibre-optic physical layer, we receive the equivalent formula:

$$D_{it}^{802.5,\,F} = D_{it}^{802.3,\,F} + \left(N \cdot \left(\frac{P_{max}^{802.5}}{C_l} - \frac{P_{max}^{802.3}}{C_l}\right)\right) \tag{5.41}$$

In contrast to Equation 5.40, Equation 5.41 only adds a single difference of the link propagation times. This follows from observations in Equation 5.37 and Equation 5.38. Finally, note that both equations, 5.40 and 5.41, also apply to the half-duplex switched case when used with $N = 1$.

5.5 Summary

In this chapter we studied the details of the data transmission in 802.12 networks using UTP and fibre-optic physical layers. Considered were single hub-, multi-hub and half-duplex switched network topologies. We first found that the service properties enforced by the Demand Priority protocol, in particular: the packet service order, the priority access mechanism and the fairness, are maintained in all topologies even when the number of hubs and nodes in the shared network becomes very large. This property is most important for our resource allocation scheme since it will enable us to use the same scheduling process and the same admission control conditions for all 802.12 network topologies.

Networks with a different cascading level however differ in respect to the network performance. We identified two parameter, the per-packet overhead and the normal priority service interrupt time, to describe the worst-case performance as required for a guaranteed service. The admission control conditions will thus differ by the cascading level specific values to be used for the per-packet overhead and the interrupt time, when applied to different network topologies.

Most of the chapter was then dedicated to the analysis of the data transmission and the derivation of upper bounds for the per-packet overhead and the normal priority service interrupt time. In UTP based cascaded networks, we found that the per-packet overheads increases rapidly with the cascading level, whereas in particular for the Level-1 and Level-2 topologies, the UTP cable length did not have such a drastic impact. The interrupt times for a UTP physical layer may also be significant. We observed a range from 252 µs for a half-duplex link of 5 m length, up to a maximum of 1.67 ms for a Level-5 cascaded network using 200 m UTP cabling. Measurements in our test network confirmed the results for five different network topologies.

In fibre-optic networks we found that even in the single hub case, the propagation delay substantially increases the maximum per-packet overhead when the fibre-optic links are long. For maximum link distances, the cascading level N is however limited to: $1 \leq N \leq 2$. For both topologies, a low worst-case data throughput can be expected. The results for the interrupt time remain below those received for UTP based cascaded networks. We observed a maximum of about 203 µs for the Level-2 fibre-optic cascaded network using links of 2 km length.

It remains to remark that we are not aware of any similar analysis performed for 802.12 networks and published anywhere in the literature. The numerical results received in this chapter are not only essential for resource allocation schemes, but will also be useful to accurately describe the 802.12 network behaviour e.g. within simulations. Finally, the analysis of STP based networks was omitted due to the many similarities of this sublayer with the UTP and the fibre-optic physical layer. Given the considerations in this chapter, it should be straightforward to determine the corresponding results for the STP case.

Chapter 6
Deterministic Service Guarantees in 802.12 Networks

Deterministic service guarantees require a worst-case upper bound for all data packets conforming to the user's traffic specification. In this chapter, we prove that such service guarantees can be provided for the end-to-end delay across cascaded and half-duplex switched Demand Priority networks. This is sufficient for supporting the Guaranteed service described in Section 2.2.2. We first concentrated on deterministic guarantees because we believed this to be more challenging. Besides, 802.12 only supports two priority levels. This restricts the number of advanced services that can simultaneously be implemented to just one, assuming that the normal priority medium access is used for best-effort traffic. Implementing the Guaranteed service has the advantage that this provides a service with a high service commitment which could, at the expense of a lower resource utilization, also be employed to serve requests for services with a lower assurance level such as the Controlled Load service, whereas the opposite case does not hold.

We begin with the overall design and the packet scheduling process that is used to enforce the service guarantees. The corresponding admission control conditions providing the required delay bound are defined in Section 6.2. In this section, we also discuss the buffer space requirements and show how resources can be partitioned such that the normal priority service does not starve. Section 6.3 describes the Time Window algorithm which is used to estimate the packet sizes an application is using if these are neither fixed nor negotiatable. Section 6.4 reports implementation issues. We outline the mechanism used for resource management and report some of the problems we encountered during the implementation of the new service. The performance of our resource allocation scheme is evaluated in Section 6.5. This starts with a comparison between analytical and measurement results obtained for the data throughput and the end-to-end delay. We then present results for the Time Window algorithm and discuss resource utilization issues. Also investigated is the impact of system parameters on the resource allocation limit. In Section 6.6, we then look at related work in this area before we summarize the results of this chapter in Section 6.7.

6.1 Packet Scheduling

6.1.1 Design Decisions and Constraints

To built an efficient Guaranteed service in Demand Priority networks, two fundamental problems have to be solved: (1) the Demand Priority overhead has to be considered when computing the

available network resources, and (2) we need a mechanism to find the packet sizes which applications are using. Without the former, the admission control either provides a low resource utilization or non-deterministic service guarantees. The second condition enables us to compute the Demand Priority overhead based on the results for the per-packet overhead and the interrupt time obtained in the previous chapter. We implemented the guaranteed service on top of the 802.12 high priority access mechanism. No changes to the existing LAN standard were required, which ensures backward compatibility and an easy deployment. It however also established the round-robin service discipline as the fundamental packet service order to be considered in the admission control.

The resource reservation itself is based on a time frame concept. It was chosen because this allows us to derive a delay bound, provided all high priority traffic passed into the shared network can be controlled. This further requires that the packet sizes used for the data transmissions are known. In existing operating systems the link layer however cannot negotiate the packet sizes with the application or the upper layer such as e.g. IP. One could be extremely pessimistic and assume the use of minimum sized data packet for all flows. This however reduces the allocatable bandwidth in a single hub network to about 35 Mbit/s, and further decreases in higher cascaded topologies, as could be observed in Figure 4.9 in Section 4.3.1. We thus considered this as an unacceptable solution.

Instead, we used the Time Window algorithm described in detail later in Section 6.3, to find an approximation of the packet sizes. The algorithm can only be applied for applications which do not change their packetization process over time. This was the case for the multimedia applications which we tested. Instead of measuring the packet size directly, the algorithm measures the maximum *number of packets* each flow sends in a time frame. This enables us to compute the total packet overhead, but also allows a flow to use a variety of different packet sizes, including minimum sized packets, as long as the number of packet overheads used within the time frame stays below a certain upper bound.

To restrict the amount of data and the number of data packets passed into the network, we use rate regulators within hosts, routers and LAN switches. The packet scheduling process in switches is thus identical to Rate Controlled Static Priority (RCSP) [ZhFe93] queuing when this scheme is used with just a single priority level. Our admission control conditions however differ significantly from the conditions in [ZhFe93] due to the constraints of the Demand Priority medium access mechanism. In [ZhFe94] and [Zhan95], it is shown that RCSP belongs to a class of service disciplines called *Rate-Controlled Service Disciplines*. There are two basic properties of Rate-Controlled service disciplines which are important in our case. Both are intuitive, but were also formally proved in [Zhan95]:

1. In a network with rate-controlled servers, a deterministic end-to-end packet delay bound can be guaranteed if a deterministic delay bound can be derived: (1) at each server along the data path, and (2) across all network segments connecting these servers. In this case, the end-to-end delay bound is the sum of these bounds.

2. The buffer space requirements for a flow remain constant at all rate-controlled servers along the data path in the network provided all rate regulators use the same traffic shape parameters for this flow.

The packet delay introduced on each segment may be variable as found in a shared LAN environment. Since condition 1 holds for any scheduling scheme providing a deterministic delay bound, it can also be applied to LAN switches with for example different physical medium specific packet schedulers. Both properties, however, rely on rate regulators reshaping each flow's data traffic at each switching node along the data path. Real-time traffic can thus not become burstier as it traverses through the switched network. This allows the derivation of end-to-end performance bounds in arbitrary network topologies.

Using rate controlled LAN switches in the network allows us to extend performance results obtained for a single segment to a bridged network consisting of many segments. In the following, we thus first focus on the packet scheduling process in a single segment and derive a deterministic delay bound for this case. We then look at the end-to-end delay characteristics in bridged networks. Before we begin however, we introduce the model that is used in this thesis to characterize data traffic.

6.1.2 Traffic Characterisation

To allocate resources for an application, the traffic passed into the network by this application needs to be characterized. For this, we use the *Token Bucket* filter since it is simple and used in the Guaranteed and Controlled Load service specifications. In the literature, the token bucket filter is sometimes also called *Leaky Bucket* or (δ, r) *Regulator*. The scheme is analysed for example in [Cruz91a]. The token bucket filter has two parameters: (1) a token generation rate r and a bucket depth δ (the burst size). Tokens are generated at rate r and stored in the token bucket. The bucket depth δ limits the maximum number of tokens that can be stored. Sending a data packet consumes p tokens from the bucket, where p denotes the packet length in bytes. If the bucket is empty or does not contain enough tokens ($p > \delta$) then the packet is stored in a queue until sufficient tokens are available. The maximum size of the queue is bounded and depends on the allocation strategy. Relevant issues for this are discussed in the following section.

The token bucket filter enforces the amount of data which can leave the system in any time interval Δt. A data source i conforms to the (δ^i, r^i) characterisation if in any existing time interval Δt no more than $b^i(\Delta t)$ bytes leave the token bucket, where

$$b^i(\Delta t) \leq \delta^i + r^i \Delta t \tag{6.1}$$

is the *Traffic Constraint Function* [Cruz91a] of source i.

6.1.3 Packet Scheduling Process

In Demand Priority networks, all nodes maintain two link level output queues: one for normal- and one for high priority traffic. In our system we added rate regulators to control the access to the high priority queue on a per-flow basis on each network node. Note that this includes hosts, routers, gateways and LAN switches but not hubs. Each rate regulator is an implementation of the token bucket filter discussed in the previous section. The number of flows using the high priority access mechanism is restricted by admission control. Rate regulation and the Demand Priority protocol thus define the order in which high priority data packets from different nodes are transmitted in the network. Ill behaved *nodes* can be prevented from using the high priority access by network management control of the hub. This is however outside the scope of this thesis.

The link level rate regulators have several functions in our system. We use them: (1) to protect the Guaranteed service from ill behaved applications by controlling the amount of data passed into each high priority output queue in the shared network, and (2) to limit the *number* of data packets which can leave the regulator within a time frame (packet regulator). If resources are not allocated at peak rate then: (3) our rate regulators also smooth out traffic bursts before they can enter the network. Functions (1) and (3) describe traditional functions of a rate regulator. Feature (2) was added in our design.

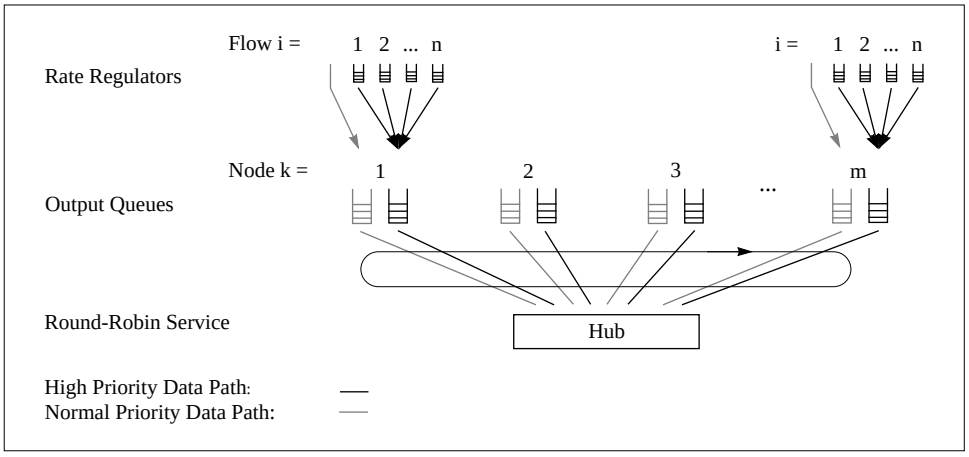


Figure 6.1: The Packet Scheduling Process in a Single Network Segment.

The packet scheduling process is shown in Figure 6.1 for a single hub network. The same process applies to multi-hub networks and half-duplex links. Data packets received from the overlying network layer are first classified. Those using the Best Effort service are immediately passed to the normal priority output queue without being rate regulated. We will not consider them any further in our analysis since their service is isolated and pre-emptable. Each data packet using the Guaranteed service is either: (1) instantly passed on into the high priority output queue when a sufficient number of tokens is available, (2) is stored in the flow's rate regulator-queue until it becomes eligible to send, or (3) is dropped if the regulator-queue has reached its maximum storage capacity.

The time frame concept underlying our resource allocation scheme requires that the total amount of data entering the high priority output queue on each node within a time frame is controlled. This is achieved using the rate regulators which sit immediately above the high priority queue. In hosts, the parameters of each rate regulator could be set so that they either correspond to the *peak* rate of a flow entering the regulator, or to the *average* rate.

If they are set at the peak rate, the regulator does not introduce any delay as long as the flow conforms to its traffic characterisation. In this case, there is always a sufficient number of tokens available to pass the packet into the high priority queue. No buffer space needs to be reserved for the rate regulator-queue. If they are set at the average rate - or more typically to a value between the peak and the average rate - then the regulator in the hosts smoothes out traffic peaks. This reduces the bandwidth to be allocated on the network and thus increases the resource utilization. However, delay is additionally introduced by holding packets in the regulator-queue.

If for example source *i* on host *k* generates data traffic according to the $(\delta^i_{src}, r^i_{src})$ characterisation and the link layer on *k* controls the traffic output using a rate regulator with the parameters (δ^i, r^i), where $r^i_{src} \leq r^i$ and $\delta^i_{src} \geq \delta^i$, then the maximum delay dR^i introduced by the rate regulator is upper bounded by: $dR^i \leq (\delta^i_{src} - \delta^i)/r^i$. Resources corresponding to: (δ^i, r^i) need to be reserved for *i* in the network. Furthermore, a buffer space of δ^i_{src} bytes is required for the rate regulator queue to avoid packet loss. Both follow from the considerations for the token bucket filter in [Cruz91a]. Smoothing data traffic at hosts is not a problem because host memory is typically not a scarce resource. It might however be hard to find the optimum rate regulator parameters (δ^i, r^i) such that the delay requirements and a high network resource utilization are met. In contrast, the rate regulators in LAN switches are only used to smooth out traffic distortions due to load fluctuations in the network. Their parameters correspond to the resources allocated for the flow. This would be (δ^i, r^i) in our example.

In the following, we describe the interaction of network nodes with the Demand Priority medium access protocol and how this leads to the admission control conditions. We first define a time frame of length *TF*. Flows which use the Guaranteed service are denoted as *real-time* flows. For each real-time flow *i* on node *k*, we further define the packet count $pcnt^i_k$ as the maximum number of packets this flow is allowed to pass into the high priority queue within any interval of length *TF*. If we now assume that node *k* has *n* real-time flows, and sufficient resources are allocated such that the packet backlog in the output queue on *k* is always cleared faster than *TF*, then the maximum number of packets in the high priority queue of node *k* is bounded by:

$$PCNT_k = \sum_{i=1}^{n} pcnt^i_k \tag{6.2}$$

The simple round-robin service policy of the hub ensures that the $PCNT_k$ packets in the high priority queue at node *k* will be transmitted within the next $PCNT_k$ high priority round-robin cycles.

Since the maximum number of all packets that become eligible within the time frame *TF* on all other nodes on the segment is known, the scheme can provide a deterministic delay bound for network node *k*.

All bounds are inversely proportional to the data rate passed into the network and thus to the bandwidth allocated for it: nodes with small reservations may receive a smaller delay bound than nodes with large reservations. Assume for example node *k* generating just one data packet per time frame *TF*. In a network with *m* nodes sending with high priority, the queuing delay for *k* is bounded by $m \cdot (P_{max}/C_l)$ plus some Demand Priority overhead, where P_{max}/C_l is the time it takes to transmit one data packet of maximum size. This results from the fact that node *k* is guaranteed to be served once within one round-robin cycle. In contrast, to serve several data packets per time frame as generated by high-bitrate data sources requires several round-robin cycles - which leads to a higher delay bound. The time frame *TF* is the upper bound for all individual node delay bounds. Since the 802.12 standard only supports a single high priority level, the network can only provide a single queueing delay bound per node *k*. This bound applies to all real-time flows on *k*. The end-to-end delay of different flows might however vary dependent on the additional delay that is introduced in the flow's rate regulator at the source node.

The computation of the packet count $pcnt^i$ for flow *i* is straightforward when *i* uses data packets of fixed size. In this case we have:

$$pcnt^i = b^i(TF) / p^i \tag{6.3}$$

where $b^i(TF)$ is the maximum number of bytes which can leave flow *i*'s rate regulator within *TF*, and p^i the packet size used. Equation 6.3 also provides a valid bound for a flow which uses variable sized packets, when p^i is set to the minimum packet size *used* by the flow - or when set to the minimum packet size supported on the link. The latter is 64 bytes in 802.12 networks and always provides a valid bound for the packet count.

In order to provide deterministic service guarantees, all rate regulators must enforce the amount of data which enters the high priority queue in any time interval Δt. In a real implementation, we have to consider the fact that the clocks available to a regulator are granular. With a timer granularity *T*, where $0 < T \leq \Delta t$, all packets which become eligible within the next time tick of length *T* are instantly granted by the regulator. This increases the burstiness of the traffic output. The traffic constraint function initially defined in Equation 6.1 then becomes:

$$b^i(\Delta t) \leq \delta^i + r^i \Delta t + r^i T \tag{6.4}$$

This is used in our implementation. Note first, that $b^i(\Delta t)$ describes the traffic output of the rate regulator for flow *i* and thus the resources to be allocated on the network - and not the traffic that goes

into the regulator. Note further, that we could have retained the traffic constraint function $b^i(\Delta t) \leq \delta^i + r^i \Delta t$ and only transmitted packets after they became eligible. This however introduces a delay of T because of the timer granularity.

6.2 Admission Control

In our resource allocation scheme, the bandwidth, the packet delay and the buffer space conditions in the network need to be checked during the admission control. The core of the admission is the Bandwidth Test defined in Theorem 6.1. It proves that a segment has sufficient spare bandwidth to support the new reservation request. The Delay Bound Test is defined in Theorem 6.2. It takes advantage of the round-robin service policy, which allows us to calculate a delay bound for each individual network node that can potentially be lower than the overall time frame. This increases the flexibility of the allocation system and makes mechanisms for negotiating the time frame to support lower delay bounds less stringent. The bound for the end-to-end delay and the buffer space requirements then follow from Theorem 6.1 and Theorem 6.2. Note that in the admission control, we use the traffic constraint function $b^i(\Delta t)$ for fixed time intervals $\Delta t = TF$. The time frames of different network nodes are further not synchronized.

6.2.1 Bandwidth Test

Theorem 6.1 *Consider an 802.12 network segment with m nodes, where each node k has n real-time flows, which are already admitted. Assume a time frame of TF, a link speed of C_l and that the packet count for flow i on node k is $pcnt_k^i$. Further let P_{min} be the minimum network packet size and D_{pp}, D_{it} be the topology specific worst-case per-packet overhead and normal priority service interrupt time, respectively. Assume further, that the traffic passed into the segment by each real-time flow i on each k obeys the corresponding Traffic Constraint Function $b_k^i(\Delta t)$ for all time intervals $\Delta t = TF$, where $b^i(\Delta t) \leq \delta^i + r^i \Delta t + r^i T$. Sufficient bandwidth for the new flow ν with $b^\nu(TF)$, is available if:*

$$b^\nu(TF) \leq \frac{TF - D_{it} - \frac{1}{C_l} \sum_{k=1}^{m} \sum_{i=1}^{n} b_k^i(TF) - \sum_{k=1}^{m} \sum_{i=1}^{n} pcnt_k^i \cdot D_{pp}}{\frac{1}{C_l} + \frac{D_{pp}}{P_{min}}} \tag{6.5}$$

Before we provide the proof, we briefly discuss this result. Theorem 6.1 tests that the data generated by all real-time flows within the time frame *TF* can also be transmitted within *TF*. The time frame itself is thus always also a deterministic upper bound for the queuing and the propagation delay on the segment. The rather complicated structure of Equation 6.5 is caused by considering the Demand Priority per-packet overhead. The importance of Theorem 6.1 is its capability to accurately provide the available network bandwidth for all valid packet sizes. This is shown later in the performance evaluation in Section 6.5.

Each new flow is admitted based on the worst-case assumption that it initially only uses minimum sized packets for the data transmission. For each already admitted flow *i* however, the corresponding packet count $pcnt^i$ is used in the admission control. If the results for the packet counts are estimated based on measurements in the network, as carried out by the Time Window algorithm, then this admission strategy is similar to the one used in [JDSZ95] for admitting Predictive Service flows based on measurement results of previously admitted flows.

The packet count $pcnt^i$ in Equation 6.5 represents the maximum number of packet overheads which flow *i* may consume within a time frame. Since this overhead is independent of the size of the data packet, flow *i* may for example use its credit to either send $pcnt^i$ minimum- or maximum sized packets. The sum of the packet counts of all flows is the maximum number of packets that are sent on the segment within the time interval *TF*. It corresponds to a *Minimum Average* Packet Size $P_{MIN\_AVE\_S}$ over the time frame *TF*. The relation is given by:

$$P_{MIN\_AVE\_S} = \frac{\sum_{k=1}^{m}\sum_{i=1}^{n} b_k^i(TF)}{\sum_{k=1}^{m}\sum_{i=1}^{n} pcnt_k^i} \tag{6.6}$$

Proof of Theorem 6.1

Theorem 6.1 is implicitly based on a Simple Sum approach which was previously used for example in [JDSZ95] and [JSD97]. Our approach differs from this by additionally considering the Demand Priority protocol overhead. To prove Theorem 6.1, we first define the time frame *TF* as the *Busy Period* interval. This is similar to the definition used in [Cruz91a]. The Busy Period is an upper bound on the time in which high priority data is sent on the network at link speed C_l. The idea is that during the Busy Period, the amount of traffic that enters the system is equal to the amount of data that is served. This is ensured by allocating resources for all data which can leave the link-level rate regulators at all nodes in the network within the time interval *TF*.

In our case, the Busy Period may also include a time offset required at the start of the interval to preempt the normal priority service. The maximum for this offset is the worst-case normal priority service interrupt time D_{it}. It follows that, if the amount of data that is passed in the high priority output queue on each node *k* is bounded by the traffic constraint function $b_k^i(\Delta t)$ for all flows *i* on node *k* and all time intervals $\Delta t = TF$, then *TF* is the Busy Period of the system if:

$$D_{it} + \frac{1}{C_l} \cdot \sum_{k=1}^{m}\sum_{i=1}^{n} b_k^i(TF) \leq TF \tag{6.7}$$

applies, where *m, n* denote the number of network nodes and the number of flows with reservations on each node, respectively. If used for admission control in an overhead free network, Equation 6.7 would ensure that any backlog of high priority packets in any of the high priority output queues in the network is cleared in a time interval smaller or equal to *TF*.

In order to additionally bind the Demand Priority per-packet overhead, we consider the number of packets sent by each flow *i* in every time frame *TF*. This number is denoted $pcnt^i$. In can be the exact number of packets sent by flow *i*, or an upper bound if packet sizes are neither fixed nor negotiatable. Since (1) $pcnt^i$ exists for all real-time flows, and (2) the per-packet overhead is independent of the length of a data packet, the total transmission overhead within the time frame *TF* can be computed. Both is exploited for Theorem 6.1. If we assume that the worst case per-packet overhead is D_{pp} and that $pcnt_k^i$ denotes the maximum number of packets sent by flow *i* on node *k*, then by adding D_{pp} for each data packet served, we get from Equation 6.7:

$$D_{it} + \frac{1}{C_l} \cdot \sum_{k=1}^{m} \sum_{i=1}^{n} b_k^i(TF) + \sum_{k=1}^{m} \sum_{i=1}^{n} pcnt_k^i \cdot D_{pp} \quad \leq \quad TF \tag{6.8}$$

This introduces the non-linear characteristic which we could observe in the measurement results in Figure 4.9. It follows that, a new flow ν with a traffic constraint function $b^\nu(\Delta t)$ can be accepted if for all time intervals $\Delta t = TF$ condition:

$$D_{it} + \frac{1}{C_l} \cdot \sum_{k=1}^{m} \sum_{i=1}^{n} b_k^i(TF) + \sum_{k=1}^{m} \sum_{i=1}^{n} pcnt_k^i \cdot D_{pp} + \frac{b^\nu(TF)}{C_l} + \frac{b^\nu(TF)}{P_{min}} \cdot D_{pp} \quad \leq \quad TF \tag{6.9}$$

holds. If the packet size used by flow ν is fixed and larger than the maximum link packet size P_{min} then we can replace this parameter in Equation 6.9 with the actual packet size p^ν used. If ν uses variable packet sizes and the actual number of data packets transmitted is known or can be negotiated then the term: $b^\nu(TF)/P_{min}$ in Equation 6.9 can be replaced by flow ν's packet count: $pcnt^\nu$. Theorem 6.1 follows directly from re-arranging Equation 6.9 □

6.2.2 Delay Bound Test

After testing that the network has sufficient spare bandwidth to admit the new flow, Theorem 6.2 can be used to derive a tighter delay bound than given by the time frame *TF*. The test can be omitted when the delay bound requested for the segment is larger than the current time frame *TF*. Since the admission of a new flow can change the delay bounds for all nodes with reservations on the local segment, the verification must be carried out for all of them.

Theorem 6.2 *Consider an 802.12 network segment with m nodes, where each node k has n real-time flows, which are already admitted. Assume a link speed of C_l and that the packet count for flow i on node k is $pcnt_k^i$. Further let P_{max} be the maximum link packet size and D_{pp}, D_{it} be the topology specific worst-case per-packet overhead and normal priority service interrupt time, respectively. If Theorem 6.1 applies, and if the traffic passed into the network segment by each real-time flow i on each node k obeys the corresponding Traffic Constraint Function $b_k^i(\Delta t)$ for all intervals $\Delta t = TF$, where $b^i(\Delta t) \leq \delta^i + r^i \Delta t + r^i T$, then the sum of the queuing delay and the propagation delay on the segment, denoted with dS_k for node k, is bounded by:*

$$\sum_{j=1,\, j\neq k}^{m}\left(MIN\left(\sum_{i=1}^{n} pcnt_k^i,\ \sum_{i=1}^{n}\frac{b_j^i(TF)}{P_{max}}\right)\cdot\frac{P_{max}}{C_l}+MIN\left(\sum_{i=1}^{n} pcnt_k^i,\ \sum_{i=1}^{n} pcnt_j^i\right)\cdot D_{pp}\right)\ +$$

$$\frac{1}{C_l}\sum_{i=1}^{n} b_k^i(TF)+\sum_{i=1}^{n} pcnt_k^i\cdot D_{pp}+D_{it}\ \ \leq\ \ dS_k\ \ \leq\ \ TF \qquad (6.10)$$

Proof of Theorem 6.2

Network node *k* may pass a maximum of $PCNT_k$ data packets into its high priority output queue within each time frame *TF*, where $PCNT_k$ is given by Equation 6.2. This is controlled for all flows *i* on each node *k* by the packet regulating mechanism of the rate regulators used in our system. If Theorem 6.1 applies then the delay for all high priority data packets within the network segment is bounded by *TF*. Otherwise the condition $dS_k \leq TF$ is not true for all nodes *k* on the segment. In the worst case, the output queue length and thus the queuing delay on nodes with $dS_k > TF$ could grow unboundedly since data packets can be generated faster on these nodes than the network can serve them. Theorem 6.2 thus requires that Theorem 6.1 applies.

Assuming that all $PCNT_k$ data packets are passed into *k*'s high priority output queue in a single packet burst, then the sum of the worst-case queuing delay and the propagation delay for the last packet of the burst consists of: (1) the Interrupt Time: required to signal the high priority service request and to pre-empt the normal priority network service, (2) the Local Packet Transmission Delay: defining the time it takes to transmit all locally queued data packets through the network stack and over the physical medium, and (3) the External Packet Transmission Delay: caused by the fact that data packets on node *k* might have to wait until high priority requests on other nodes have been served according to the round-robin service policy carried out by the network. We thus have for dS_k:

$$D_{it} + dL_k + dE_k\ \leq\ dS_k\ \leq\ TF \qquad (6.11)$$

where D_{it}, dL_k, dE_k denote the Interrupt Time, the Local- and the External Packet Transmission delay, respectively. We now provide bounds for all three components. The worst-case normal priority service interrupt time D_{it} was analysed in Chapter 5. The Local Transmission Delay dL_k required to transmit the maximum of $PCNT_k$ data packets queued at node *k* is bounded by:

$$dL_k\ \leq\ \frac{1}{C_l}\cdot\sum_{i=1}^{n} b_k^i(TF)+\sum_{i=1}^{n} pcnt_k^{\,i}\cdot D_{pp} \qquad (6.12)$$

This follows from the considerations made in the previous section and Equation 6.2. Note that dL_k also considers the propagation delay due to the parameter D_{PHY} included in the per-packet overhead

D_{pp}. The External Packet Transmission Delay dE_k for node *k* depends on the number of high priority packets queued on all other network nodes $j \neq k$ within the time interval *TF*. This number is bounded by $PCNT_j$ for each node *j*.

The service of packets from node *k* is most delayed by node *j*, when node *j* has at least as many packets in its output queue as node *k*. In general, two cases can be identified: if we first assume that node *j* has more than $PCNT_k$ maximum size packets in its output queue, then the network serves the same number of packets from node *j* and node *k* until all packets on *k* have been transmitted. Some data packets are still in the queue on *j*, but they do not have to be considered for the delay computation on *k*. We thus have the relation:

$$dE_{k,j} \leq PCNT_k \cdot \frac{P_{max}}{C_l} \qquad \text{if} \qquad PCNT_k \leq \sum_{i=1}^{n} \frac{b_j^i(TF)}{P_{max}} \tag{6.13}$$

for the External Transmission Delay $dE_{k,j}$ imposed by node *j* on node *k*. If however node *j* has less data packets to send than node *k*, then all packets on *j* are served during the time it takes to transmit $PCNT_k$ data packets from node *k*. This is enforced by the round-robin service policy. For this case, we receive the relation:

$$dE_{k,j} \leq \sum_{i=1}^{n} \frac{b_j^i(TF)}{P_{max}} \cdot \frac{P_{max}}{C_l} \qquad \text{if} \qquad PCNT_k > \sum_{i=1}^{n} \frac{b_j^i(TF)}{P_{max}} \tag{6.14}$$

for the External Transmission Delay $dE_{k,j}$. If we now consider the transmission delays caused by all network nodes *j* with $j \neq k$, we have from Equation 6.13 and Equation 6.14:

$$dE_k \leq \sum_{j=1,\, j\neq k}^{m} \left(MIN\left(PCNT_k, \sum_{i=1}^{n} \frac{b_j^i(TF)}{P_{max}} \right) \cdot \frac{P_{max}}{C_l} \right) \tag{6.15}$$

where *m* is the number of nodes with real-time flows on the network. Equation 6.15 provides an upper bound on the service time required to serve the maximum number of data packets from all nodes *j* in the network, while $PCNT_k$ packets are served from node *k*.

The last overhead to be considered in Theorem 6.2 is the Demand Priority per-packet overhead D_{pp}. For all data packets transmitted within *TF* from node *j*, the delay introduced by the per-packet overhead is upper bounded by: $PCNT_j \cdot D_{pp}$. It follows from the considerations made for Equation 6.13 and Equation 6.14 that only the minimum of $PCNT_k$ and $PCNT_j$ needs to be considered for the delay imposed by node *j* on node *k*. This is because the high priority output queue on: (1) node *k*, or (2) node *j*, or (3) on both nodes *k* and *j* will be empty after: $MIN(PCNT_k, PCNT_j)$ round-robin cycles. We thus receive a delay of: $MIN(PCNT_k, PCNT_j) \cdot D_{pp}$ to be considered for node *j*. By adding this result for all nodes $j \neq k$ to Equation 6.15, we have:

$$dE_k \leq \sum_{j=1,\, j\neq k}^{m} \left(MIN\left(PCNT_k, \sum_{i=1}^{n} \frac{b_j^i(TF)}{P_{max}} \right) \cdot \frac{P_{max}}{C_l} + MIN(PCNT_k, PCNT_j) \cdot D_{pp} \right) \tag{6.16}$$

If we now substitute Equation 6.2 in Equation 6.16 and insert 6.12 and 6.16 in Equation 6.11, then we receive for the delay bound of node *k*:

$$\sum_{j=1,\, j\neq k}^{m} \left(MIN\left(\sum_{i=1}^{n} pcnt_k^i, \sum_{i=1}^{n} \frac{b_j^i(TF)}{P_{max}} \right) \cdot \frac{P_{max}}{C_l} + MIN\left(\sum_{i=1}^{n} pcnt_k^i, \sum_{i=1}^{n} pcnt_j^i \right) \cdot D_{pp} \right) +$$

$$\frac{1}{C_l} \sum_{i=1}^{n} b_k^i(TF) + \sum_{i=1}^{n} pcnt_k^i \cdot D_{pp} + D_{it} \leq dS_k \leq TF \tag{6.17}$$

This is Theorem 6.2. □

6.2.3 End-to-End Delay Characteristics

Figure 6.2 illustrates the packet forwarding in a bridged network consisting of switches with a rate-controlled server. All data packets are depicted as arrows and belong to the same flow. They are sent by the Data Source and traverse Switch 1, Switch 2 and Switch 3 on their way to the destination (not shown). The x-axis in Figure 6.2 represents the time consumed in the network. The upper part of the y-axis shows the data path from the Data Source to Switch 3, the lower part illustrates the packet delay encountered by the first, third, fifth and seventh packet.

To determine the end-to-end delay, we assume that: (1) the traffic passed into the output queue at the data source is rate regulated and conforms to the (δ, r) characterisation, (2) the flow is reshaped upon arrival at each switch in the bridged network such that the traffic pattern sent into the switch's output queue also conforms to (δ, r), and (3) the token bucket depth δ (the burst size) is at least as big as the maximum packet size p ($\delta \geq p$) used by the data source. In the example in Figure 6.2, we restricted the third condition by assuming that all data packets are of the same size p and by setting: $\delta = p$. Finally, all network nodes k within the data path of a flow are assumed to be continuously numbered such that the flow's data source has $k = 1$, the first switch $k = 2$, and so on towards the receiver which then bears number $k = m$.

The data flow in Figure 6.2, starts on Segment 1. Due to cross traffic on the segment (not shown), the transmission of all data packets from the Data Source is delayed by dS_1 time units. The result is a packet burst arriving afterwards at Switch 1. Since $\delta = p$, the first data packet is instantly passed on into the output queue without being delayed by the rate regulator. All following packets are however held in order to reconstruct the original traffic pattern. This ensures that the data traffic passed into the output queue at Switch 1 has the same interpacket time difference as the flow which left the rate regulator at the Data Source. The second and third segments in the data path delay the traffic by dS_2 and dS_3 time units. The resulting packet bursts are afterwards smoothed by the flow's rate regulator in Switch 2 and Switch 3, respectively.

The end-to-end delay encountered by data packets in the network may include several or all of the following components: (1) a holding time in the rate regulator at the data source, (2) a queuing and a propagation delay on each segment, (3) a holding time in the rate regulator within all switches in the data path, and (4) an overhead delay introduced at the data source, in each switch and in the receiver. The last component, the overhead delay, denotes the time consumed by the packet processing within the data source, the switches along the data path and within the receiver.

By adding up the worst-case delays of all applicable components between the Data Source ($k = 1$) and Switch 3 ($k = 4$), we obtain: $dR_1 + dO_1 + dS_1 + dO_2 + dS_2 + dO_3 + dS_3 + dO_4$ as an upper bound for the *first* data packet in Figure 6.2. The parameter dR_1 denotes the holding time in the source's rate regulator. dO_1, dO_2, dO_3 and dO_4 are the overhead delays for the data source and for the three switches in the data path, respectively. The above result is straightforward to see because the first packet is never delayed by a rate regulator in any of the switches.

An important property of networks consisting of rate controlled servers is that holding data packets in rate regulators within switches will *not* increase the end-to-end delay bound of the flow, provided that the rate regulators in all switches reshape the flow's traffic based on the same traffic characterisation.

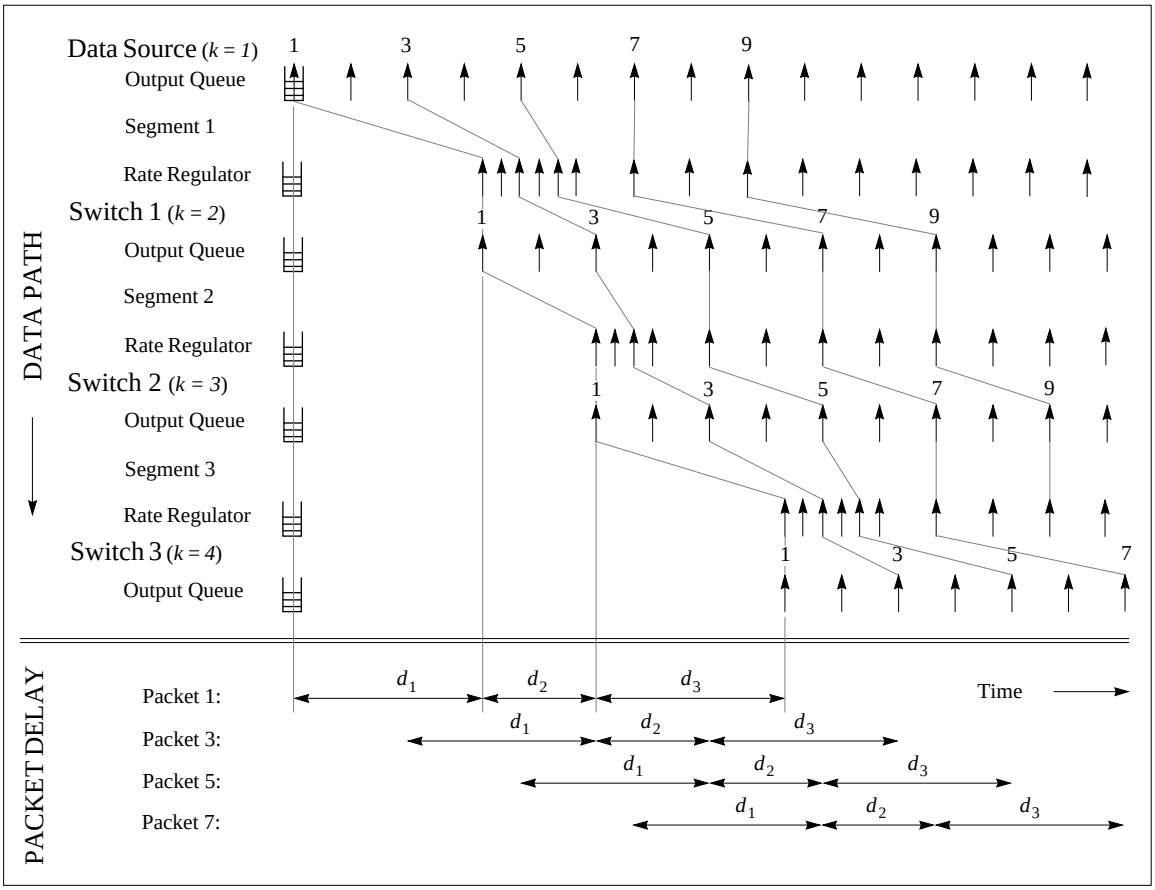


Figure 6.2: Packet Forwarding in a Network with Rate Controlled Servers.

This behaviour can also be observed in Figure 6.2 for the data packets following the first packet. Even though these packets are delayed in the rate regulators, they nevertheless encounter the same end-to-end delay as the first packet. The formal proof of this property is given in [Zhan93] or [ZhFe94] for the $(X_{min}, X_{ave}, I, S_{max})$ traffic model, and in [GGPS96 - Section III] for the token bucket traffic characterisation. It is thus omitted here.

By considering this property in the end-to-end delay bound $dN^i_{End\text{-}To\text{-}End}$ for flow *i* whose data packets traverse *m -1* network segments on their way from node *1* to node *m*, we obtain:

$$dN^i_{End\text{-}To\text{-}End} = dR^i_1 + \sum_{k=1}^{m} dO^i_k + \sum_{k=1}^{m-1} dS^i_k \tag{6.18}$$

Intuitively, a data packet is only delayed in a rate regulator when its interpacket time difference to the previous data packet is smaller than the reference value given by the flow's traffic characterization (δ, r). This however does not increases the end-to-end delay bound. Note that parameter dS^i_k in Equation 6.18 denotes the sum of the queuing delay on node *k* and the propagation delay on the segment connecting *k* with node *k +1*.

Most of the end-to-end delay is typically caused by queuing data packets within the network. The rate regulator at the source node may however introduce a significant delay when the flow is bursty and bandwidth is allocated close to the average data rate. Otherwise, when allocating at peak bandwidth, we get $dR^i_1 = 0$. The overhead delays dO^i_k in our test network are in the order of a few hundred microseconds. For each of our 802.12 LAN switches for example, we have: $dO \approx 132\ \mu s = (120\ \mu s + 12\ \mu s)$ assuming IEEE 802.3 frame formats. Both delay components in dO were discussed in Section 5.2.6. It remains to remark that, when neglecting the overhead delays, Equation 6.18 is basically identical to Equation 13 in [GGPS96 - Section III], provided the rate regulators within all switches along the data path reshape a flow's data traffic according to the same traffic characterisation.

Let us now discuss the relation of Equation 6.18 with the ISPN framework and the Guaranteed service described in Chapter 2. To support the reservation for a flow *i* requesting the Guaranteed service, a bridged Demand Priority LAN exports the following results for the C^i and D^i error terms:

$$\begin{aligned} C^i &= 0 \\ D^i &= dN^i_{End\text{-}To\text{-}End} \end{aligned} \tag{6.19}$$

where $dN^i_{End\text{-}To\text{-}End}$ is the result received from Equation 6.18. Both parameters are used in Equations 2.1 - 2.3 in Section 2.2.2 for computing the end-to-end delay. The mapping between the error terms and Equation 6.19 is however not ideal because of a rate dependent error term of: $C = 0$. This makes it harder for the service requestor to predict the impact of more bandwidth on the actual end-to-end delay. A request for a lower delay implicitly specified in the *RSpec* will nevertheless lead to a

lower delay bound, provided the Demand Priority LAN has sufficient spare resources to support the request. A successful allocation then results in an update of the error terms exported to the service requestor.

When used for a single bridged Demand Priority LAN, Equation 6.18 further provides a tighter upper delay bound than Equation 2.1 or Equation 2.2 using the mapping in Equation 6.19. This is caused by the different strategies underlying these equations. The bound defined by Equation 2.1 - 2.3 implies the ability of exploiting dependencies between all servers within the flow's data path such that the fluid delay: δ / R can be split off the result. This model fits well for networks consisting of WFQ servers[1] but is difficult to follow in networks where the data transmission of different flows is less isolated such as shared or half-duplex switched LANs. In contrast, the bound in Equation 6.18 is achieved by adding up the local bounds obtained on all segments traversed by the flow. In this model, each local queuing delay bound is independent from the result received on the previous segment. Equation 2.1 and 2.2 will however be accurate for reservations across heterogeneous internetworks including for example routers with WFQ servers and Demand Priority subnetworks at the edges of the data path. In this case, the fluid delay δ / R is required as part of the delay bound covering the wide area data path.

Note that summing the worst-case delays at each node within an internetwork does not automatically lead to high end-to-end delay bounds. In [GGPS96] it is shown that any end-to-end delay bound that can be achieved with the WFQ service discipline in an internetwork, can also be guaranteed by a Rate Controlled service discipline which uses a proper reshaping algorithm and packet scheduler. The sum of the local bounds of this scheme is then no larger than the bound received from the WFQ discipline.

6.2.4 Buffer Space Requirements

To prevent packet loss, sufficient buffer space needs to be reserved within the network. This includes buffer capacity to hold data packets in the output queues and in the rate regulators. If admission control is performed and Theorem 6.1 applies, then an upper bound on the buffer space exists for all real-time flows. Furthermore, using rate regulators within switches ensures that these requirements remain constant for all switches along the data path, provided their rate regulators reshape the data traffic based on the same traffic characteristics.

An upper bound required for flow *i* to prevent packet loss at the output queue is given by: $b^i(TF)$. This follows from: (1) the rate regulation of the flow guaranteeing that within any time interval *TF*, never more than $b^i(TF)$ bytes can enter the output queue, and (2) Theorem 6.1 - which ensures that there is always sufficient network bandwidth to transmit $b^i(TF)$ bytes from the output queue within *TF*. The actual buffer space required for flow *i* will however be lower than $b^i(TF)$ because of the round robin service policy and the fact that resources are typically not allocated up to the allocation

1. See for example the result for the end-to-end delay in [PaGa94 - Section X.C].

limit. The latter can be exploited using Theorem 6.2. If the delay bound dS provided by Theorem 6.2 is lower than the time frame, then $b^i(dS)$ is a tighter bound for the buffer space. This follows from the same considerations as made for $b^i(TF)$. The disadvantage is however that $b^i(dS)$ depends on the allocated network resources and will change whenever a new flow is admitted. From this, we have for the buffer space sQ^i required for flow *i* in the output queue of node *k*:

$$sQ_k^i \le b_k^i(dS_k) \le b_k^i(TF) \tag{6.20}$$

We now look at the requirements for flow *i*'s rate regulator at switch *k*. Due to the rate regulation at switch *k -1*, switch *k* can never receive more than $b^i(TF)$ bytes for flow *i* within *TF*. Upon arrival, $\delta^i + r^i T_k$ bytes are instantly passed on into the output queue. This follows from the definition of the traffic constraint function (Equation 6.4) in Section 6.1.3. The term $r^i T_k$ represents the amount of data sent ahead of schedule at switch *k* due to the timer granularity T_k. In the following interval *TF*, data equivalent to $r^i \cdot TF$ may enter the rate regulator at *k*. This again relies on the regulator at switch *k -1*. The same amount of data may however also leave the regulator at *k* since both switches, *k -1* and *k*, control flow *i* based on the same parameter set: (δ^i, r^i). We thus have an upper bound of:

$$sR_k^i \le b_{k-1}^i(TF) - (\delta^i + r^i T_k) = r^i(TF + T_{k-1} - T_k) \tag{6.21}$$

for the buffer space required for flow *i*'s rate regulator at switch *k*. The parameters T_{k-1} and T_k in Equation 6.21 denote the timer granularity at switch *k -1* and switch *k*, respectively[1], where $T_k \le TF$ for all *k*. Typically, we however have: $T_k \ll TF$. If we now add the upper bounds for the output queue and the rate regulator, we receive:

$$sS_k^i \le sQ_k^i + sR_k^i \le b_k^i(TF) + r^i(TF + T_{k-1} - T_k) = \delta^i + 2r^i TF + r^i T_{k-1} \tag{6.22}$$

where sS_k^i denotes the buffer space requirements for flow *i* on switch *k*. The worst case occurs when flow *i*'s data packets experience a maximum delay at switch *k -1* such that the resulting packet burst fills up the regulator queue at *k*. All following data packets are then forwarded with the minimum delay at switch *k -1*, whereas the data in the output queue at *k* are delayed up to the maximum of $dS^i{}_k \le TF$. It remains to add that using the result of Theorem 6.2 in Equation 6.22 may again lead to the lower, but utilization dependent upper bound for the buffer space.

1. We assume here that all rate regulators at switch *k* are served with the same timer granularity T_k.

6.2.5 Resource Partitioning

To ensure that normal priority data traffic does not starve, network resources must be partitioned. The availability of resources for normal priority traffic is guaranteed by restricting the access to the high priority service. This is enforced by admission control.

To control the resource share for high priority traffic, we first define the *High Priority Utilization Factor f*, where $0 \leq f \leq 1$. f defines the maximum resource share that can be allocated for high priority traffic. A utilization factor of $f = 1$ thus allows the allocation of all network resources available. Since our resource allocation scheme is based on a time frame concept, the resource maximum corresponds to the total transmission time that is available within the time frame *TF*. In addition to parameter *TF*, we define the *minimum* normal priority transmission time *LTT*. It represents the minimum resource share that is guaranteed to be available for normal priority traffic. The minimum for LTT is the interrupt time D_{it}. The resources represented by D_{it} can not be allocated since they are required for pre-empting the normal priority service. The maximum for *LTT* is the time frame itself. In this case no resources can be allocated for the high priority service. We thus have the relation: $D_{it} \leq LTT \leq TF$. If we now additionally consider the high priority utilization factor *f*, then we receive for the minimum normal priority transmission time:

$$LTT = MAX(D_{it}\,;\, TF \cdot (1 - f)) \tag{6.23}$$

where $D_{it} \leq LTT \leq TF$ is achieved for utilization factors of: $0 \leq f \leq 1$. If we now replace the interrupt time D_{it} in Theorem 6.1 with the minimum normal priority transmission time *LTT* then we have:

$$b^{\nu}(TF) \leq \frac{TF - LTT - \frac{1}{C_l}\sum_{k=1}^{m}\sum_{i=1}^{n} b_k^i(TF) - \sum_{k=1}^{m}\sum_{i=1}^{n} pcnt_k^i \cdot D_{pp}}{\frac{1}{C_l} + \frac{D_{pp}}{P_{min}}} \tag{6.24}$$

To enable the network administrator to control the high priority allocation limit, Equation 6.23 and Equation 6.24 are used for admission control. The allocation limit is changed by adjusting the utilization factor *f*. An example is given later in Figure 6.9 in Section 6.5.1. Theorem 6.2 does not need to be updated to support resource partitioning since for all utilization factors, the normal priority data transmission is still pre-empted after D_{it} time units. For low utilization factors, the delay bounds given by Theorem 6.2 are always significantly smaller than the time frame *TF*. This is due to the smaller total amount of resources allocated.

The partitioning mechanism described in this section provides a simple method for network administrators to set a basic policy required in Integrated Services networks: the minimum bandwidth available for normal- and high priority traffic. We believe that without any such control, an

advanced service based on a static priority queueing system can not be deployed because of the starvation problem. This section however showed that such control can easily be integrated in our allocation system.

6.3 A Time Window Algorithm for the Packet Count Estimation

The time window measurement algorithm described in this section is used to find a realistic upper bound on the number of data packets generated by a flow within the time frame *TF*. This bound allows us to compute the Demand Priority overhead to be considered for this flow in the admission control. The development of the algorithm was motivated by the fact that in existing systems, the link layer cannot negotiate the packet size with upper layers or the application. Without such an algorithm, either: (1) fixed sized data packet must be used, (2) new mechanisms for negotiating the packet count with upper layers have to be introduced, or (3) the allocation must be performed based on the minimum packet size used by the flow. For flows using variable sized packets, this is often the minimum packet size supported on the network.

6.3.1 The Estimation Process

The algorithm is carried out on a per flow basis at end-systems such as hosts generating data traffic to be passed into the network. The upper bound on the number of data packet sent by flow *i* is denoted with: $pcnt^i$. Two parameters are measured at the link layer. The measurement variable $scnt^i$ tracks the number of packets seen from flow *i* within the current time frame *TF*. This is measured after the flow is rate controlled. In $scnt^i_{TW}$, we keep a record of the maximum value observed for $scnt^i$ within the current measurement time window *TW,* where $TF \ll TW$. The second parameter measured is flow *i*‘s data rate r^i_{TW}, averaged over the time window *TW.*

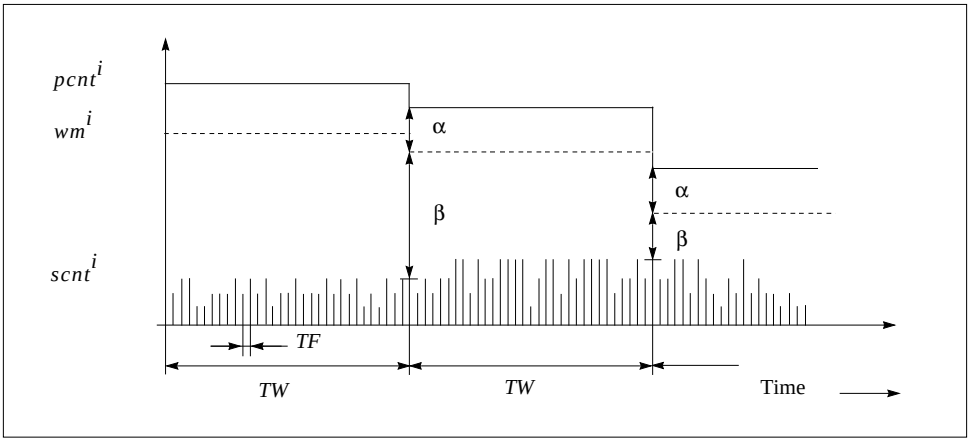


Figure 6.3: The Measurement Process for Flow *i*.

The parameter MAX_PCNT^i denotes the worst-case packet count for the flow. It corresponds to the case when the application only uses minimum sized packets for transmitting its data. MAX_PCNT^i is computed using the minimum link packet size $p^i = P_{min}$ in Equation 6.3. The measurement

process itself is illustrated in Figure 6.3. Realistic, measured sample patterns are shown later in Section 6.5.3. In the following, we describe how the measurements are used to estimate an upper bound $pcnt^i$ for flow *i*.

Initially, $pcnt^i$ is set to MAX_PCNT^i. The value can be changed: (1) at the end of each time window *TW*, and (2) when an individual measurement for $scnt^i$ reaches the high watermark: wm^i. The latter case is not illustrated in Figure 6.3. At the end of each time window, $pcnt^i$ is updated to reflect the measurements taken for the flow in the previous time interval *TW*. The new value which we denote with $pcnt'^i$ is the sum of the maximum observed sample and two parameters: α^i and β^i, which reflect the conservativeness and the level of uncertainty of the sample measured. $pcnt'^i$ can however never exceed MAX_PCNT^i since this is the maximum number of packets which this flow can possibly send in a time frame without violating its allocated data rate. For flow *i*, we thus have:

$$pcnt'^i = MIN((scnt^i_{TW} + \alpha^i + \beta^i),\ MAX\_PCNT^i) \tag{6.25}$$

The parameter α^i, where $0 \le \alpha^i \le MAX\_PCNT^i$, allows us to be more conservative by increasing $pcnt'^i$ to a value higher than the measured sample. It is set on a per-flow basis. The parameter β^i reflects the level of uncertainty associated with the measured sample. It is proportional to the difference between the allocated and the measured data rate. β^i is small if the rate measured is close to the rate allocated for this flow. If the difference is larger, then β^i also increases. This ensures that the new value $pcnt'^i$ is not decreased when a data source is switched off or the application temporary generates significant less data than allocated. Formally, we get for this parameter:

$$\beta^i = \frac{(r^i_{alloc} - r^i_{TW}) \cdot (TF + T)}{P_{min}} + 1 \tag{6.26}$$

where r^i_{alloc} and r^i_{TW} are the allocated and the measured data rate for flow *i*, respectively. The parameter *T* is the timer granularity of the rate regulator. It can be neglected for the case that: $T \ll TF \ll TW$ holds. Using Equation 6.26 for the computation of β^i is very conservative since it assumes the use of minimum sized packets (P_{min}) for the data rate unused by flow *i*. A less conservative approach might instead use an application specific value larger than this.

As illustrated in Figure 6.3, the packet count $pcnt^i$ for each flow *i* has a corresponding high watermark wm^i. Both differ by the parameter α^i:

$$wm^i = pcnt^i - \alpha^i \tag{6.27}$$

Whenever an individual measurement for $scnt^i$ reaches the high watermark wm^i and the existing bound $pcnt^i$ is smaller than MAX_PCNT^i then the present estimation is wrong and we immedi-

ately update $pcnt'$ to be κ times the existing value. Since the new value $pcnt''$ can again not exceed MAX_PCNT^i, we receive for this case:

$$pcnt'^i = MIN((\kappa \cdot pcnt^i), MAX\_PCNT^i) \quad (6.28)$$

where $pcnt'^i$ and $pcnt^i$ are the new and the old packet count, respectively. The packet count estimation process can be summarized as follows:

1. At the beginning of the estimation for flow *i*, set $pcnt^i$ to MAX_PCNT^i.
2. When the flow has been setup, measure the number of packets seen from *i* within the current time frame *TF*. Store the result in $scnt^i$. In $scnt^i_{TW}$, keep a record of the maximum value observed for $scnt^i$ within the current time window *TW*. Further, measure the data rate r^i_{TW} for the flow and average it over *TW*.
3. At the end of each time window *TW*, use Equation 6.25 and 6.26 to compute the new value $pcnt'^i$. If required, replace the existing packet count $pcnt^i$ with the new value and compute the high watermark wm^i using Equation 6.27.
4. Whenever an individual measurement for $scnt^i$ reaches the high watermark wm^i and $pcnt^i < MAX\_PCNT^i$ then use Equation 6.28 to compute the new packet count $pcnt'^i$. Update the existing $pcnt^i$ and compute the corresponding high watermark wm^i using Equation 6.27.

6.3.2 Admission Control and Service Issues

If the packet count estimation only relies on measured information then any new flow is initially admitted based on the assumption that it will only use minimum sized data packets. Then as the flow starts sending data, the Time Window algorithm measures the maximum number of packets used by the flow per time frame and takes a pessimistic maximum that is higher than the observed value.

The adaptation rate of the algorithm depends on two parameters: (1) the length of the time window *TW* and (2) the difference between the allocated bandwidth and the bandwidth actually used by the application. A smaller time window increases the sensitivity of the algorithm since the packet counts are more frequently updated. It however also reduces the averaging interval used to compute the rate parameter r_{TW}, resulting in a less conservative uncertainty factor β. If an application only uses a small percentage of the resources allocated then the parameter β ensures that the packet count is not decreased. This is important because the application might have stopped the data transmission or just temporarily reduced its data output because, for example, of the specific characteristics of a video encoder. If resources are sparsely used, then the algorithm might not be able to find a close approximation of the packet count within *TW* since it is uncertain whether the samples observed during that interval actually reflect the characteristic of the packetization process.

The conservativeness of the measurement process is controlled by the length of the time window *TW*. It could be as pessimistic as required at the expense of the network resource utilization. The worst case is an infinite time window which assumes that all data is sent using minimum sized data packets as assumed for new flows. This is very pessimistic, especially for realistic flows with a high data rate.

The algorithm relies on the property that the packetization process does not change over time. With the packetization process, we mean the algorithm used to break data, e.g. a video frame, into single data packets. Video frames of variable length might for example be fragmented by breaking each of them into a number of 1024 byte date packets plus one variable sized packet which contains the rest of the frame.

If the packetization process however changes over time and the packet sizes become substantially decreased, then the packet counter $scnt^i$ will hit the high watermark wm^i. This triggers an immediate update of the estimated bound. Note that increasing the packet count $pcnt^i$ implies allocating resources for flow *i* on the network. Whenever the high watermark is reached then the flow may still send α^i packets within the present time frame *TF* before a service violation actually occurs.

We believe that the measurement aspect does not conflict with the requirements of a Guaranteed service, because we only apply the algorithm for applications with a constant packetization process. Whenever a service with less stringent commitments is requested e.g. a Controlled Load service, then the algorithm might also be used for applications which do change their packetization process.

Instead of the Time Window algorithm described in this section, there is probably a multitude of similar algorithms which could be used to estimate the packet count. Due to the variety of application characteristics, it will however be hard to identify the *best* algorithm. We thus deliberately did not attempt this, but focused on feasibility and simplicity. The measurement results in Section 6.5.3 show that for the multimedia applications we tested, our algorithm is able to find an accurate upper bound without impairing the guaranteed service quality. The important advantage of using a measurement based approach is that it can substantially improve the efficiency of the allocation scheme when compared with an allocation based only on minimum sized data packets. The disadvantage is that whenever deterministic guarantees are requested, the algorithm can only be used for applications with a constant packetization process. The approach further has a slow adaptation rate which might cause the rejection of a reservation request even though, in reality, sufficient network resources are available. The optimal solution for this problem would be a mechanism for negotiating the packet count with the upper layers.

It remains to remark that our Time Window algorithm has some similarities with the Time Window algorithm proposed in [JDSZ95]. This algorithm is used as part of a measurement based admission control scheme for Predictive service. It measures: (1) the queueing delay for data packets on a per-flow basis, and (2) the average data rate for aggregations of data flows. Similar to our scheme are the estimation of performance parameters over a time interval, a system parameter to control the conservativeness of the estimation, and that the estimates are updated at the end of the time frame or

when a pre-defined threshold is exceeded. Both algorithms however differ in respect to the parameters to be estimated and in the conditions used for computing the new estimates. Our algorithm is further built on the assumption that there is a constant packetization process whose maximum packet generation rate needs to be discovered. The time window algorithm in [JDSZ95] can not make such an assumption but attempts to estimate parameters which typically continually change.

6.4 Implementation Issues

We implemented and tested our resource allocation scheme in the 802.12 test network described in Section 3.2.1. This section briefly reports some of the design decisions we made and some of the problems we encountered during the implementation.

6.4.1 Signalling and Resource Management

The link level signalling and the resource management within the test network was performed by the LLRMP protocol [Kim96]. It was installed on all workstations and LAN switches using the 802.12 high priority access mechanism. The LLRMP is a link level signalling protocol that is used to carry the traffic characterisation and the reservation request through shared and switched LANs. Resources are reserved on a hop-by-hop basis, where a *hop* denotes a shared segment or a link between two LAN switches. The protocol can support a distributed resource management, installs soft-states in hosts and bridges, and allows users to dynamically change their reservations. The latter property is also used to update the resource information e.g. the packet count, which is held at the resource arbiter. We refer to [Kim96] for the details of the protocol operation.

The host part of the LLRMP is implemented in a user space daemon. It performs the LLRMP control message processing, the admission control and the Time Window measurement algorithm. A user interface allows access to the resource data base. The daemon runs on top of the 802.12 LAN driver using the Link Level Access (LLA) [HP92a] interface. The LLA is a generalized *ioctl* based interface which provides basic low level access to device drivers in the HP-UX kernel. The LLRMP daemon uses this interface for: (1) sending and receiving control messages, (2) to control the rate regulators and the packet classifier in the kernel, and thus the medium access priority for all data packets. Application data uses the normal path through the transport and network protocol stack. We extended the LLA functionality to support asynchronous event notifications and to control the classifier and the rate regulators. Asynchronous events are implemented by using a UNIX signal. The control mechanisms are based on extended *ioctl* calls.

The LLRMP protocol was implemented as a user space daemon for reasons of simplicity. Only functionality in the data path, like the classifier and the rate regulators were kept in the kernel. Separating these mechanisms however also caused a difficulty: context information is basically maintained twice: once in the daemon and once in the kernel. This is because the rate regulation and the collection of measurement informations is performed in the kernel, but all actions are controlled by the user space daemon.

We further implemented the LLRMP on the Switch 2000 configured for switching between 12 802.12 network segments. This implementation was used in the experiments in Chapter 7 to control the static priority scheduler in the switches along the data path. The LLRMP protocol mechanisms are basically identical to the mechanisms implemented at hosts. Switches however interconnect several segments and thus additionally have to make forwarding decisions for LLRMP control messages. The main problems we encountered during the switch implementation were caused by the slow operation of the switch's processor and the tight limit of just 2 Mbyte memory for the entire switch kernel.

6.4.2 Packet Classifier and Rate Regulator

The rate regulator and the packet classifier are implemented in the device driver of the 802.12 LAN adapter card. The classification is based on filter information provided by the LLRMP daemon. The filter may specify a single or a combination of parameters in the link-level-, the network-, or the transport protocol header of the data packet. The classification can thus for example be only based on the MAC multicast destination address, when these addresses are uniquely assigned within the LAN, or can use higher level information like the IP source address and the UDP source port number.

Each rate regulator is able to support the Time Window algorithm described in Section 6.3. It counts the number of packets passed into the output queue in each time frame *TF* and measures the data rate generated by the application over the time window *TW*. All statistics collected in the kernel are periodically passed to the LLRMP daemon. Each rate regulator also limits the number of data packets which can leave the regulator within *TF*. This limit is defined by the flow's packet count: $pcnt^i$. If a flow sends more data packets than allowed, then any surplus packets become delayed into the next time frame. For this, the packets are buffered in the flow's rate regulator queue. If this queue exceeds its bound then arriving data packets are dropped. This ensures that the service of other flows is not violated when an application e.g. by mistake passes a different traffic pattern to the network than previously negotiated.

6.4.3 Timer Issues

For our reservation scheme, we assume time frames *TF* of: 10 - 40 ms in order to keep the delay bounds low for network nodes with large bandwidth requirements such as bridges or servers. From Theorem 6.1 however follows that only $T \ll TF$, where T is the timer granularity of the rate regulators, ensures an efficient use of resources. If the time frame and the timer granularity are in the same order of magnitude, then the result is a poor bandwidth utilization. For $T = TF = 10$ ms for example, just 50% of the available resources can be reserved for data traffic. The rest must be left unallocated in order to ensure that worst-case guarantees are met.

Most operating systems on existing workstations however only provide a timer granularity of 10 ms. We solved this problem in our prototype by changing the timer granularity used on the test

workstations. We implemented a second, fast timer in the HP-UX kernel, which is able to provide granularities as fine as 100 µs on a 75 MHz machine. The function of the operating system (OS) was not affected since all OS routines are served at their usual times. Only kernel resident modules e.g. LAN device drivers can register for the fast timer and receive service at the lower processor level 5[1]. In the future, a fine granularity timer on the LAN adapter card would be an appropriate solution.

6.5 Performance Evaluation

In this section we discuss experimental results which we received for the throughput, the packet delay, the Time Window algorithm and the resource utilization. These were collected using the implementation outlined in the previous section. All measurements for the Guaranteed service were taken in single segment topologies. This was because our test switches only support simple static priorities and do not have rate regulators.

6.5.1 Throughput

To show the accuracy of the Bandwidth Test and of the results received for the Demand Priority overhead, we now compare the measured network throughput with results computed from Theorem 6.1. The measurement results and the experimental setup were already discussed in the analysis in Section 4.3.1 and Section 4.3.2. The numerical results for the per-packet overhead D_{pp} and the interrupt time D_{it} were taken from Table 5.5, Table 5.6, Table 5.8 and Table 5.9, respectively.

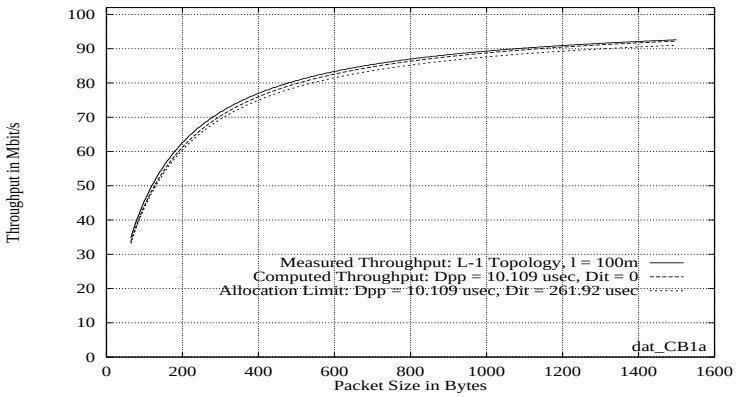


Figure 6.4: Comparison: Measured Throughput and Computed Allocation Limit in a Single Hub 802.12 Network using 100 m UTP Cabling.

The comparison for a single hub network using 100 m UTP cabling is shown in Figure 6.4. The upper curve is the measured worst-case throughput as shown in Figure 4.9 for this topology. The second curve is the computed worst-case throughput. It was computed assuming: (1) there is only one active flow, (2) a time frame of $TF = 20\ \text{ms}$, (3) a single hub topology with 100 m UTP

1. The system timer itself runs on processor level 7 which is the highest priority in the system.

cabling represented in a per-packet overhead of $D_{pp\_L1} = 10.109\ \mu s$, and (4) a low priority service interrupt time of $D_{it\_L1} = 0$. The third curve is the maximum resource allocation limit. It differs from the theoretical throughput such that the computation additionally considered the interrupt time for this topology, where $D_{it\_L1} = 261.92\ \mu s$. The computation of both graphs assumed a non-bursty flow and a timer granularity of $T = 0$ to show the accuracy of the admission control.

In Figure 6.4, one can observe that the measured throughput is always higher than the theoretical throughput computed with Theorem 6.1. This is important since the computed throughput is the basis for the allocation limit. The difference between the theoretical throughput and the allocation limit thus reflects the minimum capacity that is guaranteed to be available for the normal priority service. Some network resources must always be left unallocated since these are required to preempt the normal priority service. Figure 6.4 shows the worst case for this and thus the maximum allocation limit that can be achieved. If for example all real-time flows had a minimum average packet size of 512 byte or more, then bandwidth up to about 79 Mbit/s could theoretically be allocated. The actual available bandwidth however is guaranteed to be slightly higher, which is necessary for providing deterministic service guarantees. It can further be observed that the theoretical and the measured result match closely. This demonstrates the accuracy of the packet transmission model and of the results computed in Chapter 5. Resources could potentially be allocated almost up to the actually available network capacity.

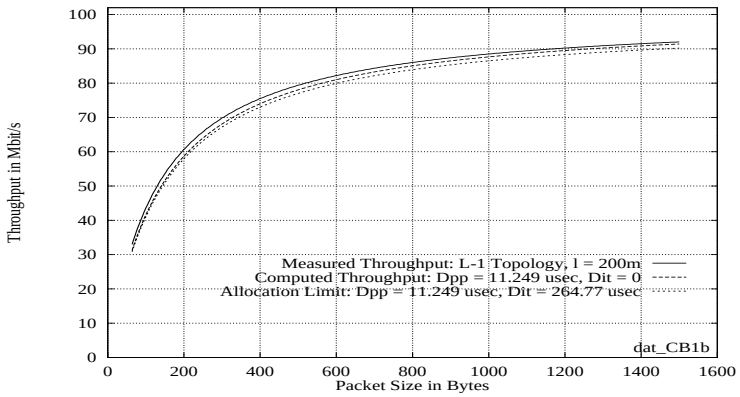


Figure 6.5: Comparison: Measured Throughput and Computed Allocation Limit in a Single Hub 802.12 Network using 200 m UTP Cabling.

Since the maximum supported UTP cable length for 802.12 networks is 200 m, we next compare the maximum throughput in such a topology. The results shown in Figure 6.5 are in general similar to the results in Figure 6.4, except that the throughput and the allocation limit are decreased for all packet sizes by a very small constant offset. This offset is caused by the additional propagation delay within the UTP cable.

The comparison shows that, despite the signalling overhead, the cable length does not have a significant impact on the worst case network performance when UTP cabling is used. This could be expected considering the results for the per-packet overhead in Table 5.5. The measurement results

for the throughput in Figure 6.5 are based on the same setup as described in Section 4.3.1 for the 100 m case, except a different UTP cable length. The allocation limit was computed using Theorem 6.1 with a packet overhead of $D_{pp\_L1} = 11.249\ \mu s$ and an interrupt time of $D_{it\_L1} = 264.77\ \mu s$.

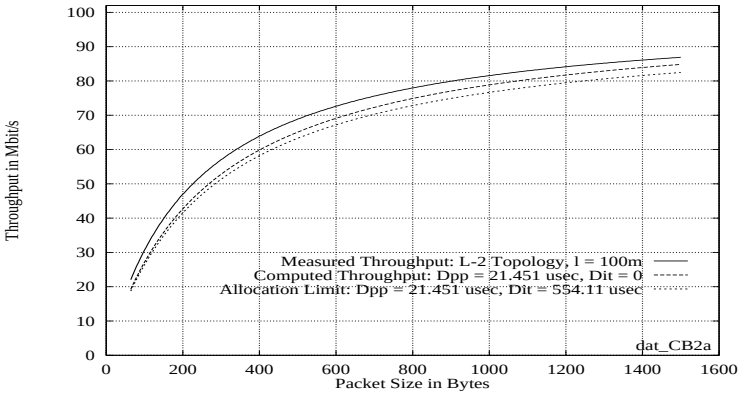


Figure 6.6: Comparison: Measured Throughput and Computed Allocation Limit in a Level-2 Cascaded 802.12 Network using 100 m UTP Cabling.

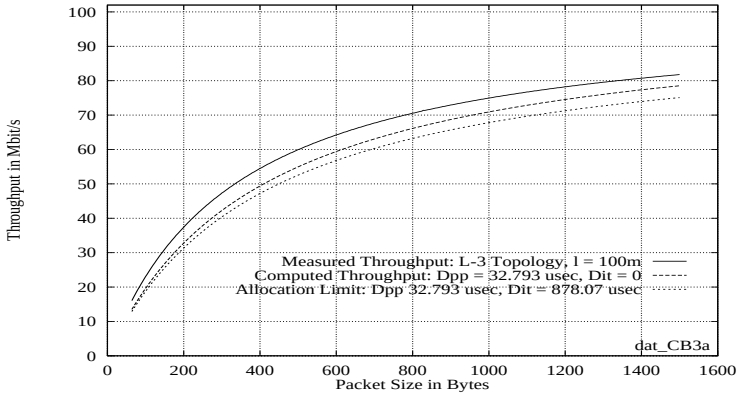


Figure 6.7: Comparison: Measured Throughput and Computed Allocation Limit in a Level-3 Cascaded 802.12 Network using 100 m UTP Cabling.

Figure 6.6 and Figure 6.7 show the equivalent comparison for the Level-2 and Level-3 cascaded 802.12 networks. The measured throughput curves are identical to the ones shown in Figure 4.9 for these topologies. The computed results were computed under the same assumptions as made for the single hub network, except that the computation used the topology specific results for the per-packet overhead and the interrupt time. In both figures, we can also observe that the measured throughput is always higher than the computed result and that both curves match closely. Looking at Figure 6.4 however shows that the results in Figure 6.6 and Figure 6.7 do not match as accurately as the results received for the single hub network.

The difference between the measured and the computed data throughput is caused by the worst-case character of the per-packet overhead D_{pp}. This overhead is computed by adding up the worst-case delay of all network components along the data path. In reality however, simultaneous worst case

conditions at all layers of the network stack e.g. at the MAC, PMI and PMD are rarely met, so that data packets on average are forwarded faster than described by our worst-case transmission model. For the 100 m UTP cables used in the experiments for example, we measured a propagation delay of about 480 ns using an oscilloscope. The standard however allows a maximum delay of 570 ns.

For the single hub network, we still receive the most accurate results because the data path between any two nodes only included two UTP links and one repeating hub. Higher cascaded topologies however have a longer maximum data path. Our Level-2 test network for example connected any two end-nodes via a chain that included 4 links and 3 repeating hubs. The differences in the delay between the model and the reality add up along the longer data path and thus decrease the accuracy between the measured and the computed throughput in higher cascaded topologies.

The difference between the computed throughput and the allocation limit in Figure 6.6 and Figure 6.7 has also become larger when compared with the results received for the single hub network. This is caused by the larger normal priority service interrupt time to be considered in these topologies. The measurement results obtained for the Level-4 topology further confirm the behaviour observed for the Level-2- and the Level-3 topology. They are however omitted here.

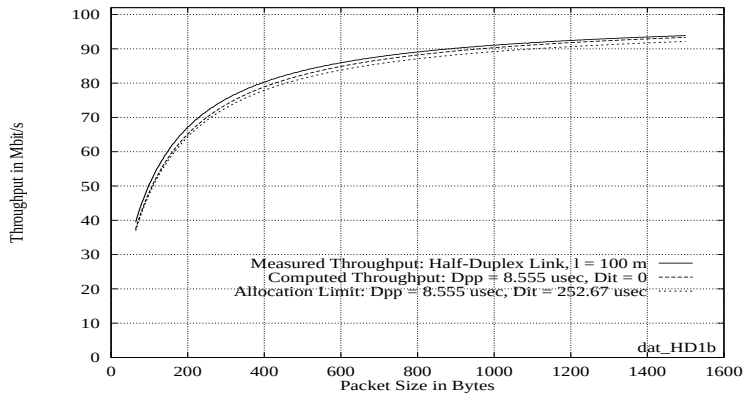


Figure 6.8: Comparison: Measured Throughput and Computed Allocation Limit for a 100 m UTP Half-Duplex Switched 802.12 Link.

The result for a half-duplex switched link is shown in Figure 6.8. It confirms that Theorem 6.1 is also able to provide accurate results for this topology when used with the topology specific per-packet overhead. In general, the same fundamental characteristics as discussed for the single hub network can also be identified for the half-duplex switched link. The measured throughput curve is identical to the result shown in Figure 4.10. The computation of the maximum allocation limit used a per-packet overhead of: $D_{pp\_HD} = 8.555\ \mu s$ and an interrupt time of: $D_{it\_HD} = 252.67\ \mu s$. This was performed based on the same assumption as made for the cascaded network topologies.

Finally, Figure 6.9 shows the resource allocation limit in a Level-2 cascaded network with a High Priority Utilization Factor of: $f = 0.6\ (60\%)$. The space between the second curve (the computed throughput) and the third curve (the allocation limit) represents the minimum bandwidth guaranteed

to be available for the best-effort traffic. The average available network capacity will however be higher than $1 - f$ because: (1) the resource allocation limit was determined based on worst-case assumptions (the worst-case computed throughput), and (2) any resources unused by Guaranteed service flows are immediately available for Best Effort traffic.

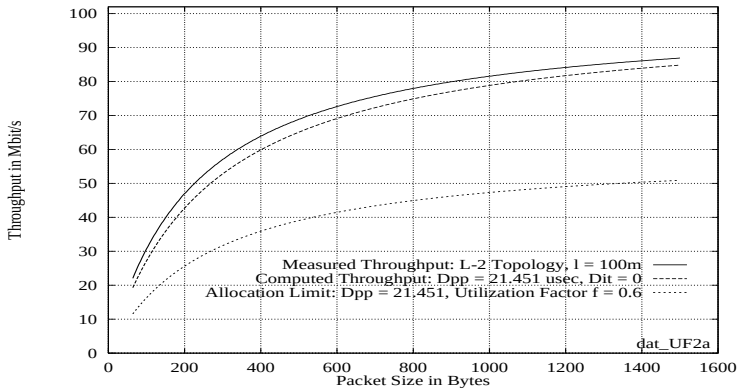


Figure 6.9: Resource Allocation Limit in a Level-2 Cascaded Network for a High Priority Utilization Factor of: $f = 0.6$.

6.5.2 Delay Characteristics

In the following experiments, we measured the end-to-end delay for data packets using the high priority service. These experiments had three goals: (1) to experimentally determine the operating system overhead in our test workstation, (2) to confirm that the delay bounds assigned by our allocation system are valid[1], and (3) to compare the measured maximum- and average delay with the deterministic upper bound.

In the first experiment, we measured the maximum end-to-end delay for a single high priority data source in dependence of the normal priority network load. All measurements were taken by the Measurement Client. The setup was basically identical to the one used in Section 5.2.5 to determine the interrupt time. It differed in respect to the normal priority cross traffic, which was now addressed with unicast. We thus only summarize the setup here. The test network was a single hub network. The Measurement Client generated data packets with a constant bit rate of 0.56 Mbit/s. The cross traffic was generated by 10 Normal Priority Traffic Clients and rate regulated at the link layer. Note that our rate regulators can also regulate normal priority traffic. This was exploited in this test.

After the measurement, we repeated the experiment three times while increasing the number of High Priority Clients. Figure 3.8 in Section 3.6 illustrates the equivalent setup for a Level-2 cascaded network. Each High Priority Client generated unicast data packets with (1) a constant data rate of 20 Mbit/s and (2) a length of 1500 bytes to show the worst-case impact. This used the traffic

1. This implies that all real-time data packets encounter a smaller delay than predicted by the admission control (basically Theorem 6.2).

generator described in Section 3.4. The results of all four experiments are shown in Figure 6.10. It contains the results for the maximum-, and the minimum end-to-end delay recorded by the Measurement Client.

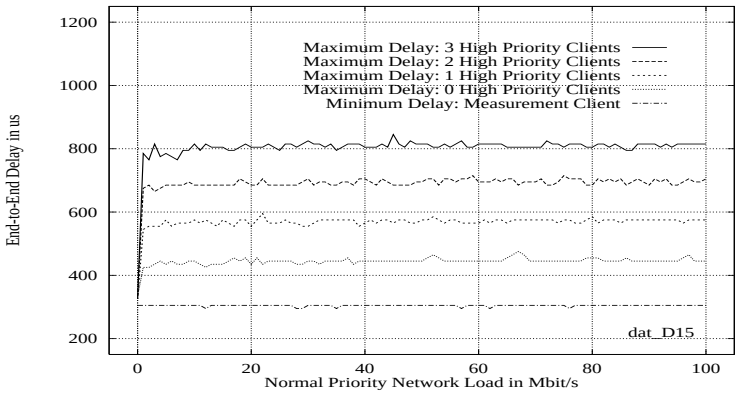


Figure 6.10: End-to-End Delay using the High Priority Service
in a Setup with several High Priority Traffic Clients.

The minimum delay is about 300 µs. This consists of 145 µs required for DMA-ing the data packet (twice: to and from the LAN adapter card) and flushing the cache, about 25 µs for the context switch, and about 130 µs for a single data packet transmission and the corresponding protocol overhead in the network. We can further observe that the maximum delay is bounded in all experiments and does not increase with higher network loads. This shows the isolation from the normal priority traffic in the network. The difference between the minimum and the maximum end-to-end delay increases with each new High Priority Client by about 130 µs - one maximum data packet transmission time plus Demand Priority protocol overhead. The maximum delay is encountered when the normal priority service is pre-empted and the Measurement Client is the last high-priority node to be served in the round-robin cycle. This experimentally confirms the approach taken by Theorem 6.2 which provides a tighter delay bound based on the round-robin service policy in the network.

In the second part of this section, we report results for the maximum-, the average-, and the minimum end-to-end-delay, which we measured for *vat*, *vic*, *OptiVision* and *MMC* application traces in a Level-2 cascaded network with normal priority cross traffic. The characteristics of the traces were discussed in Section 4.2.1. The trace driven measurement approach was described and evaluated in Section 3.2.2 and Section 3.4. Table 6.1 summarizes the source and the token bucket parameters used in the tests. The resources corresponding to Column 5 and Column 6 were allocated at the link layer using the LLRMP signalling protocol. We selected tight token bucket parameters (δ, r) with a rate r close to the average data rate of each flow to maximize the high priority network utilization. A delay bound of 10 ms was requested for all applications. Columns 7 in Table 6.1 shows the maximum length of the rate regulator queue at the source node. The results for the packet counts in Column 8 were estimated using the Time Window algorithm. The measurement results for this algorithm are discussed later in Section 6.5.3. Furthermore, all packets were sent using IP multicast.

The test network was a Level-2 cascaded network as shown in Figure 3.8. It however included two additional Level-2 hubs. This created a topology with five Level-2 hubs and one Root hub. The Measurement Client and the hubs were interconnected using 100 m Category 3 UTP cabling. The Traffic Clients were linked to the Level-2 hubs via a 5 m cable of the same type. In all experiments, the test network was always overloaded with normal priority data traffic. For this, we used two Normal Priority Traffic Clients. They generated constant bit rate data traffic with a total network load of about 89 Mbit/s - corresponding to the maximum data throughput in a Level-2 topology. This used the traffic generator in the kernel. The packet size was 1500 bytes to enforce maximum normal priority service interrupt times.

The Measurement Client and the High Priority Traffic Clients generated data traffic based on the application traces. In each experiment, we admitted homogeneous applications e.g. only *vic* flows or only *MMC1* flows until we reached the allocation limit. The timer granularity T of the rate regulators was 1 ms . The High Priority Traffic Clients and the Measurement Client had, whenever possible, an identical setup in respect to the type and the number of application flows generated. This simplified the measurement process since we did not have to measure the delay at High Priority Clients. Measurements were only taken for data packets generated on the Measurement Client. We can however assume that the basic results achieved for the Measurement Client such as the average delay are also valid for each High Priority Client since on average, they passed a similar traffic pattern into the shared network.

All measurements were carried out on a per-flow basis by measuring the end-to-end delay for each data packet generated for the selected flow. Any delay introduced by the rate-controller at the source node was not considered because our investigations were focused on the actual network behaviour. The measurement interval was 30 minutes for each individual experiment.

Test	Application Trace	Encoding Scheme	Delay Bound requested in ms	Per-Flow Link Layer Resources allocated			
				Data Rate r in Mbit/s	Burst Size δ in Bytes	Maximum Rate-Reg. Queue in Pkts	Packet Count considered (TF = 10ms)
1	vat	PCM2 audio	10	0.075	1500	3	2
2	vat	PCM2 audio	10	0.075	1500	3	2
3	vat	PCM2 audio	10	0.075	1500	3	2
4	vic	JPEG video	10	1.0	1500	16	5
5	vic	JPEG video	10	1.0	1500	16	5
6	vic	JPEG video	10	1.0	1500	16	5
7	OVision	MPEG-1 video	10	1.8	1500	137	7
8	OVision	MPEG-1 video	10	1.8	1500	137	7
9	OVision	MPEG-1 video	10	1.8	1500	137	7
10	MMC1	JPEG video	10	3.0	1500	62	8
11	MMC1	JPEG video	10	3.0	1500	62	8
12	MMC1	JPEG video	10	3.0	1500	62	8

Table 6.1: Source and Token Bucket Parameters for the Delay Tests in a Level-2 Cascaded Network.

Table 6.2 shows the measurement results. The first three columns of the table contain the test number, which corresponds to the number in Table 6.1, the name of the application trace and the total number of flows admitted in the test. The fourth column shows the deterministic delay bound provided by Theorem 6.2 for the Measurement Client after all flows had been admitted. The delay bounds for the High Priority Traffic Clients are always lower or identical to this bound. Topology information is given in Column 5 and Column 6. For each application trace, we carried out three experiments, in which we varied the number of High Priority Traffic Clients and the number of local flows on each network node. In Test 10 for example, we admitted a single 3 Mbit/s *MMC1* flow on 13 computers (12 High Priority Clients plus one Measurement Client).

In Test 11, the network contained five High Priority Clients and one Measurement Client. Each High Priority Client sent two 3 Mbit/s *MMC1* flows into the network, the Measurement Client generated three 3 Mbit/s *MMC1* flows in this experiment. The total number of flows admitted in each test was determined by the allocation limit, and implicitly, by the delay bound requested. A 14th *MMC1* flow could thus not have been admitted.

The difference between the delay bound requested (10 ms) and the provided upper bound shown in Table 6.2 is mainly caused by the use of the Time Window algorithm and its initial pessimistic assumption that a new flow only uses minimum sized packets for the data transmission. This requires more free resources at call admission due to the additional per-packet overhead to be considered. It can be shown that the 14th *MMC1* flow is rejected even though sufficient resources for supporting the flow are actually available in the network. This is because the admission control does not yet know that the new flow does not only use minimum sized packets. In high loaded networks, applications requesting a higher data rate will thus have a lower probability of being accepted.

Test	Trace	Number of Flows admitted	Delay Bound in ms	Topology Information		Measured Parameters						
				Nodes with Reser-vations	Number of Flows Per-Node	High Priority Data Rate in Mbit/s	Min. Delay in ms	Ave. Delay in ms	90 % in ms	99 % in ms	Max. Delay in ms	Ave. Packet Size in Bytes
1	vat	55	9.97	11	5	4.07	0.155	0.477	0.545	0.595	0.755	368
2	vat	55	9.97	5	11	4.07	0.095	0.468	0.545	0.595	0.695	368
3	vat	55	9.97	1	55	4.07	0.155	0.484	0.535	0.575	0.805	368
4	vic	26	9.32	13	2	23.89	0.105	0.611	0.715	0.915	1.685	934
5	vic	26	9.32	8	3, Mclient: 5	23.77	0.095	0.628	0.755	0.975	1.625	934
6	vic	26	9.32	2	13	23.91	0.105	0.628	0.735	0.955	1.725	934
7	OVision	18	8.98	9	2	21.49	0.235	0.734	0.845	1.045	1.965	1332
8	OVision	18	8.98	6	3	22.94	0.235	0.745	0.875	1.085	2.065	1332
9	OVision	18	8.98	1	18	22.77	0.235	0.757	0.885	1.385	2.225	1331
10	MMC1	13	8.64	13	1	38.90	0.145	0.746	0.875	1.105	2.055	1356
11	MMC1	13	8.64	6	2, Mclient: 3	38.90	0.135	0.752	0.875	1.255	2.445	1356
12	MMC1	13	8.64	2	6, Mclient: 7	38.86	0.115	0.771	0.955	1.615	2.545	1356

Table 6.2: Comparison: Computed and Measured Delay in a Level-2 Cascaded 802.12 Network.

Column 7 in Table 6.2 shows the high priority data rate measured over the measurement interval of 30 minutes. The results for *vat*, *vic* and *MMC1* are close to their allocation limit. The data rates observed for the *OptiVision*-tests are significantly lower since resources were over-allocated to avoid long maximum queuing delays in the rate regulator of the data source.

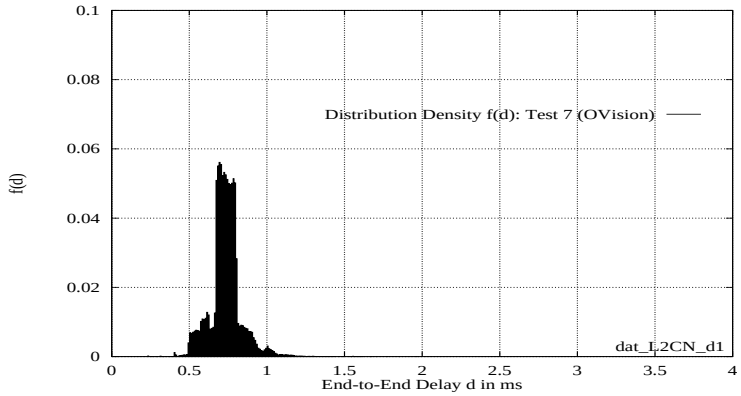


Figure 6.11: The Delay Distribution (Density) for the Results of Test 7 in Table 6.2.

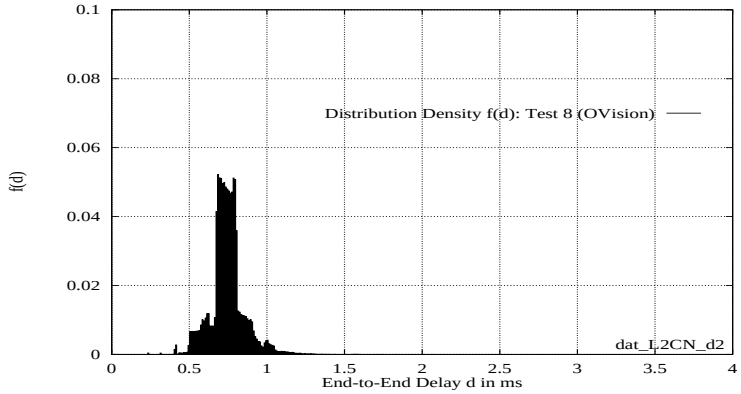


Figure 6.12: The Delay Distribution (Density) for the Results of Test 8 in Table 6.2.

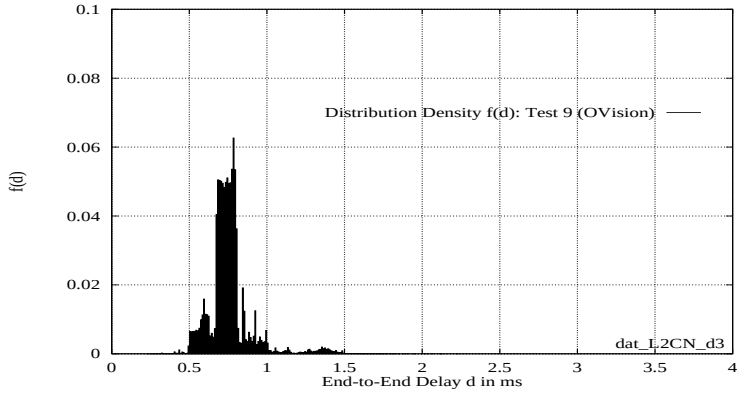


Figure 6.13: The Delay Distribution (Density) for the Results of Test 9 in Table 6.2.

The following 5 columns (8 - 12) contain the main results of the experiments. They show the minimum-, average-, 90th percentile, 99th percentile and the maximum end-to-end delay measured for a single *vat*, *vic*, *OptiVision* or *MMC1* flow. For the tests 7, 8 and 9 (*OptiVision*) in Table 6.2, the delay density and the corresponding distribution functions are shown in Figure 6.11, Figure 6.12, Figure 6.13 and Figure 6.14, respectively.

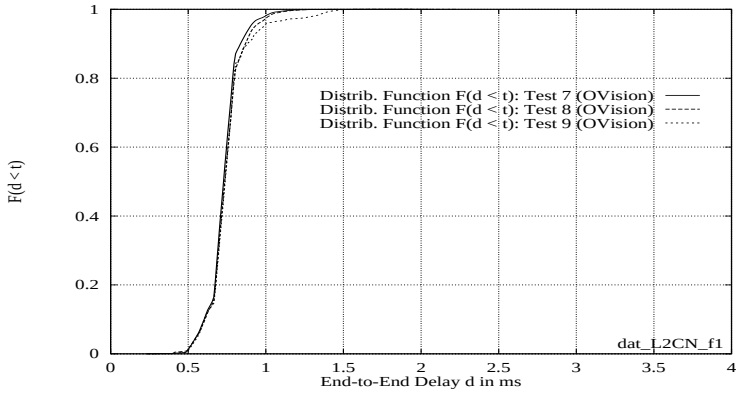


Figure 6.14: The Distribution Function for the Results of Test 7, Test 8, Test 9 in Table 6.2.

We found that all results for the average- and the maximum delay are significantly lower than the deterministic upper bound computed with Theorem 6.2. This was expected since: (1) simultaneous worst case conditions in the network and at all Clients are rare, and (2) several High Priority Clients were connected to the same Level-2 hub in our test network. The latter reduced the average Demand Priority signalling overhead because Level-2 hubs could sometimes subsequently serve data packets from several High Priority Clients. Since the available data rate in a Level-1- and a Level-2 network may differ by more than 10 Mbit/s, some transmission requests were thus served much faster than assumed in the worst case for the Level-2 topology. This increased the total throughput and thus reduced the delay.

It can further be observed that varying the network topology while keeping the total high priority load constant did not have any significant impact on the average delay. We assume that this is due to: (1) the rather low high priority network load, and (2) the fairness of the round-robin packet service policy which enforces a sufficient sharing of resources between all nodes in the network.

Given the low high priority utilization, the results for the average delay, especially those received for the *vat* and *OptiVision* traces, might at a first glance seem rather high when for example compared with results for the same load on a full-duplex 100 Mbit/s link. This is however caused by the interrupt time. To show the impact on the average delay received in Test 7 (*OptiVision*), we performed three additional experiments. In the first (1), we measured the average delay for the Measurement Client generating two *OptiVision* flows as carried out in Test 7 but without any other high or normal priority cross traffic on the network. (2) We then repeated the experiment using the same setup but additionally overloaded the network with normal priority unicast traffic.

The third experiment (3) differed from the second such that all normal priority cross traffic was now send using multicast. Figure 6.15 shows the results for all three tests. We further added the result of Test 7, whose setup additionally included eight High Priority Clients, each of which was generating two *OptiVision* flows. This is curve (4).

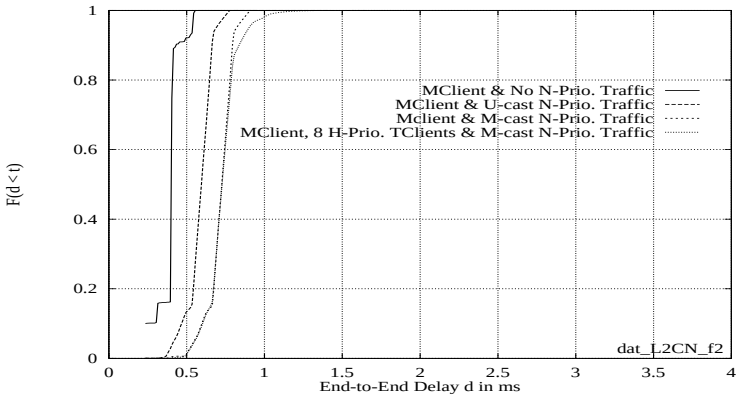


Figure 6.15: The Impact of the Interrupt Time on the Average Delay in Test 7 in Table 6.2.

One can observe that the average delay for the no cross traffic case (1) in Figure 6.15 is low. We measured 396 µs. This significantly increases in experiment (2) when unicast cross traffic is added. As in all previous tests, the normal priority cross traffic is generated at a data rate close to the network capacity. In contrast, the Measurement Client sent at a low data rate. For almost every high priority data packet transmitted, the normal priority service thus had to be interrupted which raised the average delay. It further increased when the cross traffic is sent using multicast as shown by the results for experiment (3). Finally, the additionally high priority traffic added in the fourth test did not have any significant impact on the average delay measured by the Measurement Client. It only changed the tail of the distribution.

The results in Table 6.2 have shown that the network is capable of providing very small end-to-end packet transmission delays. We believe that these are sufficient for supporting existing time critical applications. The use of the priority access combined with admission control guarantees that these delays remain very low when the normal priority network load is high, or when the shared network incorporates many more hubs and nodes as used in our experiments.

6.5.3 Results for the Time Window Algorithm

The tests reported in this section had two goals: (1) to show that the Time Window algorithm is able to find an accurate upper bound for the packet count and thus for the Demand Priority overhead, and (2) to confirm that it is sufficiently conservative such that no service violation occurs.

So far we tested the algorithm using the applications: *vat*, vic, *nv, MMC* and the *OptiVision MPEG Communication System*. *vat*, *vic*, *MMC* and *OptiVision* were discussed in Section 4.2.1. *nv* [Fred94] is a video conferencing tool for the Internet. It was used as well since it is publicly available[1]. In

each test, we recorded the packet count estimation process and the data rate generated by the application over a measurement time interval of at least 15 min. We further varied, where possible, the parameters of the input source and temporarily switched off the source itself, in order to enforce large scale data rate variations. The estimation process itself was also restricted. At the end of each time window *TW*, the packet count $pcnt$ was only updated when the new value *pcnt*' was *smaller* than the existing estimation. This reduced the number of updates and thus minimized the LLRMP signalling overhead in the network. The packet count could thus have only been increased if a sample had reached the corresponding high watermark. This however never happened in any of the tests we performed in the context of the Guaranteed service. The parameters of the measurement algorithm, which we used in all experiments are shown in Table 6.3.

Measurement Time Window *TW*	40 s
Allocation Time Frame *TF*	20 ms
Timer Granularity *T*	1 ms
α	1
κ	2
Minimum Packet Size P_{min}	64 byte

Table 6.3: Parameters of the Time Window Algorithm used for the Packet Count Estimation.

In the first experiment, we tested *vic* operating in the configuration described in Section 4.2.1. At the link layer, we allocated 1 Mbit/s for application data using the LLRMP. The burst size δ was 1500 bytes in all experiments reported in this section. The video camera was switched off during the time intervals: 0 - 120 s, 480 - 540 s and 780 - 840 s. Figure 6.16 shows the measured data rate, Figure 6.17 the packet count estimation process. The upper curve in Figure 6.17 represents the bound for the packet count (*pcnt*) estimated by the algorithm. The lines at the bottom of the diagram show the maximum samples (*scnt*) measured during the test. It can instantly be observed that there is an upper bound on the number of data packets generated by *vic* within each time frame.

The estimation process starts after the flow is admitted. This is at time 0. The initial value for the packet count is *MAX_PCNT*, which is 42 in this setup. It reflects the worst case, in which the algorithm assumes that *vic* only generates minimum sized data packets. The estimated packet count does not change until *vic* starts sending video data (at time 120 s) because the parameter β in Equation 6.25 causes any new estimate to be *MAX_PCNT*. As the data rate approaches the allocation limit of 1 Mbit/s, the algorithm is able to find more accurate estimations for the maximum packet count actually used by this application. The most accurate bound in this test is found after about 430 seconds. It is retained despite the fact that the data rate changes later since we only increase *pcnt* when an individual measurement sample (*scnt*) reaches the high watermark. This however never occurs as can be observed in Figure 6.17.

The next application tested was *MMC* operating in video conferencing mode. This used the configuration described in Section 4.2.1 for the *MMC1* trace. We allocated a bandwidth of 3 Mbit/s at the

1. *nv* can be found under: ftp://ftp.parc.xerox.com/pub/net-research/.

link layer. The video camera was switched off during the time intervals: 180 -300 s, 540 - 660 s and 780 - 840 s. Figure 6.18 (a) shows the results. In contrast to the first test, the algorithm found an accurate estimation within a single *TW* interval. This is because MMC instantly used all the resources reserved for it. The estimation result was retained through the entire test since there is again no measurement sample that reaches the high watermark. Such an estimation process is desired for each real-time flow in the network since it minimizes the number of resource adjustments at the resource arbiter in the network.

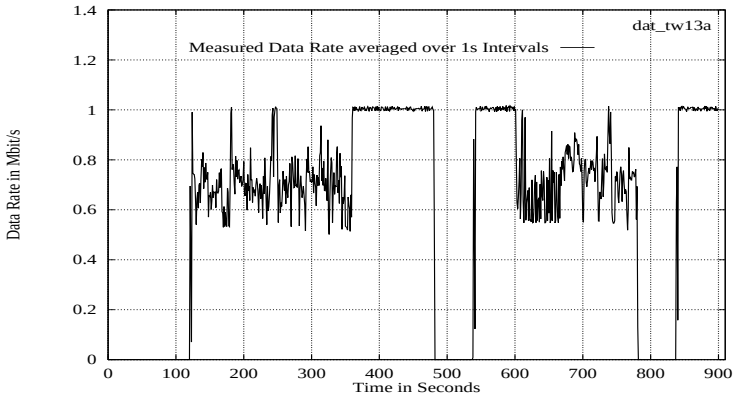


Figure 6.16: Data Rate generated by *vic* during the Packet Count Estimation.

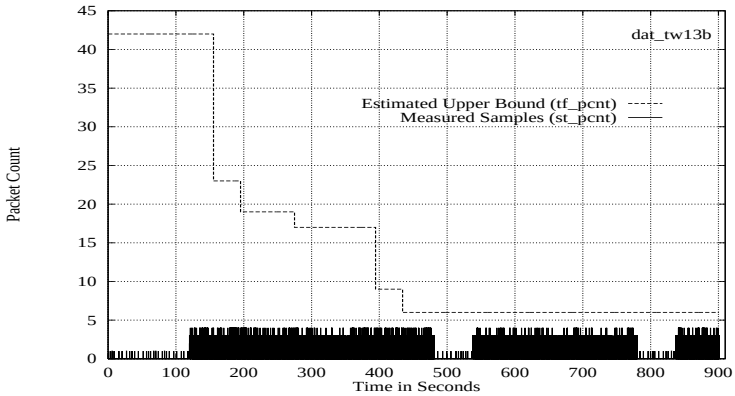
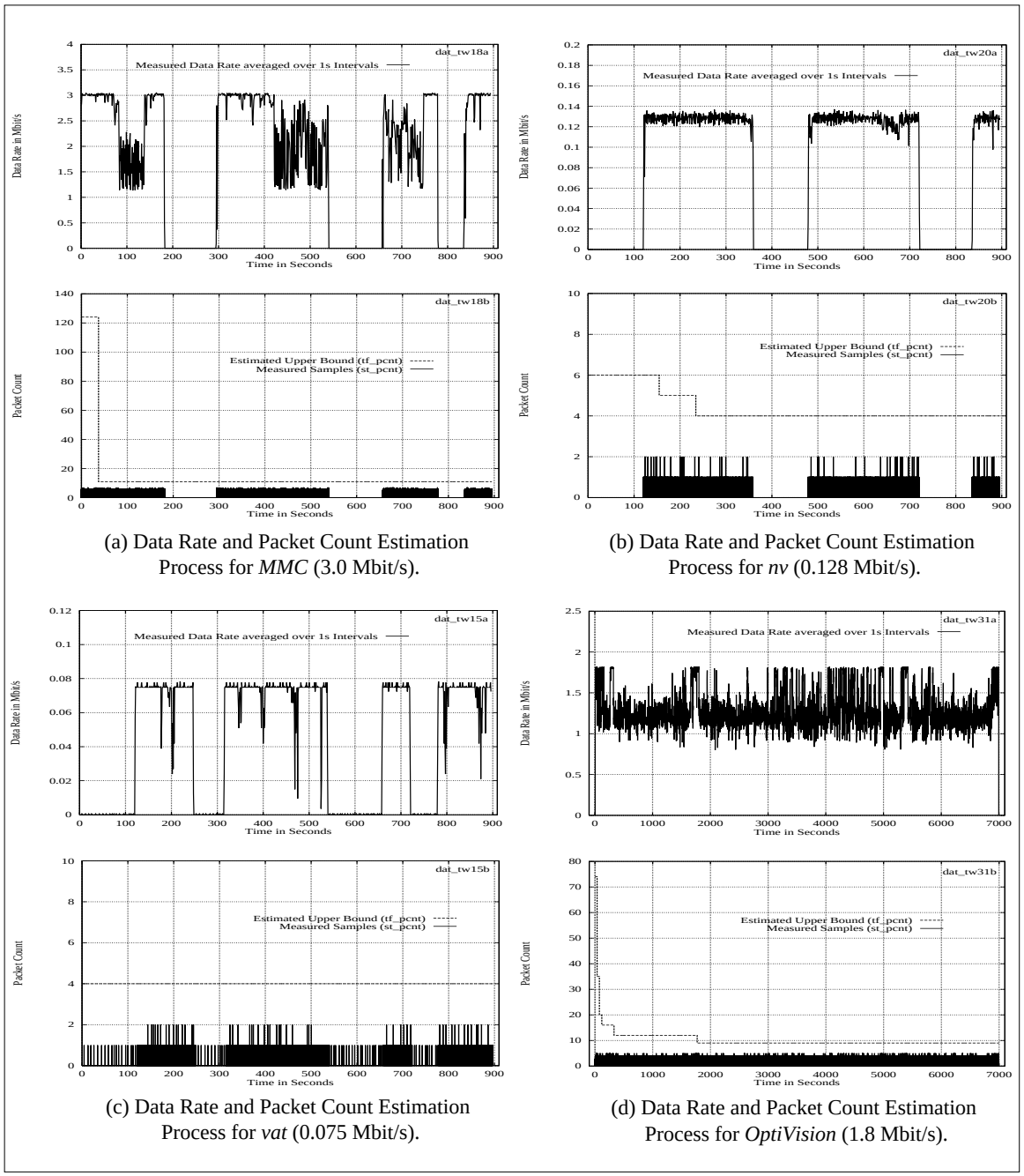


Figure 6.17: Packet Count Estimation Process for *vic*.

Similar experiments as reported for *vic* and *MMC* were also carried out for *vat*, *nv* and *OptiVision*. Example results for these applications are shown in Figure 6.18 (b - d). The configurations of *vat* and *OptiVision* were identical to the ones described in Section 4.2.1 for these applications. We allocated 0.075 Mbit/s and 1.8 Mbit/s at the link layer, respectively. Note that Figure 6.18 (d) shows the estimation process over the entire 2 hour adventure movie *Jurassic Park*. *nv* (version 3.3 beta) generated a compressed video stream with a data rate of about 0.128 Mbit/s in the test.

Hardware support was provided by an HP A.B9.01.3A frame grabber card. The picture resolution was 320 x 240 pixel (8 bit colours). We allocated 0.128 Mbit/s. For all applications, we repeated the

test and varied, where possible, the data rate generated and the data encoding scheme used. All measurement results are similar to the ones discussed for *vic* and *MMC*. They only differ in respect to: (1) the traffic pattern and the samples measured, (2) the adaptation rate and (3) the difference between the worst-case packet count and the estimated upper bound.



(a) Data Rate and Packet Count Estimation Process for *MMC* (3.0 Mbit/s).

(b) Data Rate and Packet Count Estimation Process for *nv* (0.128 Mbit/s).

(c) Data Rate and Packet Count Estimation Process for *vat* (0.075 Mbit/s).

(d) Data Rate and Packet Count Estimation Process for *OptiVision* (1.8 Mbit/s).

Figure 6.18: The Packet Count Estimation Process for the Applications: *MMC*, *nv*, *vat* and *OptiVision*.

The experiments showed that if an application generates data with a rate close to the resources allocated for it, then the measurement algorithm is able to find an accurate upper bound for the packet count actually used. The difference (estimation gain) between the worst case (*MAX_PCNT*) and the final estimated upper bound (*pcnt*) depends on the packet sizes used and on the data rate. The gain observed in the first two experiments was large because *vic* and *MMC* generated data at a high rate and mainly used large sized data packets. No benefit will be achieved when applications use small sized packets or only generate a low bitrate data stream. This can for example be observed for *vat* and is caused by: (1) the conservativeness of the algorithm, and (2) the small difference between the worst case packet count ($MAX\_PCNT = 4$) and the maximum sample measured ($scnt_{TW} = 2$). No gain could possibly be achieved for low bitrate flows e.g. a 20 kbit/s audio flow because the worst case packet count is already one ($MAX\_PCNT = 1$). This however is the minimum number of packet overheads to be reserved for an application in a time frame. It can not be decreased.

In all measurements carried out so far, we did not detect a service violation for a single data packet. This could be observed despite that all applications changed their data rate in a large scale. We also did not observe the case that an individual measurement sample ($scnt$) reached the high watermark and caused the reallocation of resources. We thus believe that the algorithm can be used to estimate the packet overhead for applications using the guaranteed service, provided that the packetization process is constant. This is for example the case for the IP packet fragmentation mechanism implemented in HP-UX 9.05. We expect that other applications using the same mechanism e.g. to break up a large video frame into single data packets will generate a similar sample pattern as observed for *MMC*. Further generalizations can be made within the bounds of the Controlled Load service due to the weaker service commitment.

6.5.4 Resource Utilization

Table 6.4 shows the maximum number of *vat*, *nv*, *vic*, *OptiVision* and *MMC* flows which our allocation scheme was able to simultaneously admit in a single hub network while guaranteeing a certain deterministic delay bound. The same utilization can be achieved in a bridged network composed of several segments of the same type due to the rate controlled service discipline in bridges. Since the number of admitted flows depends on the traffic characteristics of the flows, in particular the data rate and the packet size distribution, we used the characteristics of our test applications for admission control. The results for the packet count are based on the use of the Time Window algorithm.

The goal of this section is to show the maximum high priority resource utilization that can be achieved for a set of test applications by using the allocation scheme in a realistic setup. A generalization of the results for other applications can not easily be made since these applications may have different traffic characteristics e.g. use smaller packet sizes for the data transmission, which then requires the allocation of additional resources. A higher utilization can be achieved when the packet sizes are fixed since this removes the overhead introduced by the Time Window algorithm. This however is less realistic considering currently available applications and operating systems.

The admission control was based on Theorem 6.1 and Theorem 6.2 using the network parameters for a single hub network with 100 m UTP cabling. Following the worst-case model, each flow was first admitted assuming the use of only minimum sized packets. For all existing flows, the admission control used the application specific packet count (*pcnt*) measured during the experiments in the previous section. The application details for *vat*, *vic*, *OptiVision* and *MMC* (*MMC1* type setup) can be found in Section 4.2.1. The configuration of *nv* was described in the previous section. Note that flow arrival and lifetime statistics were not considered in these experiments since we focused on determining the highest utilization in a pre-defined setup.

In Column 5, Table 6.4 shows the maximum number of flows (N_{max}) that could be admitted for three different time frames: 10 ms, 20 ms and 40 ms. The delay bound requested for all flows was always equal to the time frame. The timer granularity T was 1 ms, the burst size δ was 1500 bytes. We further always admitted homogeneous flows. Each row in Table 6.4 provides the result for one application in a given setup: e.g. for a time frame of *TF = 20 ms*, a maximum of 49 *vic* flows, each generating data at a rate of about 1 Mbit/s , can be simultaneously admitted while providing a deterministic delay bound of 19.404 ms for each of them.

The maximum high priority network utilization is computed by relating the allocated bandwidth to the maximum allocation limit. The maximum allocation limit is the maximum capacity that can be allocated when all data is sent with *maximum* sized packets. Since it is fixed for each topology, we used it as reference value for computing the network utilization. For a single hub network and a time frame of 20 ms, the maximum allocation limit is 91.02 Mbit/s. This leads to a maximum high priority network utilization of 53.83% for the 49 1 Mbit/s *vic* flows.

Time Frame *TF* in ms	Delay Bound (Theorem 6.2) in ms	Application	Per-Flow Data Rate allocated in Mbit/s	Max. Number of Flows admitted (N_{max})	Packet Count (*pcnt*) measured	Total Bandwidth allocated in Mbit/s	Maximum High Priority Network Utilization (%)
10	9.912	vat	0.075	65	2	4.88	5.43
	9.962	nv	0.128	59	3	7.55	8.41
	9.801	vic	1.0	34	5	34.00	37.86
	9.592	OVision	1.8	24	7	43.20	48.10
	9.287	MMC	3.0	17	8	51.00	56.78
20	19.995	vat	0.075	112	4	8.40	9.23
	19.931	nv	0.128	105	4	13.44	14.77
	19.404	vic	1.0	49	6	49.00	53.83
	19.110	OVision	1.8	32	9	57.60	63.28
	18.347	MMC	3.0	21	11	63.00	69.21
40	39.918	vat	0.075	197	5	14.78	16.13
	39.896	nv	0.128	170	6	21.76	23.75
	38.759	vic	1.0	61	10	61.00	66.58
	37.993	OVision	1.8	37	16	66.60	72.69
	36.787	MMC	3.0	24	17	72.00	78.58

Table 6.4: Maximum High Priority Network Utilization in a Single Hub Network.

Several observations can be made in Table 6.4. The maximum utilization achieved for low bitrate flows such as *vat* or *nv* is low. This has two reasons: (1) the small sized data packets used by *vat* and *nv*, and (2) the allocation overhead. The impact of the packet size on the data throughput in the network was discussed in Section 4.3.1. The allocation overhead is caused by the fact that the allocation scheme always reserves resources for one maximum size data packet in each time frame to ensure that deterministic service guarantees are met. This is required since the time frames of different nodes in the network are not synchronized. For flows with a data rate larger than: P_{max}/TF , this does not create any overhead. For low bitrate flows however, additional resources need to be reserved to cover the worst case.

The allocation overhead could be reduced, at the expense of a more complicated allocation system, by: (1) introducing a synchronization mechanism between high priority network nodes, or (2) by using a lower bound for the maximum packet size used by each flow. We however believe that the utilization in the existing scheme is sufficient so that such mechanisms are not necessary. For higher bitrate streams e.g. 1 Mbit/s *vic* flows, a much higher utilization can be achieved because of the smaller overhead and the larger allocation limit. An increase of N_{max} can further be observed for all applications in Table 6.4 when larger delay bounds and time frames are used.

Table 6.5 shows the equivalent results for the Level-2 Cascaded Network using 100 m UTP cabling. Similar observations as discussed for the single hub network can be made: the maximum resource utilization is low when only low bitrate flows are admitted, but increases for flows with large reservations. A comparison with the results in Table 6.4 shows that in a Level-2 network, as expected, less flows can be admitted for all applications.

Time Frame *TF* in ms	Delay Bound (Theorem 6.2) in ms	Application	Per-Flow Data Rate allocated in Mbit/s	Max. Number of Flows admitted (N_{max})	Packet Count (*pcnt*) measured	Total Bandwidth allocated in Mbit/s	Maximum High Priority Network Utilization (%)
10	9.967	vat	0.075	55	2	4.12	5.15
	9.880	nv	0.128	47	3	6.02	7.51
	9.323	vic	1.0	26	5	26.00	32.45
	8.981	OVision	1.8	18	7	32.40	40.43
	8.635	MMC	3.0	13	8	39.00	48.67
20	19.829	vat	0.075	87	4	6.53	7.91
	19.867	nv	0.128	83	4	10.62	12.88
	18.902	vic	1.0	40	6	40.00	48.49
	18.521	OVision	1.8	26	9	46.80	56.74
	17.315	MMC	3.0	17	11	51.00	61.83
40	39.770	vat	0.075	152	5	11.40	13.63
	39.708	nv	0.128	130	6	16.64	19.89
	37.779	vic	1.0	50	10	50.00	59.77
	36.590	OVision	1.8	30	16	54.00	64.55
	34.847	MMC	3.0	20	17	60.00	71.72

Table 6.5: Maximum High Priority Network Utilization in a Level-2 Cascaded Network.

The largest difference in the allocated bandwidth can be observed for MMC flows. Even though only 4 flows less became admitted in the Level-2 network, the allocated bandwidth decreased by 12 Mbit/s for all time frames.

Table 6.6 provides results for the case that the time frame and the delay bound differ. The first row (*TF* = 10 ms) is identical to the one in Table 6.4 and was added for comparison. The following rows show results based on larger time frames. The requested delay bound (10 ms) is left constant in all tests. The highest utilization is achieved when the time frame is identical to the delay bound. The utilization significantly decreases when many flows requesting a lower delay bound become admitted. This is due to the more coarse-grained traffic control performed by the admission control when larger time frames are used. In this case, more traffic needs to be considered by Theorem 6.2 in computing the delay bound. This leads to larger results since the packet sizes are unknown and worst-case assumption must be made to comply with the deterministic service guarantees.

The utilization loss can be viewed as the cost for the higher flexibility in the admission control. It also suggests that the time frame should be decreased whenever the majority of admitted flows requested a lower delay bound than provided by the time frame itself. Finding the optimum such that a maximum resource utilization is achieved, is left for further study.

6.5.5 Performance Parameters

Our reservation scheme has four system parameter which determine its performance. These are: (1) the per-packet overhead D_{pp}, (2) the normal priority service interrupt time D_{it}, (3) the time frame *TF*, and (4) the timer granularity *T* of the rate regulators.

Time Frame *TF* in ms	Delay Bound (Theorem 6.2) in ms	Application	Per-flow Data Rate allocated in Mbit/s	Max. Number of flows admitted (N_{max})	Packet Count (*pcnt*) measured	Total Bandwidth allocated in Mbit/s	Maximum High Priority Network Utilization (%)
10	9.912	vat	0.075	65	2	4.88	5.43
	9.962	nv	0.128	59	3	7.55	8.41
	9.801	vic	1.0	34	5	34.00	37.86
	9.592	OVision	1.8	24	7	43.20	48.10
	9.287	MMC	3.0	17	8	51.00	56.78
20	9.952	vat	0.075	55	4	4.12	4.53
	9.815	nv	0.128	51	4	6.53	7.17
	9.247	vic	1.0	21	6	23.00	25.27
	9.097	OVision	1.8	15	9	27.00	29.66
	8.874	MMC	3.0	9	11	29.66	29.66
40	9.924	vat	0.075	48	5	3.60	3.93
	9.821	nv	0.128	41	6	5.25	5.73
	9.728	vic	1.0	14	10	14.00	15.28
	9.440	OVision	1.8	8	16	14.40	15.72
	7.871	MMC	3.0	4	17	12.00	13.10

Table 6.6: Maximum High Priority Network Utilization for different Time Frames in a single Hub Network.

The impact of D_{pp} and D_{it} on the network performance could be observed in Section 6.5.1. A discussion of these parameters is thus omitted here. Instead, we focus on the performance trade-offs made in setting the time frame and the timer granularity.

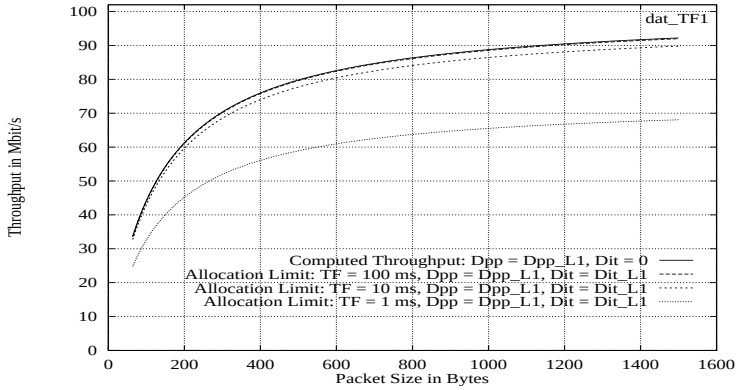


Figure 6.19: Impact of the Time Frame on the Allocation Limit in a Single Hub Network using 100 m UTP Cabling .

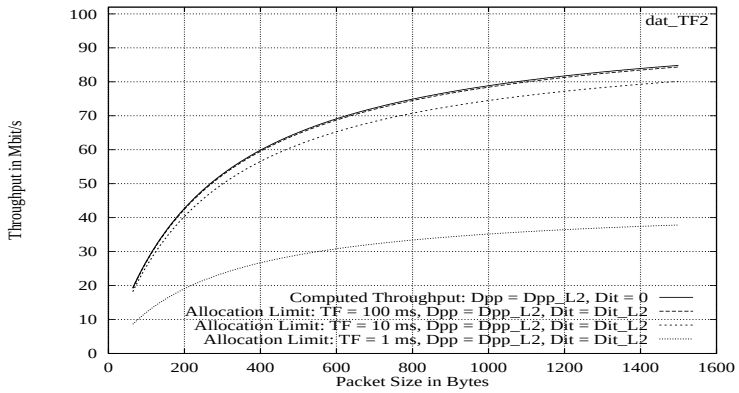


Figure 6.20: Impact of the Time Frame on the Allocation Limit in a Level-2 Cascaded Network using 100 m UTP Cabling.

The results in Figure 6.19 shows how the time frame affects the maximum resource allocation limit. The upper curve is the computed throughput. It it identical to the result in Figure 6.4 in Section 6.5.1. The other three curves represent the maximum allocation limit for different time frames used in the admission control. As in Section 6.5.1, we assumed a non-bursty flow and a timer granularity of $T = 0$ in the computation. Note that under these assumptions, the result for the computed throughput is independent of the time frame since $D_{it} = 0$. This curve can thus be used as a reference.

We can observe that the allocation limit significantly decreases when small time frames such as $TF = 1$ ms are used for the allocation. This is due to the interrupt time D_{it} which must be left unallocated within each time frame. In contrast, for large frames e.g. $TF = 100$ ms, the small value

of D_{it} has hardly any impact, which leads to a larger resource utilization. Small time frames are nevertheless desirable because they keep the overall delay bound low. Fortunately, the allocation limit and the length of the time frame are not linearly related. We believe that frames in the order of 10 - 20 ms represent a reasonable compromise. For this range, we obtain an acceptable resource utilization and a useful bound on the maximum end-to-end delay. For time frames larger than 20 ms the allocation limit still increases but the gain is not as large any more as can be observed in Figure 6.19.

Similar characteristics can be identified for cascaded networks. Figure 6.20 shows the equivalent results for the Level-2 topology. Due to the higher interrupt time, the loss is more significant when large time frames are used. For a time frame of 10 ms however, we still receive an acceptable result.

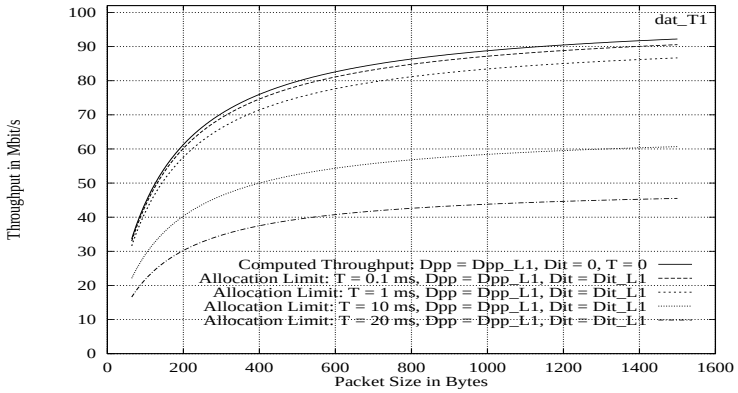


Figure 6.21: Impact of the Timer Granularity on the Allocation Limit (*TF* = 20 ms) in a Single Hub Network using 100 m UTP Cabling.

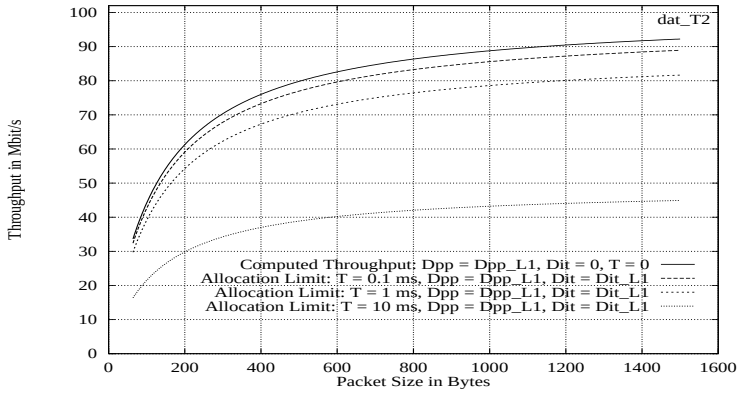


Figure 6.22: Impact of the Timer Granularity on the Allocation Limit (*TF* = 10 ms) in a Single Hub Network using 100 m UTP Cabling.

Figure 6.21 and Figure 6.22 show the impact of the timer granularity *T* on the resource allocation limit. A large timer granularity leads to bursty traffic when the corresponding rate regulator queue is always filled such that a burst of data packets is sent at the end of each timer interval *T*. If the timer granularity and the time frame are in the same order of magnitude, then this additional burstiness

affects the allocation limit because it requires additional resources to be allocated. These resources ensures that the deterministic delay bound is also met under the new burstiness constraints.

As could be expected, the allocation limit increases when the timer granularity decreases. The gain however is again not linear, but increases more slowly for smaller *T*'s. Whenever a large high priority resource utilization is required and time frames are in the order of 10 - 20 ms, then the timer granularity should be at least 1 ms or smaller. In our software prototype, a granularity of 1 ms seemed to be a good compromise between efficiency and processing overhead. This will however depend on the characteristics of the workstation. Hardware implementations of rate regulators e.g. on LAN adapter cards should however consider finer granularities than that. From the results in Figure 6.21 and Figure 6.22, we suggest a value of 100 μs.

6.6 Related Work

In [GrSp93] the Target Transmission Time (TTT) technique was proposed for allocating resources on Demand Priority networks. The admission of real-time flows is based on a nominal time value: the TTT. Network capacity is allocated by reserving a certain fraction of the TTT for each real-time flow in the network. A flow generating one 1500 byte data packet every 8 ms would for example consume 0.120 ms, provided TTT = 8 ms. Reservation requests are rejected when the sum of the transmission times of the new flow and all already admitted flows would exceed the TTT. The TTT is thus the delay bound for all real-time flows admitted. The TTT allocation is independent of the underlying MAC access protocol. The basic idea is identical to our time frame concept and the Busy Period interval [Cruz91a]. The paper however only reports preliminary results. Numerical admission control conditions were not provided. It further does not address the variable throughput issue. The network capacity to be used by the TTT admission control is thus unclear.

The VGAnet Suite [ChNa97], [ChNa98] is a real-time transport protocol suite for providing quality of service in 802.12 (100VG-AnyLAN) networks. It consists of: (1) the Real-Time Connection Management Protocol (RCMP) - responsible for the connection management, (2) the Real-Time Data Transfer Protocol (RDTP) - used for transmitting data over an established connection, and (3) the Local Resource Management Protocol (LRMP) - which controls the local access to the network. The resource allocation and admission control is performed by RCMP as part of the connection setup. From [ChNa97], we have for this: $b_R + \sum_{i \in H} b_i < B$, where b_R denotes the requested bandwidth, H the group of all already admitted flows, b_i the bandwidth allocated for real-time flow i, and B the high priority bandwidth allocation limit. This condition is the Simple Sum approach. The authors suggest an upper bound of 80 Mbit/s for B. Since the admission control does not consider the Demand Priority overhead, it cannot accurately determine the available bandwidth on the network. Deterministic service guarantees can thus only be given when B is set to the worst-case data throughput. This is 35.13 Mbit/s for a single hub network and further decreases for higher cascaded networks as we observed in Section 4.3.1.

Apart from [GrSp93] and [ChNa97] we are not aware of any other scheme for allocating resources in Demand Priority networks. The support of service guarantees over LANs has however been investigated for other link technologies. In [ACZ92], [ACZ93], [ShZh93], [ACZD94], [ZhBu95] the real-time performance of the timed token protocol as used in FDDI or IEEE 802.4 has been studied. In [ACZ92] and [ACZ93], [ACZD94] resources are allocated such that the sum of the synchronous network capacities allocated to all nodes in the ring does not exceed the available portion of the Target Token Rotation Time (TTRT). Formally, this is provided when condition [ACZ93]: $\sum_{i \in n} H_i \leq TTRT - \tau$ holds, where H_i denotes the capacity allocated to node *i*. τ represents the network overhead including factors such as the ring latency. The parameter *n* denotes the number of network nodes with reservations in the ring. [ACZ93], [ACZD94] then analyses several schemes for allocating the capacity H_i to each node *i* while meeting a certain deterministic delay bound. This is extended in [ZhBu95] in respect to the generality and tightness of the result. The same fundamental allocation strategy as given by the above condition underlies the admission control used in [ShZh93]. The paper additionally investigates network performance parameters to maximize the throughput for best-effort traffic, while guaranteeing a delay bound for real-time traffic. The smallest bound that can be provided by all these schemes is given by the sum of the high priority medium access time ($2 \cdot TTRT$) and the transmission time for one maximum size data packet.

The authors of [BPSW95] investigated the use of priorities in 802.5 token-ring networks and provide simulation results for the medium access delay and the queueing delay of priority and best-effort traffic. The medium access time t_{access} in a network with *m* nodes transmitting high priority data is bounded by [BPSW95]: $t_{access} \leq (m + 1) \cdot t_{max}$, where t_{max} denotes the token holding time at node *i*. This is required to built a Guaranteed service. Admission Control conditions however were not provided.

[VeCh95] and [Venk97] report the design and the implementation of a software based timed-token protocol that provides performance guarantees on existing Ethernet hardware. A network node may only send data when it possesses the token. This applies for real-time and non-real-time data. Resources are allocated in respect to the token holding time *THT* which corresponds to a certain data transmission time on the network. The admission control is performed based on the condition [VeCh95]: $THT_{RTnew} + T_{NRT} + \sum_{i \in n} THT_{RTi} \leq TRT$, where THT_{RTnew} represents the resources to be reserved for the new flow. T_{NRT} and THT_{RTi} are the resources allocated for all non-real time flows in the network and for real-time flow *i*, respectively. The parameter *TRT* denotes the token rotation time. The relation between the Token Holding Time and the corresponding data rate for nodes sending real-time and non-real-time traffic can be found in [VeCh95].

All these schemes are explicitly or implicitly based on a time frame mechanism. Network capacity is allocated by assigning fractions of the time frame (or the token rotation time) to admitted real-time flows. The delay bound depends on the token rotation time. Our allocation scheme also uses a time frame concept. The time frame *TF* is an upper bound for the sum of the queuing- and the propagation delay for all real-time data packets transmitted. Further, it is not necessarily the minimum

delay bound that can be given by the allocation scheme. The corresponding term to the medium access time in Token Ring networks ($2 \cdot TTRT$) is the normal priority service interrupt time D_{it}.

The significant Demand Priority protocol overhead and the simple round-robin service policy differentiate our environment from that of a token ring network (or of point-to-point links connected to an ATM switch). In Demand Priority networks, we can not assume that data held in output queues are served with a constant total data rate, even though the physical link speed is constant. Instead, the data throughput will depend on the packet sizes used by all nodes in the cascaded network as we could observe in the results in Section 4.3.1 and Section 4.3.2. The packet size may also be variable within each flow. Furthermore, in 802.12 networks, hubs are not able to identify and isolate single flows. The output queues are distributed and packets from different hosts can not be scheduled in the order they arrived at the output queue. This makes the analysis of our system more complicated, and is the reason why solutions designed for other technologies do not apply to our environment.

6.7 Summary

In this chapter, we proposed a resources allocation scheme providing deterministic service guarantees across shared and switched Demand Priority networks. The chapter consisted of three logical parts. In the first, we defined the packet scheduling process and derived the admission control conditions. In the second part, we described the Time Window algorithm and discussed our implementation. The third part contained a performance evaluation comparing analytical and experimental results received from the analysis and the implementation, respectively.

We have proved that by using the high priority access mechanism with admission control, the network can support deterministic service guarantees. This applies to multi-hub cascaded, half-duplex switched, and bridged network topologies. The important analytical results are Theorem 6.1 and Theorem 6.2. Experiments showed that Theorem 6.1 can accurately model the variable data throughput in Demand Priority networks when used with the topology specific network parameters derived in Chapter 5. Theorem 6.2 additionally enables us to compute an upper bound on the end-to-end packet delay which may be lower than the time frame. In our experiments, we found that the scheme offers excellent delay characteristics. Small delay bounds can be guaranteed by using admission control. In all experiments, we never observed a single service violation. All data packets monitored for the test applications were transmitted with a delay that was significantly smaller than the upper bound provided by the admission control.

The measurements results further confirmed our network model and justified the need for an accurate analysis of the Demand Priority per-packet overhead and the normal priority service interrupt time. Less accurate bounds for repeating hubs or connecting links would have had a large impact on the theoretical data throughput for high cascaded topologies since these topologies have many hubs and links in the data path. A high accuracy however ensures that the allocation system has enough resources to manage, such that a sufficient number of real-time flows can be admitted while also guaranteeing a certain resource share for the aggregated best effort traffic.

Using rate regulators within switching then enabled us to use Theorem 6.1 and Theorem 6.2 for the admission control within bridged networks. Rate regulation within switches, however, significantly increased the complexity and the implementation costs of the solution. Such a mechanism was nevertheless required in order to ensure a deterministic delay bound and a reasonable resource utilization. The use of static priorities without rate regulators would have been simpler, but had also resulted in a poor resource utilization for real-time flows traversing several segments. The main advantages of using our allocation scheme in an unbridged network are its simplicity and its low costs. Hubs do not have to support per-flow classification or per-flow buffering and have only buffer space for a single maximum size data packet. The multi-hub network may have a large size and extension, but deterministic service guarantees can still be provided. Assuming the current price differences between 100 Mbit/s hubs and bridges, shared 802.12 networks supporting quality of service seem to be a flexible and cost effective network solution for supporting applications with stringent time constraints. Bridges are required when the total network traffic exceeds the capacity of the shared system. Whether per-flow rate regulators however become implemented in the near future is questionable. Instead, it seems currently more likely that designers trade-off the complexity with the service assurance level and provide a Controlled Load service.

The simplicity of the scheduling policy and the consideration of worst-case conditions in the admission control further result in a low resource utilization, especially for low bitrate flows. We believe that this is acceptable since any unused resources are not wasted, but can immediately be used by the network for serving normal priority (best-effort) service requests. Furthermore, a statistical multiplexing gain between real-time flows from different nodes in the network can not be exploited since all high-priority traffic is rate regulated at the edge of the shared segment and not within hubs. Other drawbacks are the general costs for the link level reservation setup mechanism and for the packet classifier. These are however not specific to our solution, but will also occur in other multiservice networks.

It remains to emphasize that the allocation scheme does not require any changes to the 802.12 standard. When deployed, then only network nodes which use the high priority medium access mechanism need to be updated. Normal priority data sources do not have to take part in the resource allocation since their service can be suspended.

Chapter 7
An Approximation of the Controlled Load Service

The low assurance level of the Controlled Load service enables different tradeoffs in the design of the traffic control and traffic enforcement mechanisms required for this service. Firstly, this may be used to increase the resource utilization by extensively exploiting the statistical properties of the traffic. Considering the network characteristics discussed in Section 4.3, we can expect large statistical multiplexing gains to be achieved without a significant loss of the service quality, provided the network traffic is bursty and constraint by admission control. Secondly, simple service disciplines in the network can be used to provide the desired service quality. This allows low cost solutions for switches at the expense of service reliability. Due to interactions of flows in the network, schemes based on simple packet schedulers, e.g. Static Priorities in switches, may however exhibit a lower resource utilization or different delay characteristics than those providing the same service but isolating each single flow in the bridged network.

To built a Controlled Load service, various approaches could be pursued. The simplest is probably the Simple Sum approach discussed in Section 6.6. In [JSD97], this scheme was used to provide Controlled Load quality of service. The underlying service discipline was an approximation of Weighted Fair Queuing. Much research on admission control has been performed based on the concept of the *Effective Bandwidth* (or *Equivalent Capacity*). In [GAN91] this is defined as the amount of bandwidth required to achieve the quality of service desired for a class of flows multiplexed on a link. More precisely [Floy96]: it is the capacity $C(\varepsilon)$ such that the stationary arrival rate of the class (e.g. including all Controlled Load service flows) exceeds $C(\varepsilon)$ with a probability of at most ε. If the Effective Bandwidth can be derived, then admission control could for example be performed by computing $C(\varepsilon)$ for the sum of all already admitted flows plus the new flow, and comparing the result to the maximum bandwidth share B allocated for the class or the service. If the result is lower ($C(\varepsilon) < B$) then the new flow is admitted.

One approach to compute the Effective Bandwidth, or an approximation for it, is to choose a statistical source model for the data arrival process at a switch and to select appropriate values for the model parameters. Afterwards the effective bandwidth is derived based on ε and the model. This approach was for example used in: [GAN91], [KWC93], [AS94], [GKK95], [Floy96], [GiKe97], [DJM97]. The parameter selection may be based on parameters declared by the sources at reservation setup e.g. their token bucket parameters (δ, r), or parameters measured on-line in the network.

An alternative approach is based on the theory of Large Deviation [Weis95]. Instead of choosing a statistical source model, the authors of: [DLC+95], [CLL+95], [CLH+95], [VeSo97] estimate the large deviation rate function, which is directly related to the Effective Bandwidth. This uses load measurements of the arriving traffic within the switch. We will discuss these and other approaches more in detail later in Section 7.4.

In spite of the previous research on statistical service guarantees, we use a Simple Sum style approach that is based on an average rate allocation to provide Controlled Load type service guarantees in Demand Priority networks. We believe that probabilistic end-to-end service guarantees will be difficult to derive in bridged topologies consisting of shared medium segments. Furthermore, a Simple Sum approach enables us to use a static priority scheduler with only two priority levels in LAN switches. This is probably the simplest scheduler that can be used to enforce Controlled Load quality of service and will keep our LAN switches cost competitive.

In contrast, to provide statistical service guarantees e.g. based on the Effective Bandwidth concept, two problems have to be solved: (1) a statistical source model to characterize the traffic must be selected or developed, and (2) end-to-end probabilistic bounds need to be derived. Choosing an appropriate source model is typically difficult because existing applications exhibit a variety of traffic characteristics which will not conform to a single model. It is further impossible to predict the characteristics of all the applications that will be used in a future Integrated Services Packet Network. Even if the traffic at the entrance of the network can be accurately described, this does not imply that the traffic in the core of the network can be characterized. We believe that the latter task is particulary hard in bridged Demand Priority LANs because: (1) the medium access is shared, resulting in a variable medium access delay, (2) the data throughput in the network is variable and depends on the packet size distribution of the traffic, and (3) the use of a static priority scheduler in LAN switches enables large interactions between flows which may temporarily affect the performance in adjacent network segments.

Now, all of the schemes described above were basically designed for ATM networks and assume a network with switches interconnected by point-to-point links. Each ATM switch is typically modelled as a simple single server queue which is served with a constant service rate. All cells arriving at the input have a constant size. Most of the algorithms rely on these assumptions and can thus not easily be applied to Demand Priority networks. Some of them might however be modified to do so. The measurement based approach described in [DLC+95] for example, still holds when used with a variable service rate [OCon98], provided the service rate function is known. An approximation for this function could also be measured at the switch. Even though this algorithm seems to be feasible, it requires additional mechanisms in LAN switches which will increase the costs. Furthermore, these mechanisms are not yet implemented in existing switches and will probably also not be available in next generation products which will delay the deployment of the algorithm indefinitely. We thus focus on a simpler approach which is likely to be more cost competitive to pure bandwidth.

The rest of this chapter is organized as follows. In Section 7.1 we discuss the basic assumptions made in the computation and describe the packet scheduling process. (The details of the Controlled Load service specification were described earlier in Section 2.2.3.) Section 7.2 contains the admission control conditions. These check the bandwidth- and the buffer space conditions in the network. Then, in Section 7.3, we discuss the properties of the admission control and evaluate the packet delay and loss characteristics of the new service, as measured in three different test networks. Related work is discussed in Section 7.4. Section 7.5 then summarizes the results achieved in this chapter.

7.1 The Packet Scheduling Process

The admission control conditions were derived based on a number of observations which we will discuss in the following. During our initial experiments (see for example Figure 4.12 in Section 4.3.3), we found that the network maintains an almost constant average packet delay of the order of a few milliseconds[1] over a long load range. This means that existing delay sensitive applications with end-to-end delay budgets of around 100 - 150 ms as reported in Section 2.1.2, will see little difference between an empty (0 Mbit/s) and a moderately loaded (~60 Mbit/s) network segment in the data path. This can be exploited by the network to provide Controlled Load service.

In contrast to this, packet loss must be watched carefully. In our measurements, we observed that it may occur long before the application may be able to detect a change in the average delay. Furthermore, the Controlled Load service definition specifies a target packet loss rate close to the packet error rate of the transmission medium. This is extremely low in LAN's. The 802.12 standard [ISO95] (see Section 16.9.3 therein) specifies a bit error rate of less than 1 bit error in 10^8 bits for UTP cabling. In one experiment, our single hub test network using 200 m UTP cables served 10^{10} 1500 byte data packets without any packet corruption detected. In respect to packet loss, a Controlled Load service providing a packet loss rate close to this value could be viewed as equivalent to a Guaranteed service. For comparison, measurements in the existing Internet exhibited bit error rates of about: $4.5 \cdot 10^{-8}$ which corresponds to a corruption rate of one data packet in every 5000 [Paxs97 - Section 13.3].

Due to these constraints, we focus on controlling the packet loss rate rather than the average delay and attempt to provide a loss free packet delivery service as, we believe, is expected from a Controlled Load service in a Local Area Network. Note that no stringent service guarantees are provided by the Controlled Load service for any of its service parameters. Based on the observations in Section 4.3.3, we thus do not attempt to derive a bound for the average delay for each admitted flow since we expect this to be sufficiently low, provided there is no packet loss. Given the difficulties in accurately modelling the data sources and the Demand Priority network behaviour, it is an open question whether a calculus will be able to provide accurate upper bounds for the average delay that

1. For example, the average delay for the *OptiVision* application in Figure 4.12 (playing the MPEG encoded adventure movie *Jurassic Park)* only increases by 0.631 ms while the network load increases from ~1.3 Mbit/s (1 flow) to ~70 Mbit/s (54 OptiVision flows).

are useful in the admission control, especially when considering the small load-delay variations observed in Figure 4.12.

The Controlled Load service also uses the 802.12 high priority medium access mechanism. This assumes that the Guaranteed service is not implemented. The bridged network may include cascaded and half-duplex switched Demand Priority segments. Controlled Load (high priority) data traffic is only reshaped at the entrance of the bridged network. This is illustrated in Figure 7.1. Network entrance points are nodes with network layer functionality such as hosts, routers and gateways. The rate regulators required in these nodes are identical to those described for the Guaranteed service in Section 6.1.3. On each network segment, data packets are served according to the Demand Priority round-robin service policy. LAN switches have a static priority scheduler with two priority levels. This was chosen because static priority scheduling will be widely deployed in next generation LAN switches. All Controlled Load service traffic is aggregated into the high priority queue. On each network segment, it is isolated using the 802.12 high priority access mechanism. Best effort traffic is mapped to the lower priority level of the static priority scheduler and forwarded based on the 802.12 normal priority service.

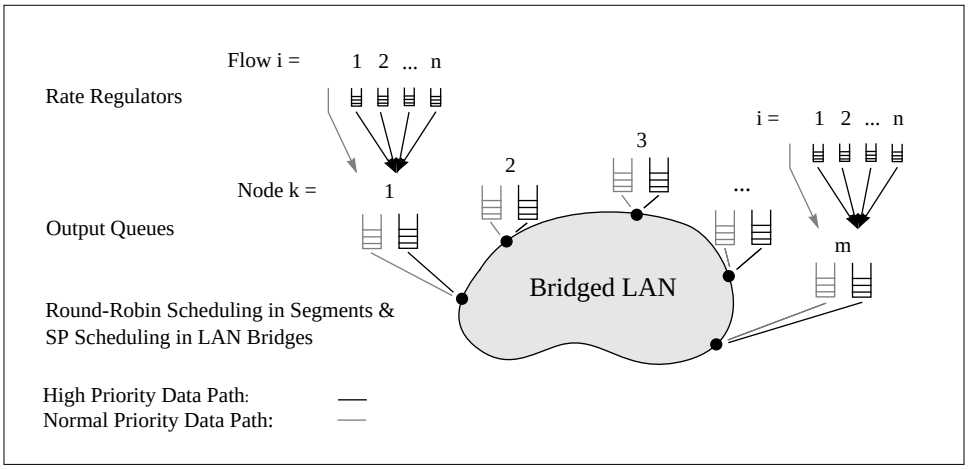


Figure 7.1: Traffic Reshaping Points for the Controlled Load Service.

The basic concept underlying our design is to control the amount of Controlled Load traffic that can enter the bridged network sufficiently conservative such that LAN switches in the core of the network do not lose data packets due to traffic distortions accumulated along the data path. The main difference to the scheduling model that was used for the deterministic service in the previous chapter is that Controlled Load flows are not reshaped in LAN switches. They may thus become more and more bursty as they travel across several segments within the network due to the interaction with other Controlled Load traffic. The degree of interaction depends on: (1) the Controlled Load (high priority) network utilization, (2) the burstiness of the traffic at the entrance of the network, (3) the length of the data path, and (4) the topology of the bridged LAN.

The high priority network utilization is controlled by admission control. This ensures that the Controlled Load service on each network segment, on average, consumes never more bandwidth than a pre-defined allocation limit. Admission control is also applied to limit the burstiness at the entrance of the network. Whenever the burstiness of the total traffic would exceed the buffer capacity in the network, then the new reservation request is rejected. Inside the network, the buffer space requirements of a flow grow monotonically along the data path. Fortunately, the data packets in the network are delivered based on a single data distribution tree. Bridged LANs may nevertheless have a meshed structure. The standard 802.1 Spanning Tree Protocol [ISO93] however ensures that there is always only one data distribution tree active. Feedback effects[1] as observed in wide area networks can thus not occur, which simplifies the analysis. It can further be assumed that the number of bridges between the source and the destination node in the LAN is limited. On average, we expect this to be of the order of two to five.

7.2 Admission Control

For admission control, we use a parameter based approach. All results are derived using the token bucket (δ, r) traffic characterisation introduced in Section 6.1.2. Resources are reserved on a per-network-segment basis (hop-by-hop) as carried out for the Guaranteed service. The admission control consists of a Bandwidth- and a Buffer Space Test. The Bandwidth Test proves that sufficient spare bandwidth is available such that *Stability* is maintained when the new flow is admitted. More precisely: assuming a network segment with N flows admitted, where each flow i obeys its Traffic Constraint Function: $b^i(\Delta t) \leq \delta^i + r^i \Delta t + r^i T$ at the entrance of the network, then Stability is given when:

$$\lim_{\Delta t \to \infty} \sum_{i \in N} b^i(\Delta t) - C_s \cdot \Delta t \; = \; -\infty \tag{7.1}$$

hold, where C_s denotes the network capacity available for serving data. It is computed later in Section 7.2.2. Equation 7.1 is basically identical to the definition in [Cruz91a]. Intuitively, the network is stable when: (1) the burst size δ^i is bounded for each flow i, and (2) the sum of the average rates of all admitted flows is smaller than the available network capacity C_s. Note that C_s is variable due to the Demand Priority overhead. To consider this dependency, the Bandwidth Test is derived from Theorem 6.1 and thus also based on a time frame concept. For each Controlled Load service flow, we however allocate network bandwidth corresponding to the average data rate r specified in the traffic characterisation (δ, r). This differs from the admission control applied for the Guaranteed service which was based on a peak-data-rate allocation. Note that a user may still request peak rate resources by choosing the parameters (δ, r) accordingly.

The Buffer Space Test checks that there is sufficient buffer space available such that none of the flow's data packets is dropped due to a queue overflow in the network. For this we first derive an

1. See for example [Zhan95] and the references therein for a discussion of feedback effects.

approximation of the Traffic Constraint Function of flow *i* after it traversed a single network segment. The corresponding analysis in Section 7.2.2 is complex due to the average rate allocation and the round-robin packet service policy used to enforce the QoS. The result however enables us to determine the buffer space. In the following, we continue with the Bandwidth Test. Furthermore, in the remaining of this chapter, we use the term *real-time* flow to denote a data flow using the Controlled Load service.

7.2.1 Bandwidth Test

Theorem 7.1 *Consider an 802.12 network segment with m nodes, where each node k has n real-time flows, which are already admitted. Assume a time frame of TF, a link speed of* C_l *and that the packet count for flow i on node k over the time interval TF is* $pcnt_k^i$. *Further let* D_{pp} *and* D_{it} *be the topology specific worst-case per-packet overhead and normal priority service interrupt time, respectively. Furthermore, assume that each real-time flow i on each node k has a bounded burst size* δ_k^i *and obeys its traffic characterisation* (δ_k^i, r_k^i) *at the entrance of the bridged network. A new Controlled Load flow* ν *with the traffic characterisation* (δ^ν, r^ν) *and a packet count* $pcnt^\nu$ *can be admitted such that Stability is maintained if:*

$$r^\nu < \frac{TF - D_{it} - \frac{1}{C_l}\sum_{k=1}^{m}\sum_{i=1}^{n} r_k^i \cdot TF - \sum_{k=1}^{m}\sum_{i=1}^{n} pcnt_k^i \cdot D_{pp} - pcnt^\nu \cdot D_{pp}}{TF / C_l} \tag{7.2}$$

Proof of Theorem 7.1

We first show, how Equation 7.2 was derived and then prove that stability is given when this equation applies. Assume that flow ν is admitted as a real-time flow on node *m +1* and that Theorem 6.1 (Equation 6.5 in Section 6.2.1) holds. For this case, we have:

$$\frac{1}{C_l} \cdot b^\nu(TF) + \frac{b^\nu(TF) \cdot D_{pp}}{P_{min}} \leq TF - D_{it} - \frac{1}{C_l}\sum_{k=1}^{m}\sum_{i=1}^{n} b_k^i(TF) - \sum_{k=1}^{m}\sum_{i=1}^{n} pcnt_k^i \cdot D_{pp} \tag{7.3}$$

The use of Theorem 6.1 ensures that the Demand Priority overhead is considered which ensures an accurate computation of the data throughput on the segment. If we now substitute the term: $b^\nu(TF)/P_{min}$ in Equation 7.2 by: $pcnt^\nu$ using Equation 6.3 in Section 6.1.3 with a packet size of: $p^\nu = P_{min}$, where P_{min} denotes the minimum network packet size, then we obtain by rearranging Equation 7.3:

$$\sum_{k=1}^{m+1}\sum_{i=1}^{n} b_k^i(TF) \leq C_l \cdot \left(TF - D_{it} - \sum_{k=1}^{m+1}\sum_{i=1}^{n} pcnt_k^i \cdot D_{pp} \right) \tag{7.4}$$

Using the Traffic Constraint Function: $b^i(\Delta t) \leq \delta^i + r^i \Delta t + r^i T$ for $\Delta t = TF$ and each flow i admitted then provides:

$$\sum_{k=1}^{m+1} \sum_{i=1}^{n} (\delta_k^i + r_k^i T_k) + \sum_{k=1}^{m+1} \sum_{i=1}^{n} r_k^i \cdot TF \quad \leq \quad C_l \cdot \left(TF - D_{it} - \sum_{k=1}^{m+1} \sum_{i=1}^{n} pcnt_k^i \cdot D_{pp} \right) \tag{7.5}$$

where T_k is the timer granularity used for the rate regulators at network node *k*. By removing the traffic bursts caused by all flows, we thus receive for an average rate allocation:

$$\sum_{k=1}^{m+1} \sum_{i=1}^{n} r_k^i \cdot TF \quad < \quad C_l \cdot \left(TF - D_{it} - \sum_{k=1}^{m+1} \sum_{i=1}^{n} pcnt_k^i \cdot D_{pp} \right) \tag{7.6}$$

Note that we also removed the equality in the equation because data are transmitted in packets and we thus have $\delta^i > 0$ for all flows *i*. If we now separate off the parameters for flow ν, then we get:

$$r^{\nu} \cdot TF + \sum_{k=1}^{m} \sum_{i=1}^{n} r_k^i \cdot TF \quad < \quad C_l \cdot \left(TF - D_{it} - \sum_{k=1}^{m} \sum_{i=1}^{n} pcnt_k^i \cdot D_{pp} - pcnt^{\nu} \cdot D_{pp} \right) \tag{7.7}$$

Theorem 7.1 follows from rearranging Equation 7.7. To show that this provides stability, we consider the sequence of time frames: $t \cdot TF$ where $t > 0, t \in N$. For $t \to \infty$, we then have:

$$\lim_{t \to \infty} \sum_{k=1}^{m+1} \sum_{i=1}^{n} r_k^i \cdot t \cdot TF + C_l \cdot D_{it} - C_l \cdot t \cdot \left(TF - \sum_{k=1}^{m+1} \sum_{i=1}^{n} pcnt_k^i \cdot D_{pp} \right) = -\infty \tag{7.8}$$

This follows from Equation 7.6. Now, if the maximum data burst ($\delta^i + r^i T$) that can be generated by each flow *i* on the segment is bounded such that: $\sum_{i \in n} \delta_i^i + r^i T < \infty$ for all nodes *k*, then Equation 7.8 also holds when each flow *i* sends an initial data burst into the network segment. For this case, we have:

$$\lim_{t \to \infty} \sum_{k=1}^{m+1} \sum_{i=1}^{n} \delta_k^i + r_k^i T_k + \sum_{k=1}^{m+1} \sum_{i=1}^{n} r_k^i \cdot t \cdot TF + C_l \cdot D_{it} - C_l \cdot t \cdot \left(TF - \sum_{k=1}^{m+1} \sum_{i=1}^{n} pcnt_k^i \cdot D_{pp} \right) = -\infty \tag{7.9}$$

Using the Traffic Constraint Function $b_k^i(t \cdot TF) \leq \delta_k^i + r_k^i (t \cdot TF) + r_k^i T$ in Equation 7.8 then provides:

$$\lim_{t \to \infty} \sum_{k=1}^{m+1} \sum_{i=1}^{n} b_k^i (t \cdot TF) + C_l \cdot D_{it} - C_l \cdot t \cdot \left(TF - \sum_{k=1}^{m+1} \sum_{i=1}^{n} pcnt_k^i \cdot D_{pp} \right) = -\infty \tag{7.10}$$

Equation 7.10 is the stability criterion. □

It differs from Equation 7.1 by: (1) the interrupt time D_{it} that is required once to pre-empt the normal priority service, and (2) the link speed C_l which combined with the Demand Priority per-packet overhead, reflect the network capacity C_s that is available for serving data. The total per-packet overhead: $\sum_{k \in m+1} \sum_{l \in n} pcnt_k^i \cdot D_{pp}$ must however be considered for each time frame *TF*.

The most significant difference in comparison to Theorem 6.1 is that the time frame *TF* in Theorem 7.1 is no longer an upper bound on the delay for the Controlled Load flows admitted. This is caused by the average data rate allocation. The time frame concept is however required because it enables us to bind the packet count for each flow and thus to find a bound for the total per-packet overhead to be considered for all real-time flows.

The difference between the data throughput actually available on the network and the bandwidth computed with Theorem 7.1 depends on the results used for the per-packet overhead and the interrupt time in the computation. For both parameters, average results could potentially be used since the Controlled Load service does not have to cover worst-case conditions. General results for the average delay are however difficult to determine so that we decided to reuse the upper bounds: D_{pp} and D_{it} derived in Chapter 5. Furthermore, we assume the use of the Time Window algorithm as described in Section 6.3 for estimating the packet count $pcnt^i$ for each real-time flow *i* in the network. The conservative nature of this algorithm and the use of the worst-case bounds for the Demand Priority overhead led to pessimistic results for the available bandwidth used in the admission control. This was intended because the interaction of real-time flows in the core of the bridged network can only be controlled implicitly by restricting the overall resource utilization and burstiness (instead of relying on the packet scheduler in LAN switches). The spare capacity however ensures that packet backlogs in queues are cleared quicker which decreases the risk of packet loss. In end-systems, the Controlled Load service thus exploits the same control mechanisms as used for the Guaranteed service. This reduced our implementation effort because additionally mechanisms did not have to be implemented.

It remains to remark that for partitioning the network bandwidth, the same method as used for the deterministic service case can also be applied for the Controlled Load service. Since this was described in Section 6.2.5, it is thus omitted here. When partitioning the resources for the Controlled Load service, the network administrator should however be conservative because Theorem 7.1 only accounts for the average data rates. In the event that Controlled Load flows pass large data bursts into the network, the 802.12 normal priority service used for Best-Effort data may temporarily receive a much lower bandwidth share than specified in the admission control.

7.2.2 Deriving the Output Traffic Constraint Function

In this section, we derive an approximation for the Traffic Constraint Function: $b^i(t)$ of flow *i* after it traversed a network segment. This is based on the analysis technique proposed by Cruz in [Cruz91a]. The result can recursively be applied to determine $b^i(t)$ on each segment along the data path of flow *i*.

The Network Model

For the analysis, we use the network model illustrated in Figure 7.2. Assumed is a shared network with *m* active nodes as the most general case. This is shown in the left part of the picture (a). On each node *k* in the network, we assume *n* real-time flows, each of which passes data traffic according to its Traffic Constraint Function $b_k^i(t)$ into the high priority output queue on *k*. The normal priority queue and the corresponding data path are omitted in the picture. We analyse the traffic constraint function of flow $i = 1$ on node *k = 1* which we denote with *FLOW 1*. All data packets of this flow are forwarded from node *k* = 1 to a LAN switch denoted with *Switch 2*. Switch 2 does not send any Controlled Load traffic into the network and therefore does not belong to *m*. For ease of reference, we further assume that node *k* = 1 is also a LAN switch with the name *Switch 1*. The data traffic of all other flows in the network can be viewed as cross traffic distorting the traffic pattern of FLOW 1 selected for analysis.

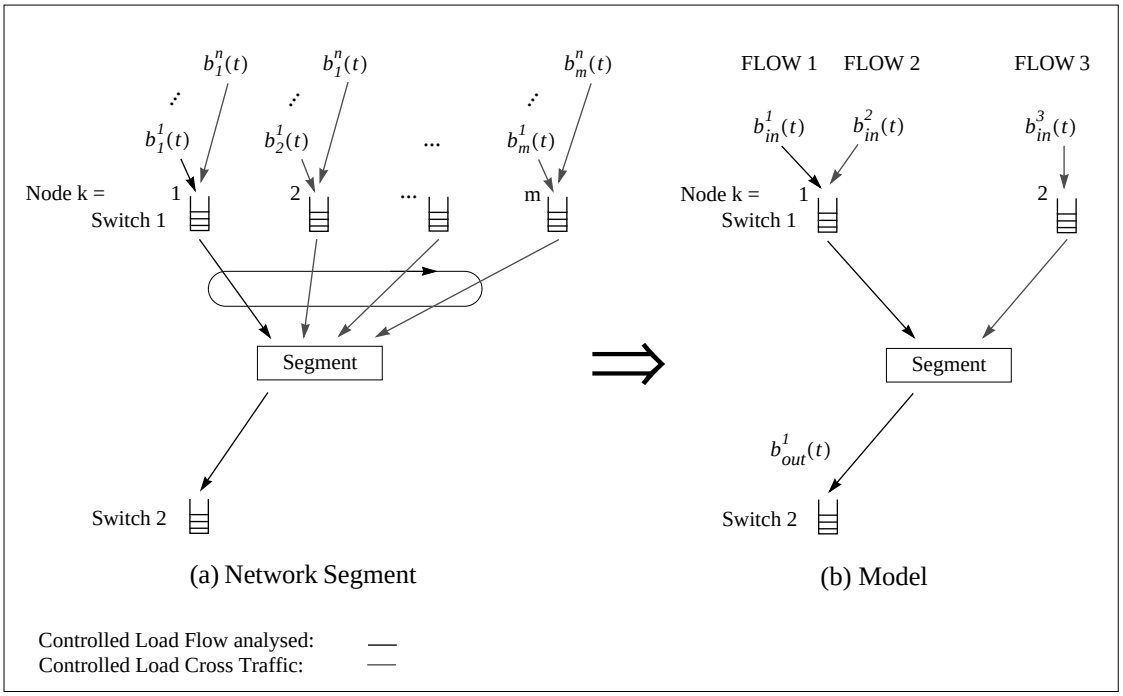


Figure 7.2: Network Model for Computing the Traffic Constraint Function of Flow *i*.

The right part of Figure 7.2 (b) shows the model and the notation, which we use to derive the results in this section. All real-time flows are mapped into a shared network with two active nodes. This results from the general observation that the data traffic of FLOW 1 is distorted by two groups of flows: (1) other Controlled Load flows, with $i \neq 1$, forwarded through Switch 1 onto the analysed segment, and (2) the flows passed into the network by other nodes $k \neq 1$. The aggregated data traffic of the former group is denoted with *FLOW 2* and described by the Traffic Constraint Function $b_{in}^2(t)$. The second group of flows is named *FLOW 3* and upper bounded by: $b_{in}^3(t)$. The function: $b_{in}^1(t)$ represents the input traffic of FLOW 1. We call these functions the Input Traffic Constraint

Functions of the corresponding flows because they describe the data which enters the segment. More formally, we have the mapping:

$$\begin{aligned} b_{in}^{1}(t) &= b_{1}^{1}(t) \\ b_{in}^{2}(t) &= \sum_{i=2}^{n} b_{1}^{i}(t) \\ b_{in}^{3}(t) &= \sum_{k=2}^{m} \sum_{i=1}^{n} b_{k}^{i}(t) \end{aligned} \tag{7.11}$$

between the network representation (a) and the corresponding model (b) in Figure 7.2. This holds because traffic constraint functions can be added using the following method: for the composite $b_{12}(t)$ of the two traffic constraint functions: $b_1(t)$ and $b_2(t)$, where $b_1(t) \le \delta_1 + r_1 \cdot t$ and $b_2(t) \le \delta_2 + r_2 \cdot t$ we have: $b_{12}(t) = b_1(t) + b_2(t) \le \delta_1 + \delta_2 + (r_1 + r_2) \cdot t$. This result further implies: $\delta_{12} = \delta_1 + \delta_2$, and $r_{12} = r_1 + r_2$. Both follow from the definition of the Traffic Constraint Function in Section 6.1.2 and are straightforward to see.

In the following analysis, we further do not explicitly consider the timer granularity *T* in the Traffic Constraint Function as performed for the deterministic case. Instead we use: $b(t) \le \delta + rt$ which is basically identical to Equation 6.1 in Section 6.1.2. This is because: (1) the weaker service commitment of the Controlled Load service does not necessarily require the consideration of *T*, provided *T* is small as discussed in Section 6.5.5, and (2) the data traffic is only rate regulated at the entrance of the bridged network and not within LAN switches. The burst size δ of the flow at the entrance of the network could be viewed as the sum of its actual burst size, denoted here using: δ', and the burstiness caused by the finite timer granularity: $\delta = \delta' + rT$. A substitution using this equation would thus lead to appropriate results.

The function $b_{out}^{1}(t)$ in Figure 7.2 is the Output Traffic Constraint Function of FLOW 1. It describes the traffic pattern that arrives at Switch 2 and is passed into the output queue of the next segment in the data path. This assumes output buffered LAN switches. The goal of this section is thus the derivation of the function $b_{out}^{1}(t)$. Note here that the cross traffic corresponding to: $b_{in}^{2}(t)$ and $b_{in}^{3}(t)$ may leave the segment at Switch 2 or at any other node in the network segment. This is however not illustrated in Figure 7.2.

The Calculus for $b_{out}^{1}(t)$

Theorem 7.2 *Consider an 802.12 segment with m network nodes and assume: (1) the network model in Figure 7.2, and (2) that the high priority traffic passed into the segment obeys the traffic constraint functions:* $b_{in}^{1}(t)$, $b_{in}^{2}(t)$ *and* $b_{in}^{3}(t)$ *according to the mapping given by Equation 7.11. Then let* D_{it}, P_{max} *and* $R^{1}_{min\_N1}$ *be the Normal Priority Service Interrupt Time, the Maximum Network Packet Size and the Minimum Service Rate of FLOW 1 located at node k = 1 (also denoted node N1), respectively. Furthermore let* Δ *and H denote two time variables, where:* $\Delta \ge 0$ *and*

$H \geq 0$. *If Theorem 7.1 applies (Stability) and the network has a total capacity of at least* C_s *available for serving data, then the output traffic of FLOW 1 is bounded by:*

$$b^1_{out}(t) \leq \max_{\Delta \geq 0}\left[b^1_{in}(t + \Delta + D_{it}) - R^1{}_{min\_N1} \cdot \left(\Delta - (m-1) \cdot \frac{P_{max}}{C_s} - H \right) \right] \tag{7.12}$$

Theorem 7.2 basically states that the traffic pattern of FLOW 1 is most distorted when the maximum amount of data defined by $b^1_{in}(t)$ is passed into the high priority output queue at Switch 1, but is afterwards only served with the minimum service rate $R^1{}_{min\_N1}$. This implicitly assumes the segment to be temporarily busy with serving data from: (1) FLOW 2 aggregating the real-time flows with $i \neq 1$ on node *k* = 1, and (2) FLOW 3 aggregating the real-time flows on network nodes with $k \neq 1$.

Note here that a Controlled Load service flow may temporarily be served with a rate significantly smaller than its allocated bandwidth. To illustrate this, assume for example an 802.12 network segment with *2* nodes, each of which passes a single flow into the segment. Let: $r_1 + r_2 < C_s$ but $r_1 + r_2 \approx C_s$, $r_1 = 3 \cdot r_2$ and $\delta_1 \gg 0$, $\delta_2 \gg 0$, where C_s denotes the available service rate and (δ_1, r_1), (δ_2, r_2) are the traffic characterisations of the two flows, respectively. Furthermore, assume that resources have been reserved on the segment and that both flows use the same fixed packet size for the data transmission. In this case, we find that although less bandwidth is reserved for the flow on *k* = 2, this flow may nevertheless temporarily consume half of the network capacity due to the average data rate allocation and the round-robin service policy. It can easily be shown that the longest interval for this effect is given by: $\Delta t \leq \delta_2 / ((C_s/2) - r_2)$, provided the flow on *k* = 2 obeys its traffic characterisation. During the time interval Δt, node *k* = 1 is however only served with a rate of: $C_s/2 < r_1$, which causes the data in the output queue on this node to grow. The data backlog on *k* = 1 is only reduced after δ_2 was cleared on node *k* = 2. This node may then only pass data according to its average rate r_2 into the network segment which leaves a capacity of: $C_s - r_1 - r_2 > 0$ to reduce the backlog on *k* = 1. For cascaded networks with many more network nodes, similar observations can be made.

Proof of Theorem 7.2

To prove the theorem, we basically follow the steps made in [Cruz91a - Section B] to prove the output traffic constraint function of flows traversing the General Multiplexer with Bounded Vacations, but apply them to our special case. First, we define the non-negative *Rate Function R* for each flow on the segment such that for arbitrary times $y \geq x$, $\int_x^y R(t)\, dt$ is the amount of data that is transmitted on the segment in the time interval [*x*, *y*]. *R(t)* can thus be viewed as the instantaneous data rate of the flow at time *t*. Furthermore, if a flow obeys its Traffic Constraint Function then the condition:

$$\int_x^y R(t)\, dt \leq b(y - x) \tag{7.13}$$

holds[1], where: $b(y-x) \le \delta + r \cdot (y-x)$ and $\delta > 0$, $r > 0$. This follows from the definition of the Traffic Constraint Function which, if enforced, limits the amount of data from this flow on the segment. Equation 7.13 also applies to flow aggregations, if $b(y-x)$ describes the traffic of a group and each flow within the group obeys its Traffic Constraint Function. In this case, the Rate Function *R(t)* represents the aggregated data traffic of these flows. In particular, we have the relations: $\int_x^y R_{in}^1(t)\ dt \le b_{in}^1(y-x)$, $\int_x^y R_{in}^2(t)\ dt \le b_{in}^2(y-x)$ and $\int_x^y R_{in}^3(t)\ dt \le b_{in}^3(y-x)$ for the Rate Functions of FLOW 1, FLOW 2 and FLOW 3 at the input to the segment in Figure 7.2, respectively. The equivalent functions can be defined for the traffic output: $\int_x^y R_{out}^1(t)\ dt \le b_{out}^1(y-x)$, $\int_x^y R_{out}^2(t)\ dt \le b_{out}^2(y-x)$ and $\int_x^y R_{out}^3(t)\ dt \le b_{out}^3(y-x)$. We call these results the Input- and Output Rate Functions according to their corresponding Traffic Constraint Function.

To prove Theorem 7.2, it is thus sufficient to show that the amount of data from FLOW 1 on the segment ($\int_x^y R_{out}^1\ dt$) is bounded by the Traffic Constraint Function $b_{out}^1(t)$ for all time intervals *t = y - x*. More precisely we have to show that:

$$\int_x^y R_{out}^1(t)\ dt \ \le \ \max_{\Delta \ge 0}\left[b_{in}^1(y-x+\Delta+D_{it}) - R^1{}_{min\_N1} \cdot \left(\Delta - (m-1)\cdot\frac{P_{max}}{C_s} - H\right)\right] \tag{7.14}$$

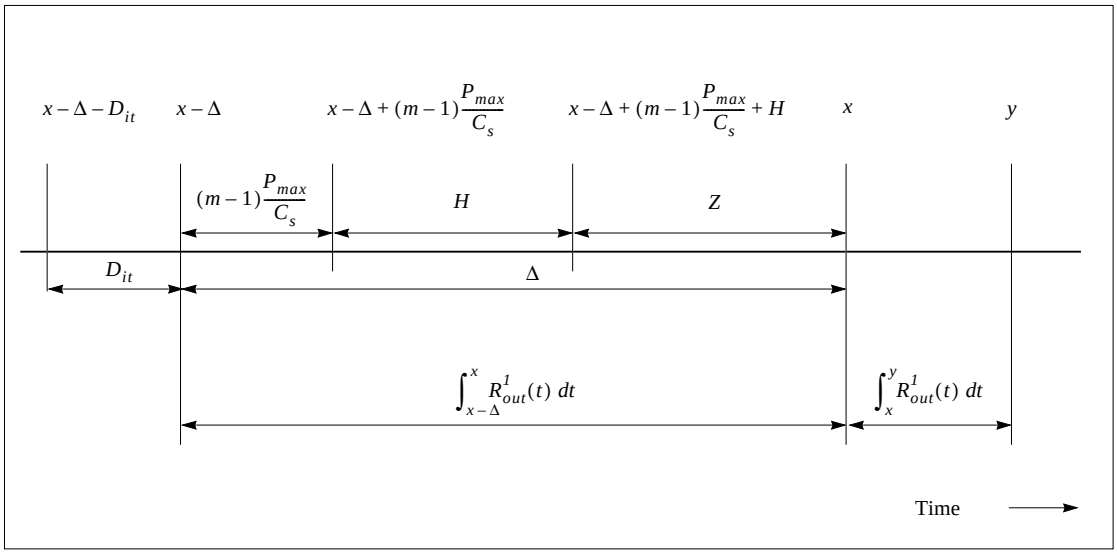


Figure 7.3: Timing Constraints for the Proof of Theorem 7.2.

Beside *x* and *y*, we further define the following time parameters whose relationships are illustrated in Figure 7.3.

$x-\Delta$: let Δ be such that at time: $t = x-\Delta$, the normal priority service is pre-empted and the network starts serving high priority data. This first however only concerns network nodes with

1. The definition of the Rate Function *R(t)* and Equation 7.13 are identical to the definitions in [Cruz91a].

$k \neq 1$ since we assume that node $k = 1$ is served last in the round-robin service sequence carried out by the network. In the worst case, node k is thus not served within: $(m-1) \cdot P_{max}/C_s$ time units. More formally we have:
$x - \Delta = inf\{t\colon (x - \Delta \leq t < x - \Delta + (m-1) \cdot P_{max}/C_s), R^1_{out}(t) = 0, R^2_{out}(t) = 0, R^3_{out}(t) > 0\}$.

$x - \Delta - D_{it}$: denotes the time when the first high priority data packet arrives at any of the nodes $k \neq 1$ on the network segment analysed. The time interval $[x - \Delta - D_{it}, x - \Delta)$ thus corresponds to the time required to pre-empt the normal priority service. This assumes that only normal priority data packets were served at time: $t \leq x - \Delta - D_{it}$. Formally, this is described by: $x - \Delta - D_{it} = inf\{t\colon (x - \Delta - D_{it} \leq t < x - \Delta), R^1_{out}(t) = 0, R^2_{out}(t) = 0, R^3_{out}(t) = 0\}$.

$x - \Delta + (m-1) \cdot P_{max}/C_s$: is the time when the network starts to serve data from node $k = 1$ beside serving data from nodes with $k > 1$. In our model, we assume that for H time units, only data from FLOW 2 aggregating the real-time flows: $i \in n$, $i \neq 1$ on node $k = 1$ are served. We thus have a Rate Function of $R^1_{out}(t) = 0$ as long as: $t < x - \Delta + (m-1) \cdot P_{max}/C_s + H$. This results in: $x - \Delta + (m-1) \cdot P_{max}/C_s = inf\{t$:
$(x - \Delta + (m-1) \cdot P_{max}/C_s \leq t < x - \Delta + (m-1) \cdot P_{max}/C_s + H)$
$R^1_{out}(t) = 0, R^2_{out}(t) > 0, R^3_{out}(t) > 0\}$.

$x - \Delta + (m-1) \cdot P_{max}/C_s + H$: denotes the time when the network starts serving data packets from FLOW 1. This is performed with the rate: $R^1{}_{min\_N1}$ which denotes the minimum service rate corresponding to FLOW 1. Since $R^1{}_{min\_N1}$ may however be smaller than the average rate ($R^1{}_{min\_N1} < r^1_1$), Switch 1 might nevertheless receive more data for the flow than it can forward on to the output segment. In this case, the buffer space used by FLOW 1 at Switch 1 is still growing despite of the service it receives from the network. More precisely, we have:
$x - \Delta + (m-1) \cdot P_{max}/C_s + H = inf\{t\colon (x - \Delta + (m-1) \cdot P_{max}/C_s + H \leq t < x),$
$R^1_{out}(t) = R^1{}_{min\_N1}, R^2_{out}(t) > 0, R^3_{out}(t) > 0\}$.

The parameter x thus corresponds to the time when the data backlog of FLOW 1 stops growing and the maximum amount of data from FLOW 1 is queued in the high priority output queue at Switch 1.

Now, from Figure 7.3, we have for the data output of FLOW 1 in the time interval [x, y]:

$$\int_x^y R^1_{out}(t)\,dt = \int_{x-\Delta}^{y} R^1_{out}(t)\,dt - \int_{x-\Delta}^{x} R^1_{out}(t)\,dt \tag{7.15}$$

Note the different intervals ($[y, x - \Delta]$, $[x, x - \Delta]$) used in the integrals on the right side of Equation 7.15. Figure 7.3 further provides for the data output in $[x, x - \Delta]$:

$$\int_{x-\Delta}^{x} R^1_{out}(t)\,dt = \int_{x-\Delta}^{x-\Delta+(m-1)\cdot P_{max}/C_s} R^1_{out}(t)\,dt + \int_{x-\Delta+(m-1)\cdot P_{max}/C_s}^{x-\Delta+(m-1)\cdot P_{max}/C_s+H} R^1_{out}(t)\,dt + \int_{x-\Delta+(m-1)\cdot P_{max}/C_s+H}^{x} R^1_{out}(t)\,dt \tag{7.16}$$

If we next consider that (1): $R^1_{out}(t) = 0$ for all times: $t < x - \Delta + (m-1) \cdot P_{max}/C_s + H$, and (2) $R^1_{out}(t) = R^1{}_{min\_N1}$ for all: $t \geq x - \Delta + (m-1) \cdot P_{max}/C_s + H$, then we have from Equation 7.16 by solving the integral:

$$\int_{x-\Delta}^{x} R^1_{out}(t)\, dt = 0 + 0 + R^1{}_{min\_N1} \cdot \left(\Delta - (m-1) \cdot \frac{P_{max}}{C_s} - H\right) \tag{7.17}$$

This follows from the definitions made for the time intervals in Figure 7.3 and from the fact that $R^1{}_{min\_N1}$ is a constant. To determine a bound for the missing term: $\int_{x-\Delta}^{y} R^1_{out}(t)\, dt$ in Equation 7.15, we look at the amount of data traffic from FLOW 1 which may leave the high priority output queue at Switch 1 within the time interval: $[x-\Delta, t]$, where $t \geq x - \Delta$. From the definition of $x - \Delta$ and $x - \Delta - D_{it}$ we obtain for this case:

$$\int_{x-\Delta}^{t} R^1_{out}(t)\, dt \leq \int_{x-\Delta-D_{it}}^{t} R^1_{in}(t)\, dt \tag{7.18}$$

by using the Input Rate Function of the flow. Intuitively, the output data rate of FLOW 1 on the segment is upper bounded by the rate at which the corresponding data arrive from the previous segment (Switch 1 in Figure 7.2). Switch 1 may however not be able to forward any high priority data for D_{it} time units after it received and processed the first high priority data packet because D_{it} is the time required to interrupt the normal priority service. Any data received for FLOW 1 within the interrupt time must thus be queued. If we now use Equation 7.13 in Equation 7.18 with the interval $[x-\Delta, y]$ then we get:

$$\int_{x-\Delta}^{y} R^1_{out}(t)\, dt \leq b^1_{in}(y - x + \Delta + D_{it}) \tag{7.19}$$

Using Equation 7.17 and Equation 7.19 in Equation 7.15 and considering the case that at the beginning of the time interval [*x*, *y*], Switch 1 must hold the maximum of data for FLOW 1, then we receive for the maximum data output:

$$\int_{x}^{y} R^1_{out}(t)\, dt \leq \max_{\Delta \geq 0}\left[b^1_{in}(y - x + \Delta + D_{it}) - R^1{}_{min\_N1} \cdot \left(\Delta - (m-1) \cdot \frac{P_{max}}{C_s} - H\right)\right] \tag{7.20}$$

This is Equation 7.14, which completes the proof of Theorem 7.2. The Output Traffic Constraint Function for flow (1, 1) follows for: $t = y - x$. □

Theorem 7.2 can however only be used for admission control when the results for the parameters: C_s, $R^1{}_{min}$, Δ and H have been computed. This is carried out in the following, before we provide an example for a flow traversing a bridged LAN.

Computing C_s

The service rate C_s can be derived from Theorem 6.1 when this theorem is applied for a single flow, where: $b(TF) = C_s \cdot TF$ according to the average rate allocation. If we additionally use a fixed packet size *p* instead of P_{min} in the computation, with $P_{min} \leq p \leq P_{max}$, then we receive from Equation 6.5 in Section 6.2.1:

$$C_s = \frac{TF - D_{it}}{TF \cdot \left(\frac{1}{C_l} + \frac{D_{pp}}{p}\right)} \tag{7.21}$$

The equality in Equation 7.21 follows from the fact that C_s is the maximum computed throughput available for the packet size *p*. The same result can also be derived using Theorem 7.1 since Equation 7.2 was also obtained from Theorem 6.1. In selecting an appropriate value for *p*, several different strategies can be applied. First, *p* could be set to the average packet size used by all flows admitted for the Controlled Load service. This can be determined from measurements in the network, or by heuristics if, for example, the Controlled Load data traffic is dominated by a single application type with a characteristic and well known packet size distribution. Alternatively, we can use the Minimum Average Packet Size $P_{MIN\_AVE\_S}$ as applied for the Guaranteed service. $P_{MIN\_AVE\_S}$ was defined in Equation 6.3 in Section 6.2.1 and describes the minimum packet size of *all admitted flows* on the segment averaged over the time frame *TF*. In general, any value between P_{min} and the average packet size can be appropriate. The selected value determines the conservativeness of the admission control but also affects the performance parameters such as the high priority resource utilization which will be lower when smaller values are used. In the experiments described later in Section 7.3.3, Section 7.3.4 and Section 7.3.5 for example, we used Equation 7.21 with $p = P_{MIN\_AVE\_S}$, mainly because the Time Window algorithm estimating the packet counts for all real-time flows was already implemented.

Computing $R^1_{min\_N1}$

To be able to compute the minimum service rate $R^1_{min\_N1}$ available for FLOW 1 in Figure 7.2, we first have to determine the minimum service rate of the corresponding node $k = 1$. For an arbitrary node *k* with $k \in m$, we use the symbol: R_{MIN_Nk} to denote the minimum service rate received by the node from the network. We further have: (1) $R_{MIN\_Nk} = \sum_{i=1}^{n} R^i_{min\_Nk}$, where $R^i_{min\_Nk}$ is the minimum service rate of flow *i* on *k*, and: (2) $0 < R_{MIN\_Nk} \leq C_s$, where C_s is the service rate on the segment.

The minimum bandwidth share for each node *k* is enforced by the round-robin service policy. It thus mainly depends on: (1) the number of nodes *m* with real-time flows on the segment, and (2) the packet sizes used by these flows. The parameter *m* is known since resources are reserved using admission control. Accurate results for R_{MIN_Nk} may however be difficult to determine when applications use variable sized packets. Making worst case assumptions in the computation may nevertheless enable us to find a bound. The result may however be overly pessimistic. Fortunately, the

Controlled Load service does not have to provide deterministic service guarantees, so that average results or pessimistic approximations of the network service can be considered. Equations 7.22 - 7.25 show different conditions that may be used to compute R_{MIN_Nk}. Each of them has a different degree of conservativeness.

$$R_{MIN\_Nk} = \left(\frac{P_{min}}{P_{min} + (m-1)P_{max}}\right) \cdot C_s \tag{7.22}$$

$$R_{MIN\_Nk} = \left(\frac{P_{MIN\_AVE\_Nk}}{P_{MIN\_AVE\_Nk} + (m-1)P_{max}}\right) \cdot C_s \tag{7.23}$$

$$R_{MIN\_Nk} = \left(\frac{P_{MIN\_AVE\_Nk}}{P_{MIN\_AVE\_S}}\right) \cdot C_s \tag{7.24}$$

$$R_{MIN\_Nk} = \frac{C_s}{m} \tag{7.25}$$

Equation 7.22 is the most pessimistic equation. Assumed is the worst case that node *k* only sends minimum sized packets whereas all other (*m* - 1) nodes on the network segment use data packets of maximum size P_{max}. Equation 7.23 is more optimistic because, instead of P_{min}, it uses the minimum average packet size $P_{MIN\_AVE\_Nk}$ of all data packets send by node *k*, where:

$$P_{MIN\_AVE\_Nk} = \frac{\sum_{i=1}^{n} b_k^i(TF)}{\sum_{i=1}^{n} pcnt_k^i} \tag{7.26}$$

This follows from Equation 6.6 in Section 6.2.1 by considering that: $P_{MIN\_AVE\_S} = \sum_{k \in m} P_{MIN\_AVE\_Nk}$. Since for existing applications $P_{MIN\_AVE\_Nk}$ is typically larger than P_{min}, Equation 7.23 will provide a larger and on average more accurate result for the actual bandwidth share of node *k*. Equation 7.23 however still assumes that nodes other than *k* use data packets of maximum size. This is overcome by Equation 7.24 which considers the minimum average packet size for all nodes on the segment.

The simplest condition is however given in Equation 7.25. It assumes that all nodes *k* on the segment, on average, will receive the same bandwidth share due to the round-robin policy. This is obviously not the case when different nodes use different packet sizes. Equation 7.25 might nevertheless be sufficient assuming that: (1) the total service rate C_s is a lower bound on the actually available network capacity, and (2) a certain amount of resources is left unallocated for the Best Effort service. Both ensures that the network has non-reserved bandwidth which prevents real-time data packets from being dropped in the network. In the experiments described later in this chapter, we used Equation 7.25 for computing R_{MIN_Nk}. This is because in each experiment, network nodes always sent homogeneous real-time flows into the network. These flows either used: (1) fixed packet sizes, or (2) variable packet sizes but with an average identical packet size distribution.

The result for R_{MIN_Nk} enables us to determine the minimum service rate for a single flow such as FLOW 1 on node $k = 1$ in Figure 7.2. In the most general case in which we do not consider the details of the packet scheduler in Switch 1, FLOW 1 can always use the bandwidth left over by all other real-time flows entering the segment through Switch 1. The corresponding aggregated data traffic is described by FLOW 2 in Figure 7.2 and has the traffic constraint function: $b^2_{in}(t) \leq \delta^2 + r^2 \cdot t$. When we thus consider the average data rate of FLOW 2 in the computation, then we receive for the service rate $R^1{}_{min\_N1}$ of FLOW 1:

$$R^1{}_{min\_N1} = \begin{cases} R_{MIN\_N1} - r^2 & \text{if } R_{MIN\_N1} > r^2 \\ 0 & \text{if } R_{MIN\_N1} \leq r^2 \end{cases} \tag{7.27}$$

This is straightforward to see. More optimistic approaches might take: (1) the details of the packet service discipline within Switch 1, (2) the number of different input ports, and (3) the link speeds on the corresponding network segments into account. The derivation and discussion of solutions considering these constraints is omitted since we use Equation 7.27 in the following.

It remains to remark in this context that whenever average parameters, heuristics or approximations such as given by Equation 7.25 are used for computing R_{MIN_Nk}, $R^1{}_{min\_N1}$, or C_s then Theorem 7.2 may only provide a loose approximation for the Output Traffic Constraint Function $b^1_{out}(t)$ of FLOW 1. If the estimation was too optimistic, then this may result in packet loss due to insufficient buffer space reserved in LAN switches.

Computing Δ

From Figure 7.3, we have for the time parameter Δ:

$$\Delta = (m-1) \cdot \frac{P_{max}}{C_s} + H + Z \tag{7.28}$$

where: $m \geq 1$, $H \geq 0$, $Z \geq 0$, $P_{max} > 0$ and $C_s > 0$. The computation of the missing parameters *H* and *Z* is based on the input and output data rates of the flows: FLOW 1, FLOW 2 and FLOW 3 in Figure 7.2. An example including one diagram for each of these flows is shown in Figure 7.4. The y-axes denote the amount of data that: (1) arrived at the node where the flow enters the segment ($\int R_{in}(t)\ dt$), and (2) that was served by the network ($\int R_{out}(t)\ dt$). The x-axes show the time *t*. For each flow, the upper curve in the diagram thus represents the data arrival rate, whereas the lower curve describes the service rate. The difference between both curves corresponds to the amount of data in the high priority queue. This is also called the Data Backlog.

The input traffic ($\int R_{in}(t)\ dt$) into each network node is limited for each flow *i* by the Traffic Constraint Function: $b^i_{in}(t + D_{it}) \leq \delta^i + r^i \cdot D_{it} + r^i \cdot t$. The offset D_{it} needs to be considered since the time $t = 0$ in Figure 7.4 denotes the condition when the normal priority service is pre-empted and

the network starts serving high priority data. During the preceding interrupt time, data equivalent to a maximum of $r^i \cdot D_{it}$ could however have been passed into the high priority queue.

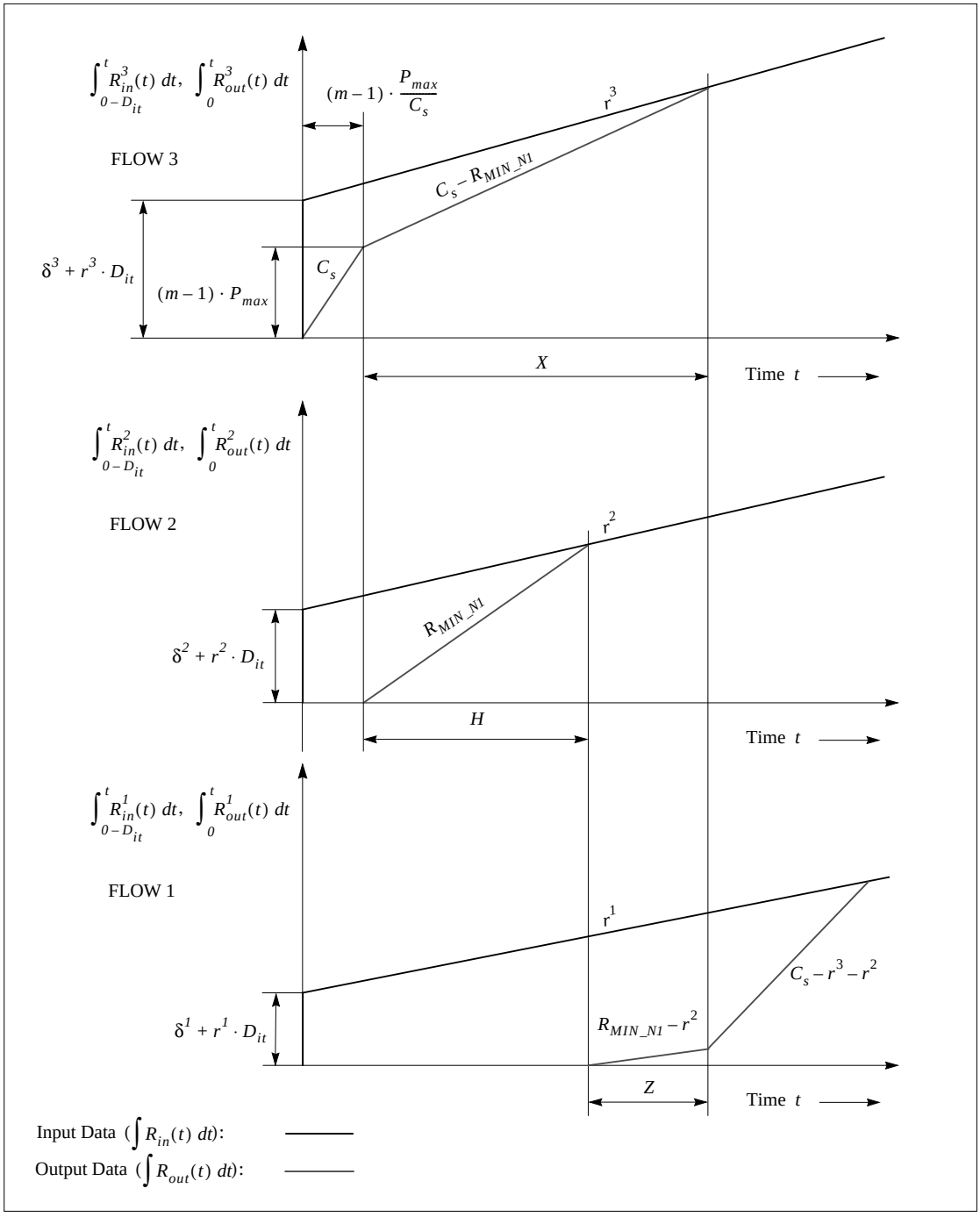


Figure 7.4: Example Data Arrival and Departure Function for FLOW 1, FLOW 2 and FLOW 3.

For the analysis of FLOW1, we assume that the network always serves data from FLOW 2 and FLOW 3 first. This is basically identical to the concept of the General Multiplexer analysed in [Cruz91a]. To compute the time parameters *H* and Z in Equation 7.28, we first define the time inter-

val *X*, where $X \geq 0$. The interval starts at time: $t = (m-1) \cdot P_{max}/C_s$ and ends when the data service curve of FLOW 3 reaches the data arrival curve: $\int R^3_{in}(t)\ dt = \int R^3_{out}(t)\ dt$. For the relation between *H*, *Z* and *X*, two general cases can be identified: (1) $0 \leq H < X$ where $0 < Z \leq X$, and (2) $0 < X \leq H$, where $Z = 0$. The former condition is illustrated in the example in Figure 7.4. It describes the case that high priority data packets from FLOW 1 are served by the network before: $\int R^3_{in}(t)\ dt = \int R^3_{out}(t)\ dt$. In the second case, the network service starts at the same time or after this event occurred. In the following, we compute results for the *X*, *H*, and *Z*. We start with the time parameter *X*.

Computing *X*

To compute the time interval *X*, we use the diagram of FLOW 3 in Figure 7.4. The data arrival is bounded by the corresponding Input Traffic Constraint Function. After a maximum data burst equivalent to: $\delta^3 + r^3 \cdot D_{it}$, which might for example be caused by several data packets arriving simultaneously, the maximum input rate of FLOW 3 is limited by r^3. The high priority service starts at *t* = 0. A maximum of *m* - 1 data packets of length P_{max} is transmitted first. This corresponds to the case that all nodes $k \in m, k \neq 1$ in the shared network have a packet to send. Note here that the service rate C_s also considers the per-packet overhead. At time: $t = (m-1) \cdot P_{max}/C_s$, the network service for FLOW 3 decreases to: $C_s - R_{MIN\_N1}$, where R_{MIN_N1} denotes the minimum service rate of node *k* = 1. For the computation of *X*, two cases can be identified based on whether: (1) the service rate is larger ($C_s - R_{MIN\_N1} > r^3$), or (2) equal or lower than the arrival rate ($C_s - R_{MIN\_N1} \leq r^3$). We now look at both of these cases separately.

When $C_s - R_{MIN\_N1} > r^3$ holds, we receive for the amount of data served by the network at time *t*:

$$\int_0^t R^3_{out}(t)\ dt \ \leq\ (C_s - R_{MIN\_N1}) \cdot t + R_{MIN\_N1} \cdot (m-1) \cdot \frac{P_{max}}{C_s} \tag{7.29}$$

where $(m-1) \cdot P_{max}/C_s \leq t \leq X$. This follows from Figure 7.4 when we consider the service rate of FLOW 3 in the interval *X* as a linear function of the form: $y = a \cdot t + c$, where $a = C_s - R_{MIN\_N1}$ and $c = R_{MIN\_N1} \cdot (m-1) \cdot P_{max}/C_s$. If we then use Equation 7.13 and 7.29 in condition: $\int_{0-D_{it}}^{t} R^3_{in}(t)\ dt = \int_0^t R^3_{out}(t)\ dt$, which defines the end of the interval, then we have:

$$b^3_{in}(t + D_{it}) \ =\ (C_s - R_{MIN\_N1}) \cdot t + R_{MIN\_N1} \cdot (m-1) \cdot \frac{P_{max}}{C_s} \tag{7.30}$$

Now, by replacing the traffic constraint function: $b^3_{in}(t + D_{it}) \ \leq \delta^3 + r^3 \cdot (t + D_{it})$ in Equation 7.30 and substituting: $t = (m-1) \cdot P_{max}/C_s + X$ in the result, we get:

$$\delta^3 + r^3 \cdot ((m-1) \cdot P_{max}/C_s + X + D_{it}) = (m-1) \cdot P_{max} + (C_s - R_{MIN\_N1}) \cdot X \tag{7.31}$$

Reordering Equation 7.31 then leads to:

$$X = \frac{\delta^3 + r^3 \cdot D_{it} + r^3 \cdot (m-1) \cdot P_{max} / C_s - (m-1) \cdot P_{max}}{(C_s - R_{MIN\_N1} - r^3)} \tag{7.32}$$

where $m \geq 1$, $\delta^3 \geq (m-1) \cdot P_{max}$ and $C_s - R_{MIN\_N1} > r^3$. The condition: $\delta^3 \geq (m-1) \cdot P_{max}$ ensures that: $X \geq 0$. It assumes that each of the (*m* - 1) nodes in FLOW 3 has a minimum burst size of P_{max}. The above result can be optimized when real-time flows use a smaller maximum packet size than P_{max}. The corresponding conditions are however omitted here.

The same technique can be used to compute *X* when: $C_s - R_{MIN\_N1} \leq r^3$. Figure 7.5 shows the data arrival and service curves for this case. In contrast to the corresponding diagram in Figure 7.4, the parameter *X* includes a time interval *G*, where $0 \leq G \leq X$, in which the data backlog of FLOW 3 does not decrease. For $C_s - R_{MIN\_N1} < r^3$ we find the backlog even growing. *G* starts at the same time as *X*: when the network begins serving node *k* = 1. The end of the interval is defined as the time when the data input of FLOW 1 and FLOW 2 on *k* = 1 is constrained by their average arrival rates r^1 and r^2. *G* thus corresponds to the time it takes the network to clear the maximum data backlog from node *k* = 1. The interval is bounded because whenever: $C_s - R_{MIN\_N1} \leq r^3$, then $r^1 + r^2 < R_{MIN\_N1}$. This follows directly from the condition: $r^1 + r^2 + r^3 < C_s$ which is enforced by Theorem 7.1.

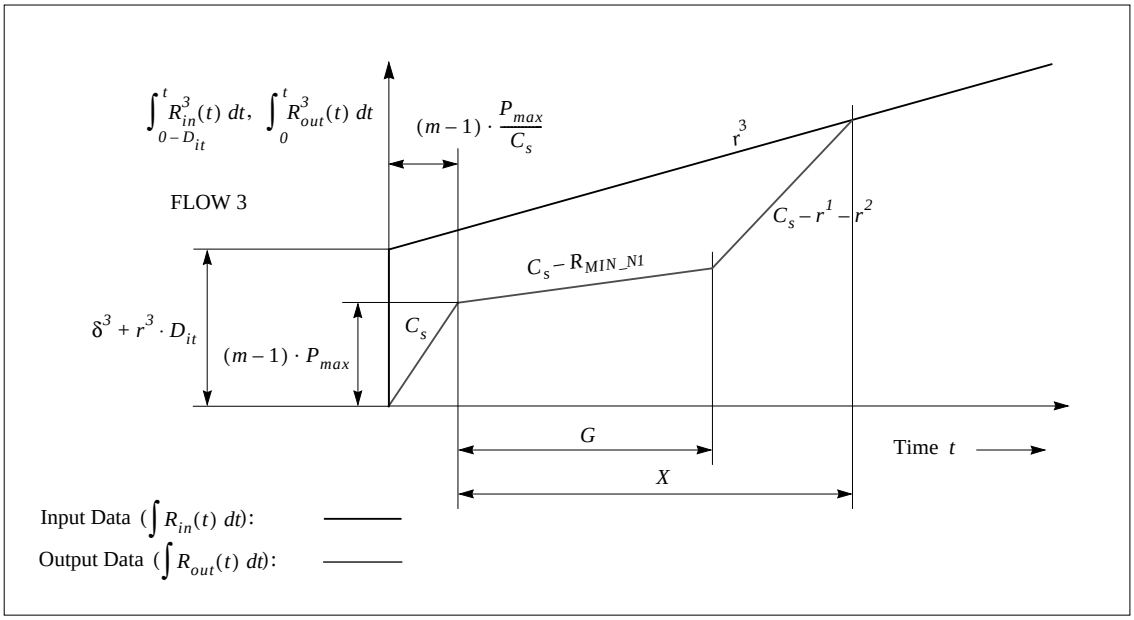


Figure 7.5: Data Arrival and Departure Function to compute Parameter *X* when: $C_s - R_{MIN\_N1} \leq r^3$.

By using the same approach as above for parameter *X*, we have for the end of the time interval *G*:

$$\int_{0-D_{it}}^{t} R_{in}^{1}(t)\,dt + \int_{0-D_{it}}^{t} R_{in}^{2}(t)\,dt = \int_{0}^{t} R_{out}^{1}(t)\,dt + \int_{0}^{t} R_{out}^{2}(t)\,dt \tag{7.33}$$

By considering the service rate of FLOW 1 and FLOW 2 as function of the form: $y = a \cdot t + c$, it follows from the definitions made for *G*, in particular: $r^1 + r^2 < R_{MIN\_N1}$, that: $a = R_{MIN\_N1}$ and $c = -R_{MIN\_N1} \cdot (m-1) \cdot P_{max}/C_s$. We thus have:

$$\int_{0}^{t} R_{out}^{1}(t)\,dt + \int_{0}^{t} R_{out}^{2}(t)\,dt \le R_{MIN\_N1} \cdot t - R_{MIN\_N1} \cdot (m-1) \cdot \frac{P_{max}}{C_s} \tag{7.34}$$

Using Equation 7.13 in Equation 7.34 then provides:

$$b_{in}^{1}(t + D_{it}) + b_{in}^{2}(t + D_{it}) = R_{MIN\_N1} \cdot t - R_{MIN\_N1} \cdot (m-1) \cdot \frac{P_{max}}{C_s} \tag{7.35}$$

If we now substitute $t = (m-1) \cdot P_{max}/C_s + G$ in Equation 7.35 and reorder the result, we receive:

$$G = \frac{\delta^1 + r^1 \cdot D_{it} + r^1 \cdot (m-1) \cdot P_{max}/C_s + \delta^2 + r^2 \cdot D_{it} + r^2 \cdot (m-1) \cdot P_{max}/C_s}{R_{MIN\_N1} - r^1 - r^2} \tag{7.36}$$

The result for parameter *G* now enables us to compute the service curve in the interval $G \le t \le X$. From this we can obtain the length of parameter *X*. The amount of data served from FLOW 3 for any *t* within $G \le t \le X$ is given by: $\int_{0}^{t} R_{out}^{3}(t)\,dt \le a \cdot t + c$, where $a = C_s - r^1 - r^2$. This follows from Figure 7.5. The parameter *c* is derived using Equation 7.29 with: $t = (m-1) \cdot P_{max}/C_s + G$. For this case we get: $(C_s - R_{MIN\_N1}) \cdot t + R_{MIN\_N1} \cdot (m-1) \cdot P_{max}/C_s = (C_s - r^1 - r^2) \cdot t + c$. Using the result for *c* then provides:

$$\begin{aligned} \int_{0}^{t} R_{out}^{3}(t)\,dt \le\ & (C_s - r^1 - r^2) \cdot t - \\ & ((C_s - r^1 - r^2) \cdot (m-1) \cdot P_{max}/C_s + (R_{MIN\_N1} - r^1 - r^2) \cdot G - ((m-1) \cdot P_{max})) \end{aligned} \tag{7.37}$$

If we now consider that, per definition, condition: $\int_{0-D_{it}}^{t} R_{in}^{3}(t)\,dt = \int_{0}^{t} R_{out}^{3}(t)\,dt$ holds at the end of the time interval *X*, then we receive by using Equation 7.37 and Equation 7.13:

$$\begin{aligned} b_{in}^{3}(t + D_{it}) \le\ & (C_s - r^1 - r^2) \cdot t - \\ & ((C_s - r^1 - r^2) \cdot (m-1) \cdot P_{max}/C_s + (R_{MIN\_N1} - r^1 - r^2) \cdot G - ((m-1) \cdot P_{max})) \end{aligned} \tag{7.38}$$

By replacing the traffic constraint function: $b^3_{in}(t + D_{it}) \le \delta^3 + r^3 \cdot (t + D_{it})$ in Equation 7.38 and substituting: (1) time *t* with: $t = (m-1) \cdot P_{max}/C_s + X$, and (2) parameter *G* in the result, using Equation 7.36, we receive for *X* after reordering:

$$X = \frac{\delta^1 + r^1 D_{it} + r^1(m-1)\frac{P_{max}}{C_s} + \delta^2 + r^2 D_{it} + r^2(m-1)\frac{P_{max}}{C_s}}{(C_s - r^1 - r^2 - r^3)} + \frac{\delta^3 + r^3 D_{it} + r^3(m-1)\frac{P_{max}}{C_s} - (m-1)P_{max}}{(C_s - r^1 - r^2 - r^3)} \tag{7.39}$$

where $m \ge 1$, $\delta^3 \ge (m-1) \cdot P_{max}$ and $C_s - R_{MIN\_N1} \le r^3$. The time: $t = (m-1) \cdot P_{max}/C_s + X$, when used with the result for *X* from Equation 7.39, corresponds to the time it takes the network to clear the worst-case data backlog on all nodes $k \in m$ on the segment. It is thus an upper bound on the delay, although the result will be large and therefore not necessarily useful.

Combining the results given by Equation 7.32 and 7.39 then provides for the time parameter *X*:

$$X = \begin{cases} \dfrac{\delta^3 + r^3 D_{it} + r^3(m-1)\frac{P_{max}}{C_s} - (m-1)P_{max}}{(C_s - R_{MIN\_N1} - r^3)} & \text{if } C_s - R_{MIN\_N1} > r^3 \\ \dfrac{\delta^1 + r^1 D_{it} + r^1(m-1)\frac{P_{max}}{C_s} + \delta^2 + r^2 D_{it} + r^2(m-1)\frac{P_{max}}{C_s}}{(C_s - r^1 - r^2 - r^3)} + \dfrac{\delta^3 + r^3 D_{it} + r^3(m-1)\frac{P_{max}}{C_s} - (m-1)P_{max}}{(C_s - r^1 - r^2 - r^3)} & \text{if } C_s - R_{MIN\_N1} \le r^3 \end{cases} \tag{7.40}$$

Computing *H*

The parameter *H* represents the time interval that is required to clear the maximum data backlog of FLOW 2 such that for any time $t \ge H$, the data input into the segment is constrained by the arrival rate r^2.

For the analysis, two specific cases can be identified based on whether *H* is smaller or larger then the parameter *X* computed in the previous section. An example for $H \le X$ is given in Figure 7.4. By using the same analysis approach as applied for *X*, we receive for the amount of data served by the network at time *t*, where: $(m-1) \cdot P_{max}/C_s \le t \le H$:

$$\int_0^t R^2_{out}(t)\, dt \le R_{MIN\_N1} \cdot t - R_{MIN\_N1} \cdot (m-1) \cdot \frac{P_{max}}{C_s} \tag{7.41}$$

This follows from the definitions for FLOW 2 illustrated in Figure 7.4. From Equation 7.13 and condition: $\int_{0-D_{it}}^{t} R_{in}^2(t)\,dt = \int_{0}^{t} R_{out}^2(t)\,dt$ defining the end of the interval *H*, we have:

$$b_{in}^2(t + D_{it}) = R_{MIN\_N1} \cdot t - R_{MIN\_N1} \cdot (m-1) \cdot \frac{P_{max}}{C_s} \tag{7.42}$$

Replacing the traffic constraint function: $b_{in}^2(t + D_{it})$ of FLOW 2 in Equation 7.42 and substituting the time *t* in the result where: $t = (m-1) \cdot P_{max}/C_s + H$, then leads to:

$$H = \frac{\delta^2 + r^2 \cdot D_{it} + r^2 \cdot (m-1) \cdot P_{max}/C_s}{R_{MIN\_N1} - r^2} \tag{7.43}$$

where $R_{MIN\_N1} > r^2$ and:

$$\frac{\delta^2 + r^2 \cdot D_{it} + r^2 \cdot (m-1) \cdot P_{max}/C_s}{R_{MIN\_N1} - r^2} \leq X \tag{7.44}$$

Condition: $R_{MIN\_N1} > r^2$ and Equation 7.44 ensure that Equation 7.43 provides a non-negative result. If one of the these conditions does not apply then there is a solution with $H > X$ and Equation 7.43 does not hold. It can however be shown that if: $R_{MIN\_N1} > r^2$ and $C_s - R_{MIN\_N1} \leq r^3$ (second condition in Equation 7.40) apply, then Equation 7.44 applies for arbitrary sets of valid flow parameters. In particular we need: $r \geq 0$, $\delta \geq P_{max}$ for all flows in the network. Further required are: $R_{MIN\_N1} > 0$ and $m \geq 1$. The proof of this is however omitted in this thesis since the future results do not depend on the relation between *H* and condition: $C_s - R_{MIN\_N1} \leq r^3$.

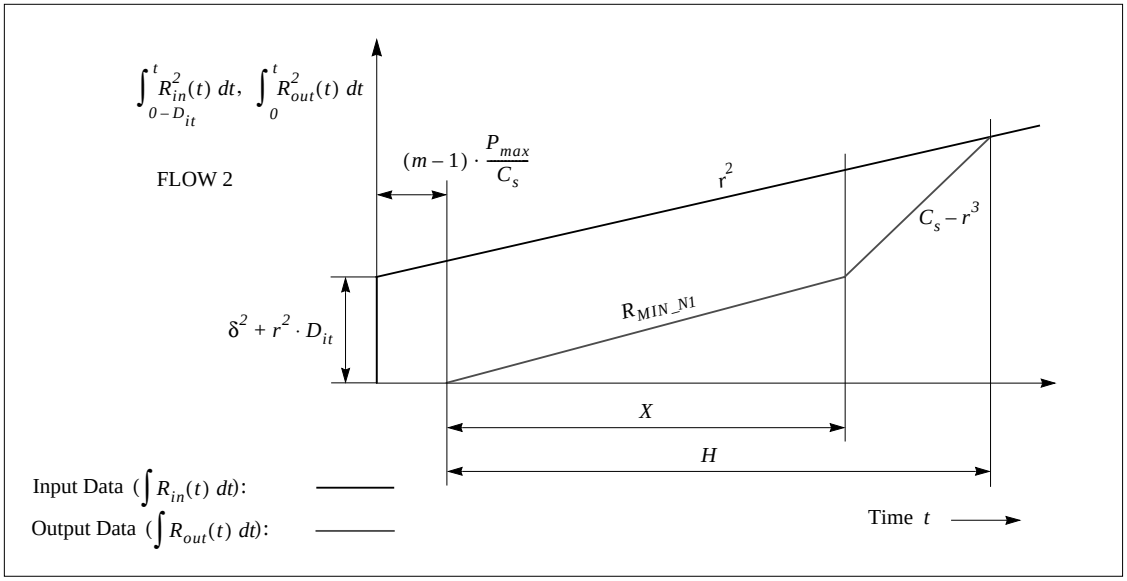


Figure 7.6: Data Arrival and Departure Function to compute the Time Interval *H*, where: $H > X$.

Instead, we continue with the computation of *H* for the case that either: (1) $R_{MIN\_N1} \le r^2$ applies, or (2) $R_{MIN\_N1} > r^2$ holds but the condition given in Equation 7.44 is not true. Figure 7.6 shows an example for case (2). We can observe that although the data backlog of FLOW 2 decreases during the time interval *X*, it is not cleared. In our model, this only occurs after the network processed the backlog of FLOW 3 and then served FLOW 2 with the new service rate: $C_s - r^3$ for further *H* - *X* time units.

By using the same approach as applied for computing the time parameter *X*, we have: $\int_0^t R_{out}^2(t)\,dt \le a \cdot t + c$ for the amount of data served from FLOW 2 for any time *t* within $X \le t \le H$. For parameter *a* in the linear function, we get: $a = C_s - r^3$ because the data input of FLOW 3 is constrained by the rate r^3. This can be observed in Figure 7.6. The parameter *c* is derived using Equation 7.41 which describes the data service that FLOW 2 receives during the time interval *X*. Combining both conditions leads to: $R_{MIN\_N1} \cdot t - R_{MIN\_N1} \cdot (m-1) \cdot P_{max}/C_s = (C_s - r^3) \cdot t + c$, where $t = (m-1) \cdot P_{max}/C_s + X$. Reordering then provides:

$$\int_0^t R_{out}^2(t)\,dt \le (C_s - r^3) \cdot t + r^3 \cdot (m-1) \cdot \frac{P_{max}}{C_s} - (C_s - r^3 - R_{MIN\_N1}) \cdot X - (m-1) \cdot P_{max} \qquad (7.45)$$

Now, from: $\int_{0-D_{it}}^{t} R_{in}^2(t)\,dt = \int_0^t R_{out}^2(t)\,dt$, Equation 7.13 and Equation 7.45, we get:

$$b_{in}^2(t + D_{it}) \le (C_s - r^3) \cdot t + r^3 \cdot (m-1) \cdot \frac{P_{max}}{C_s} - (C_s - r^3 - R_{MIN\_N1}) \cdot X - (m-1) \cdot P_{max} \qquad (7.46)$$

Substituting the Traffic Constraint Function ($b_{in}^2(t + D_{it})$) and reordering the result then provides with $t = (m-1) \cdot P_{max}/C_s + H$:

$$H = \frac{\delta^2 + r^2 \cdot D_{it} + r^2 \cdot (m-1) \cdot P_{max}/C_s + (C_s - R_{MIN\_N1} - r^3) \cdot X}{(C_s - r^3 - r^2)} \qquad (7.47)$$

Using Equation 7.43 and Equation 7.47, we finally have for the time interval *H*:

$$H = \begin{cases} \dfrac{\delta^2 + r^2 D_{it} + r^2 (m-1) P_{max}/C_s}{R_{MIN\_N1} - r^2} & \text{if } R_{MIN\_N1} > r^2 \text{ and } \dfrac{\delta^2 + r^2 D_{it} + r^2 (m-1) P_{max}/C_s}{R_{MIN\_N1} - r^2} \le X \\ \dfrac{\delta^2 + r^2 D_{it} + r^2 (m-1) P_{max}/C_s + (C_s - R_{MIN\_N1} - r^3) X}{(C_s - r^3 - r^2)} & \text{otherwise} \end{cases} \qquad (7.48)$$

Computing *Z*

The last parameter to be computed is the time interval *Z*. It denotes the interval in which FLOW 1 is served by the network but with a service rate smaller than its arrival rate ($R^1_{out}(t) < R^1_{in}(t)$). In spite of the network service, the data backlog of FLOW 1 at Switch 1 is thus still increasing. The interval ends when the high priority output queue at Switch 1 holds the maximum amount of data from FLOW 1 such that for any time $t \geq x$ in Figure 7.3 the data backlog does not increase any more. In contrast to this, the interval is zero when the maximum backlog is reached at any time $t \leq x - \Delta + (m-1) \cdot P_{max}/C_s + H$.

To compute *Z*, we consider two conditions: (1) the relation between the time parameters *H* and *X*, and (2) the minimum service rate R_{MIN_N1} for node *k* = 1. *Z* can only be positive when *H* < *X*. This follows from the definitions of *X*, *H* and *Z*. An example is given in Figure 7.4. The backlog of FLOW 2 however only increases when additionally the condition: $R_{MIN\_N1} < r^1 + r^2$ applies because only in that case we have: $R^1_{out}(t) < R^1_{in}(t)$ and thus: $0 < Z = X - H$. In contrast, we find that the amount of data buffered for FLOW 1 at the output of Switch 1 does not grow when $R_{MIN\_N1} \geq r^1 + r^2$ because for any time: $t \geq x - \Delta + (m-1) \cdot P_{max}/C_s + H$, the input rate of FLOW 1 and FLOW 2 is then constrained by r^1 and r^2, respectively. This results in: $R^1_{out}(t) \geq R^1_{in}(t)$, which avoids growth.

In the case that condition $H \geq X$ applies, we always have: $Z = 0$ regardless of the rate R_{MIN_N1}. This is because in our model, FLOW 1 does not receive service within the interval *H*. At any time *t* later than $t \geq x - \Delta + (m-1) \cdot P_{max}/C_s + H$ however, the flow is served with the rate: $C_s - r^3 - r^2$ which is larger than r^1 due to Theorem 7.1. More precisely, we receive for the parameter *Z* in Equation 7.28:

$$Z = \begin{cases} X - H & \text{if } H < X \text{ and } R_{MIN\_N1} < r^1 + r^2 \\ 0 & \text{otherwise} \end{cases} \tag{7.49}$$

Two Examples

In the remainder of this section, we compute and discuss two general examples for the Output Traffic Constraint Function $b^1_{out}(t)$. In both cases let: *m* = 2 (a half-duplex link), $\delta^1 \geq 0$, $r^1 > 0$, $\delta^2 \geq 0$, $r^2 > 0$, $\delta^3 \geq P_{max}$, $r^3 > 0$ and $D_{it} \geq 0$. We further demand that Theorem 7.1 applies.

In the first example, we compute $b^1_{out}(t)$ for the following case: (1) $C_s - R_{MIN\_N1} > r^3$ which determines the use of Equation 7.32 for computing the time interval *X*, (2) $R_{MIN\_N1} > r^2$ and $H \leq X$ according to Equation 7.43, and (3) $H < X$ and $R_{MIN\_N1} \geq r^1 + r^2$ which results in *Z* = 0. Now, from Theorem 7.2. we have for *m* = 2: $b^1_{out}(t) \leq b^1_{in}(t + \Delta + D_{it}) - R^1{}_{min\_N1} \cdot (\Delta - P_{max}/C_s - H)$. Furthermore, Equation 7.28 provides: $\Delta = P_{max}/C_s + H$ for $Z = 0$ and $m = 2$. Both then leads to: $b^1_{out}(t) \leq b^1_{in}(t + P_{max}/C_s + H + D_{it})$. Finally, replacing the Input Traffic Constraint Function with its parameters, where: $b^1_{in}(t) \leq \delta^1 + r^1 \cdot t$, provides, after reordering, for the Output Traffic Constraint Function of FLOW 1:

$$b^1_{out}(t) \le \delta^1 + r^1 D_{it} + r^1(P_{max}/C_s) + r^1\left(\frac{\delta^2 + r^2 D_{it} + r^2(P_{max}/C_s)}{R_{MIN\_N1} - r^2}\right) + r^1 t \tag{7.50}$$

Equation 7.50 is the typical result when sufficient spare capacity is available on the link such that: $C_s - R_{MIN\_N1} > r^3$ and $R_{MIN\_N1} \ge r^1 + r^2$. Note here that for a half-duplex link (m = 2), we have: $R_{MIN\_Nk} = C_s/2$ assuming the optimistic approach in Equation 7.25. Since $R_{MIN\_N1} \ge r^1 + r^2$, the data service for FLOW 1 and FLOW 2 on node k = 1 is independent from FLOW 3 on node k = 2. This can also be observed in the result in Equation 7.50 since $b^1_{out}(t)$ does not include any traffic parameters from FLOW 3. The term: $\delta^1 \cdot r^1 D_{it} + r^1(P_{max}/C_s)$ in Equation 7.50 describes the maximum data backlog in the output queue at Switch 1 just before the network starts serving node k = 1, whereas: $r^1 \cdot (\delta^2 + r^2 D_{it} + r^2(P_{max}/C_s))/(R_{MIN\_N1} - r^2))$ corresponds to the maximum amount of data received from FLOW 1 while the entire data backlog from FLOW 2 is served.

In the second example, we compute $b^1_{out}(t)$ based on the same assumptions as considered in the first example, except that we let: $H < X$ and $R_{MIN\_N1} < r^1 + r^2$, but $R_{MIN\_N1} > r^2$ This results in: $Z = X - H$ following Equation 7.49 and thus: $\Delta = P_{max}/C_s + X$ using Equation 7.28. From condition: $R_{MIN\_N1} > r^2$ and Equation 7.27, we have: $R^1{}_{min\_N1} = R_{MIN\_N1} - r^2$. Combining these results provides: $b^1_{out}(t) \le b^1_{in}(t + P_{max}/C_s + X + D_{it}) - (R_{MIN\_N1} - r^2) \cdot (X - H)$ for the Output Traffic Constraint Function. If we now substitute $b^1_{in}(t)$ and use Equation 7.32 and 7.43 in the result, then we receive after reordering:

$$\begin{aligned} b^1_{out}(t) \le \delta^1 + r^1 D_{it} + r^1(P_{max}/C_s) + r^1\left(\frac{\delta^3 + r^3 D_{it} + r^3(P_{max}/C_s) - P_{max}}{C_s - R_{MIN\_N1} - r^3}\right) - \\ (R_{MIN\_N1} - r^2) \cdot \left(\left(\frac{\delta^3 + r^3 D_{it} + r^3(P_{max}/C_s) - P_{max}}{C_s - R_{MIN\_N1} - r^3}\right) - \left(\frac{\delta^2 + r^2 D_{it} + r^2(P_{max}/C_s)}{R_{MIN\_N1} - r^2}\right)\right) + r^1 t \end{aligned} \tag{7.51}$$

where: $C_s - R_{MIN\_N1} > r^3$ and $R_{MIN\_N1} > r^2$. This is the Output Traffic Constraint Function of FLOW 1 in the example illustrated in Figure 7.4, assuming that m = 2. Similar components as discussed for the first example can also be identified in Equation 7.51. The first term: $\delta^1 \cdot r^1 D_{it} + r^1(P_{max}/C_s)$ is the maximum backlog at time: $t = P_{max}/C_s$. The second term: $r^1 \cdot (\delta^3 + r^3 D_{it} + r^3(P_{max}/C_s) - P_{max})/(C_s - R_{MIN\_N1} - r^3))$ describes the maximum amount of data received from FLOW 1 at Switch 1 while the entire data backlog of FLOW 3 is served. The service that node k = 1 receives from the network during this time interval is represented by: $(R_{MIN\_N1} - r^2) \cdot ((\delta^3 + r^3 D_{it} + r^3(P_{max}/C_s) - P_{max})/(C_s - R_{MIN\_N1} - r^3))$. The fourth large term in Equation 7.51 is: $(R_{MIN\_N1} - r^2) \cdot ((\delta^2 + r^2 D_{it} + r^2(P_{max}/C_s))/(R_{MIN\_N1} - r^2))$. It corresponds to the maximum data backlog from FLOW 2 and thus implicitly describes the maximum interaction between FLOW 1 and FLOW 2 on node k = 1.

It remains to remark that results similar to Equation 7.50 and 7.51 can also be found for cases using different assumptions than considered in the examples above. Furthermore, results can always be mapped into the form: $b^1_{out}(t) \leq \delta + r \cdot t$ such that they can then be used as Input Traffic Constraint Function for the next segment in the data path of the flow.

7.2.3 Buffer Space Test

Theorem 7.3 *Consider an 802.12 segment with m network nodes and assume: (1) the network model in Figure 7.2, and (2) that the high priority traffic passed into the segment obeys the traffic constraint functions:* $b^1_{in}(t)$, $b^2_{in}(t)$ *and* $b^3_{in}(t)$ *according to the mapping given by Equation 7.11. If Theorem 7.1 applies (Stability) and the output traffic of FLOW 1 is bounded by the corresponding Traffic Constraint Function* $b^1_{out}(t)$ *specified by Theorem 7.2, then the buffer space* sS^1 *required for FLOW 1 at the entrance to the network segment is bounded by:*

$$sS^1 \leq b^1_{out}(0) \tag{7.52}$$

This follows from the definitions made for the computation of the Output Traffic Constraint Function $b^1_{out}(t)$ in the previous section. The formal proof of Equation 7.52 thus is based on these assumptions. It further uses the same basic approach as applied in [Cruz91a - Section C] for the General Multiplexer.

Proof of Theorem 7.3

Define the parameters: m, x, Δ, D_{it}, H, Z, $R^1_{min\_N1}$, P_{max} and C_s as in the proof of Theorem 7.2 (see Figure 7.3). In particular, recall time x, which corresponds to the time when the amount of data (the maximum data backlog) hold for FLOW 1 in the high priority output queue at Switch 1 in Figure 7.2 stops growing such that for any $t \geq x$ the backlog does not increase any further. Now, the buffer space sS^1 required for FLOW 1 is equivalent to the maximum data backlog which occurs at time x. We thus have:

$$sS^1 = W^1(x) = \int_{x-\Delta-D_{it}}^{x} R^1_{in}(t)\,dt - \int_{x-\Delta}^{x} R^1_{out}(t)\,dt \tag{7.53}$$

where $W^1(x)$ denotes the data backlog of FLOW 1 at time x. Now, using Equation 7.13 and Equation 7.17 in Equation 7.53 provides:

$$sS^1 \leq b^1_{in}(\Delta + D_{it}) - R^1_{min\_N1} \cdot \left(\Delta - (m-1) \cdot \frac{P_{max}}{C_s} - H\right) \tag{7.54}$$

where the inequality follows from Equation 7.13. Equation 7.54 is identical to: $b^1_{out}(0)$. □

7.3 Performance Evaluation

After defining the scheduling process and the admission control conditions for the Controlled Load service, we now investigate the service properties enforced by these mechanisms. We first study the impact of the cross traffic characteristics on the buffer space requirements computed for a flow in shared and half-duplex switched network topologies. This is followed by a numerical example which shows how the buffer space grows along the data path in a multi-hop bridged network. We then look at the Admissible Region of a flow which is developed from the possibility that the user may select several sets of resource parameters for the reservation.

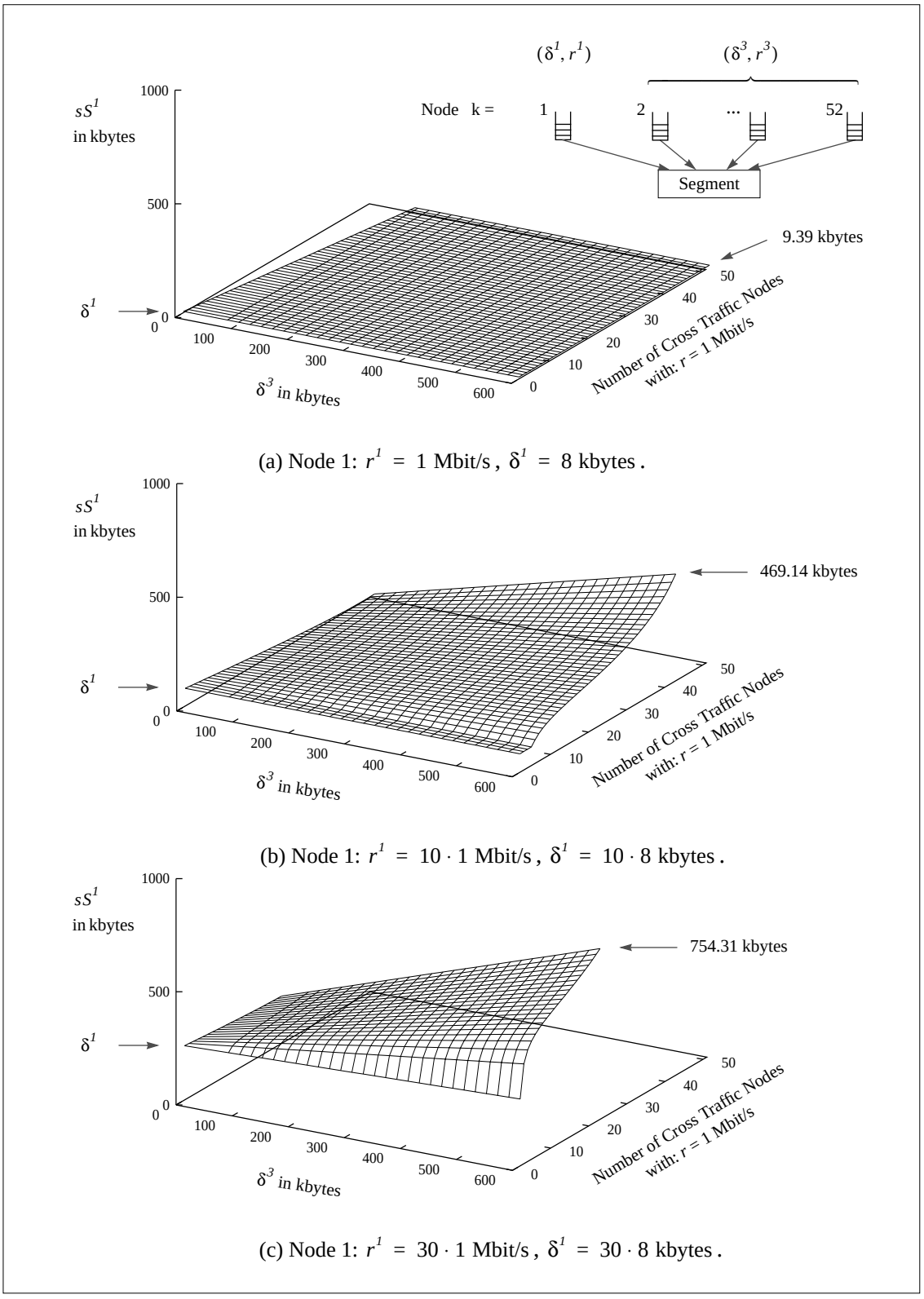
The second part of this section reports experimental results received for the end-to-end packet delay and the packet loss rate. Results for the resource utilization were also included. We do not explicitly discuss the throughput characteristics since: (1) Theorem 7.1 is based on the Bandwidth Test (Theorem 6.1) derived for the Guaranteed service, and (2) we used the same values for the Demand Priority per-packet overhead D_{pp} in the admission control. The properties discussed for this in Section 6.5.1 do thus also apply for the Controlled Load service.

7.3.1 The Impact of the Traffic Characteristics on the Buffer Space Requirements

Due to the simple service disciplines employed within hubs and switches in the bridged LAN, Controlled Load service flows may strongly interact with each other. The admission control considers this by reserving additionally buffer space in the network.

Figure 7.7 shows the dependencies for a single Level-2 cascaded network segment with 52 active network nodes as illustrated at the top of the figure. The buffer space sS^1 was computed for FLOW 1 which has the traffic characterisation: (δ^1, r^1) at the entrance of the segment. Results for three different cases are shown. In case (a), we first allocated a data rate of: $r^1 = 1$ Mbit/s and a burst size of: $\delta^1 = 8$ kbytes for FLOW 1. Afterwards we admitted 51 cross traffic flows, each of which had the same data rate but entered the shared segment at a different network node. For each setup, we computed the buffer space sS^1 of FLOW 1 while varying the total burst size δ^3 of all admitted cross traffic flows from 74.7 kbytes $(51 \cdot P_{max})$ to 600 kbytes. The last admission control test thus included the traffic parameters: $r^1 = 1$ Mbit/s, $\delta^1 = 8$ kbytes, $r^3 = 51$ Mbit/s, $\delta^3 = 600$ kbytes and $m = 52$, where (δ^3, r^3) describes the aggregated cross traffic. This follows the model introduced in Section 7.2.2 (see Figure 7.2). The parameter *m* denotes the number of nodes with reservations in the network[1].

1. In the admission control, we further used: (1) a time frame of: $TF = 20$ ms, (2) a packet count: $pcnt = 6$ for each 1 Mbit/s flow, (3) a per-packet overhead of: $D_{pp\_L2} = 21.45$ µs and an interrupt time of: $D_{it\_L2} = 554.11$ µs assuming 100 m UTP cabling, (4) a minimum service rate of: $R_{\mathrm{MIN\_N1}} = C_s/m$ for node $k = 1$, (5) a utilization factor of: $f = 0.9$, where the admission of 52 flows corresponded to the maximum number of flows that could be admitted in this setup.



(a) Node 1: $r^1 = 1$ Mbit/s , $\delta^1 = 8$ kbytes .

(b) Node 1: $r^1 = 10 \cdot 1$ Mbit/s , $\delta^1 = 10 \cdot 8$ kbytes .

(c) Node 1: $r^1 = 30 \cdot 1$ Mbit/s , $\delta^1 = 30 \cdot 8$ kbytes .

Figure 7.7: Buffer Space in Dependence of the Number of Cross Traffic Nodes and their Burst Sizes.

In the results for case (a) in Figure 7.7, we can observe that the impact of the cross traffic on the buffer space sS^1 is negligible. The maximum buffer space of FLOW 1 (9.39 kbytes) only differs from the initial burst size (8 kbytes) due to (1) the impact of the round-robin service policy, (2) the time it takes to interrupt the normal priority network service. Both is tightly limited. The independence of these results is caused by the round-robin service discipline, which isolates the data traffic of a network node as long as the data rate injected by the node, does not exceed the "Fair Bandwidth Share" ($\approx C_s / m$) of the node. This was however not the case in setup (a).

In the second example (b), we increased the data rate and the burst size of FLOW 1 such that: $r^1 = 10 \cdot 1$ Mbit/s , $\delta^1 = 10 \cdot 8$ kbytes , respectively, and then repeated the admission control. This basically used the same setup and the same parameters as described above. The number of network nodes with a reservation however decreased to $m = 43$ since the admission control only admitted a total of: $52 \cdot 1$ Mbit/s Controlled Load flows due to the utilization factor of: $f = 0.9$ that was used in the tests. The results in the corresponding diagram in Figure 7.7 show that the independence of the buffer space sS^1 is maintained when the number of nodes with a reservation is small. For $m > 5$ however, the impact increases gradually. We observed a maximum of 469.14 kbytes for the case that $42 \cdot 1$ Mbit/s cross traffic flows with a total burst size of 600 kbytes and $m = 43$ were admitted for the Controlled Load service.

The third diagram (c) in Figure 7.7 shows the equivalent admission control results for the case that: $r^1 = 30 \cdot 1$ Mbit/s and $\delta^1 = 30 \cdot 8$ kbytes . In this test, the maximum number of nodes with reservations thus further decreased to $m = 23$. Basically the same characteristics as discussed for case (b) can be identified. The computed results for the buffer space sS^1 however increase faster than previously observed, which is caused by the large resource share allocated for FLOW 1 and the worst-case policy considered in the computation (Theorem 7.2).

Figure 7.8 shows the dependencies for a single half-duplex switched link. In contrast to each of the diagrams in Figure 7.7 whose computation was based on a fixed data rate for FLOW 1, we additionally varied this parameter while computing the results in Figure 7.8. This was possible because the link had only two network nodes, which simplified the illustration. As in the previous case, we admitted flows with a data rate of 1 Mbit/s. The y-axis in Figure 7.8 shows the aggregated data rate of FLOW 1 (r^1). The data rate of the cross traffic sent from node 2 can be derived from r^1 by using the equation: $r^3 = 70$ Mbit/s $- r^1$. In the test, we thus always had resources equivalent to 70 Mbit/s allocated on the link and only changed the resource share of the two nodes. The burst size of FLOW 1 was fixed for all data rates ($\delta^1 = 100$ kbytes). In contrast, the burst size of the cross traffic was varied during the admission control ($\delta^3 = 20$ kbytes ... (20 kbytes) ... 600 kbytes). This parameter is shown at the x-axis in Figure 7.8.

It remains to remark that the admission of: $70 \cdot 1$ Mbit/s flows reflects the allocation limit for this setup including a high priority utilization factor of: $f = 0.9$. Furthermore, the admission control used the same parameters as listed for the Level-2 cascaded topology, but with $m = 2$.

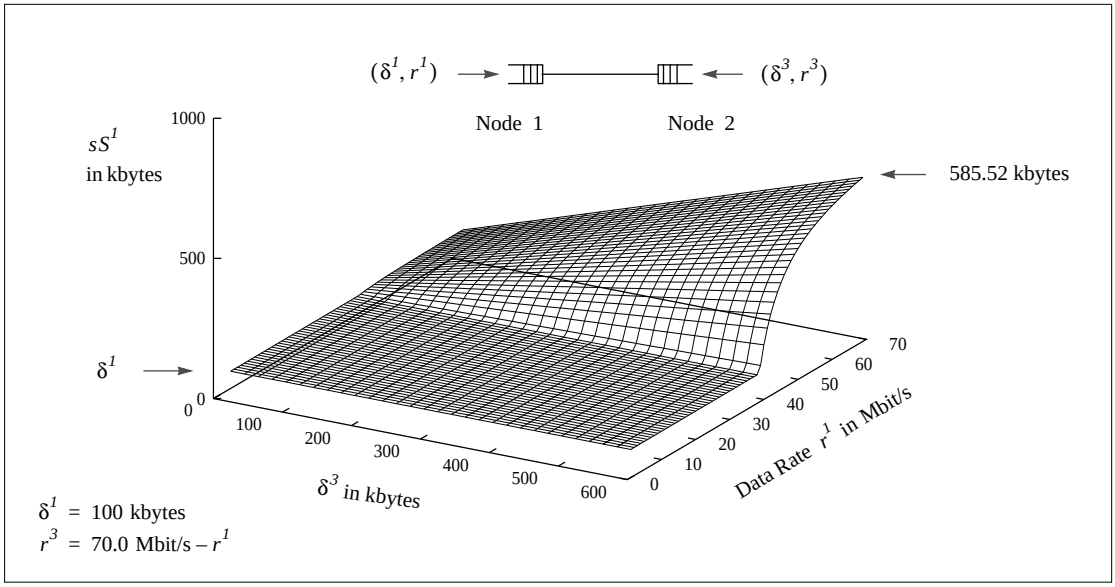


Figure 7.8: Buffer Space of FLOW 1 on a Half-Duplex Switched Link in Dependence of the Data Rate and the Cross Traffic Burst Size.

In Figure 7.8, we find that the results for the buffer space sS^1 include a large region in which sS^1 is independent of: (1) the data rate r^1 of FLOW 1, and (2) burst size δ^3 of the cross traffic from node 2. This is equivalent to the characteristic observed for the cascaded network. The “Fair Bandwidth Share” of node 1 is typically however much higher than in a cascaded network since the half-duplex link may only have two nodes with reservations. In this setup, we have on average a fair share of: $C_s/2$ because we only admitted homogeneous flows. If the allocated data rate exceeds this threshold, then the computed buffer space increases fast when δ^3 is large. The dependencies are discussed more in detail in the following using a numerical example.

The setup and the results for this are given in Table 7.1. Figure 7.9 shows the example network topology. We computed the buffer space for single flows of different data rate traversing the network from the data source S to the receiver R. The data path included the three LAN switches: *Sw1*, *Sw2*, *Sw3* and one hub that was denoted with *H1*. The source node *S* and the switches were interconnected via the half-duplex switched links: *L1*, *L2* and *L3*, respectively. The receiver was located on the Level-1 cascaded segment *L4*. Figure 7.9 further illustrates other switches and hubs in the bridged network. These are however not relevant for our discussion.

The buffer space in the example was computed for FLOW 1 whose traffic parameters (δ^1, r^1) are listed in the first two columns in Table 7.1. The cross traffic is described by the last four columns. To simplify the experimental setup, we assumed the same setup on each of the four links. Columns 7 and 8 list the total amount of cross traffic on the link. During the computation, we varied the high priority load from 20 Mbit/s to 60 Mbit/s. The total burst size however was left constant (400 kbytes) in all tests. Columns 9 and 10 describe the traffic (δ^2, r^2) that shared the high priority out-

put queue with FLOW 1 in each of the LAN switches along the data path. This assumes the model illustrated in Figure 7.2. We always selected the parameters: δ^2 and r^2 such that the cross traffic entering the link at both nodes (or at $k = 1$ and at $k \neq 1$) had the same characteristics. For the Level-1 cascaded segment, we further assumed 24 active network nodes. The results for the buffer space are then shown in the Columns 3 to 6, where sS_2^1 for example denotes the buffer space computed for FLOW 1 at the entrance to link *L2* (switch *Sw1*)[1].

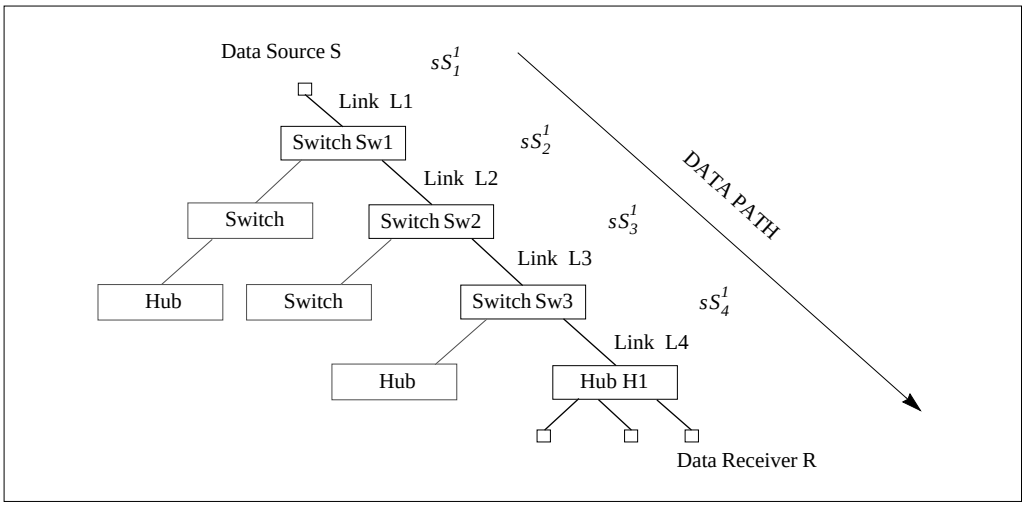


Figure 7.9: Example Network Topology for Results in Table 7.1 and Table 7.2.

Investigating the results, we find that in all tests, sS^1 increased along the data path. The per-hop growth rate depends on the characteristics of the high priority data traffic that enters each link, in particular: (1) the traffic parameters of FLOW 1 (δ^1, r^1), (2) the burstiness of the total traffic on the network $(\delta^1 + \delta^2 + \delta^3)$, (3) the spare network capacity $(C_s - r^1 - r^2 - r^3)$, (4) the total data rate that enters the link at node $k = 1$ $(r^1 + r^2)$, and (5) the minimum resource share R_{MIN_N1} of this node. For a cross traffic rate of only 20 Mbit/s, we can still only observe a moderate growth when compared with the results in row 2 and 3. This is because in this case, the network has a large amount of spare capacity available to clear the worst-case data backlog. The growth nevertheless differs substantially for different input parameters of FLOW 1 (δ^1, r^1) as can be observed for the three data sets computed in Table 7.1. For the first flow, with: $\delta^1 = 2$ kbytes, $r^1 = 0.128$ Mbit/s, the buffer space increases by 188% to 5.76 kbytes, whereas we have: 430.75% and a maximum of 106.15 kbytes for the 3 Mbit/s flow. These results increase significantly when the total cross traffic increases further. This may lead to large upper bounds computed with Theorem 7.3 when many bursty flows: (1) traverse across several segments in the bridged LAN, and (2) encounter large cross traffic reserva-

1. For the computation of the buffer space sS^1, we further used: (1) a time frame of: $TF = 20$ ms, (2) a per-packet overhead of: $D_{pp\_HD} = 8.555$ µs and an interrupt time of: $D_{it\_HD} = 252.67$ µs for all half-duplex switched links and $D_{pp\_L1} = 10.109$ µs, $D_{it\_L1} = 261.92$ µs for the Level-1 cascaded segment (corresponding to 100 m UTP cabling), (3) a minimum service rate of: $R_{MIN\_N1} = C_s/m$, and (4) a fixed packet size of 375 bytes for all data packets of all flows admitted.

tions. Large results can further be expected for cascaded networks including many nodes with reservations since the parameter R_{MIN_N1} for each of these nodes will be low. This can also be observed in the results shown in Table 7.1 for the Level-1 cascaded segment (L4).

Data Rate r^1 of FLOW 1 in Mbit/s	δ^1 (Source) in kbytes	sS_1^1 (Link L1) in kbytes	sS_2^1 (Link L2) in kbytes	sS_3^1 (Link L3) in kbytes	sS_4^1 (Link L4) in kbytes	Total Cross Traffic reserved on each of the Links: L1, L2, L3, L4		Cross Traffic on each Link sharing the Output Queue with FLOW 1	
						Data Rate ($r^2 + r^3$) in Mbit/s	Burst Size ($\delta^2 + \delta^3$) in kbytes	Data Rate (r^2) in Mbit/s	Burst Size (δ^2) in kbytes
0.128	2.0	2.93	3.86	4.78	5.76	20	400	10	200
1.5	10.0	20.62	31.24	41.86	53.08	20	400	10	200
3.0	20.0	41.24	62.48	83.72	106.15	20	400	10	200
0.128	2.0	3.43	4.86	6.30	7.86	40	400	20	200
1.5	10.0	26.39	42.78	59.17	77.01	40	400	20	200
3.0	20.0	52.78	85.55	118.33	154.03	40	400	20	200
0.128	2.0	5.13	8.27	11.40	15.22	60	400	30	200
1.5	10.0	45.87	81.73	117.60	161.29	60	400	30	200
3.0	20.0	91.73	163.47	235.20	322.57	60	400	30	200

Table 7.1: Buffer Space Requirements for FLOW 1 in Dependence of the Cross Traffic reserved along the Data Path.

In general, we will find that whenever a node *k* requires more resources than its fair bandwidth share, as assumed in all test setups in the numerical example, then the buffer space sS computed for the analysed flow basically can only remain low when the total traffic passed into the network is non-bursty. This follows from the worst-case assumptions applied during the derivation of Theorem 7.2. The results in Figure 7.7 and Figure 7.8 have however also shown that as long as the resources allocated for *k* are lower than the fair share, the growth rate of the buffer space stays low.

Data Rate r^1 of FLOW 1 in Mbit/s	Total Buffer Space for the Total CL Traffic to L1 (in Source) in kbytes	Total Buffer Space for the Total CL Traffic to L2 (in Sw1) in kbytes	Total Buffer Space for the total CL Traffic to L3 (in Sw2) in kbytes	Total Buffer Space for the Total CL Traffic to L4 (in Sw4) in kbytes	Total Cross Traffic reserved on each of the Links: L1, L2, L3, L4		Cross Traffic on each Link sharing the Output Queue with FLOW 1	
					Data Rate ($r^2 + r^3$) in Mbit/s	Burst Size ($\delta^2 + \delta^3$) in kbytes	Data Rate (r^2) in Mbit/s	Burst Size (δ^2) in kbytes
0.128	202.51	203.43	204.36	229.62	20	400	10	200
1.5	210.57	221.19	231.82	271.22	20	400	10	200
3.0	220.65	241.89	263.13	318.05	20	400	10	200
0.128	203.00	204.44	205.87	275.31	40	400	20	200
1.5	211.07	227.46	243.85	333.60	40	400	20	200
3.0	221.15	253.92	286.70	398.71	40	400	20	200
0.128	203.50	206.64	209.77	346.30	60	400	30	200
1.5	211.57	247.44	283.30	459.24	60	400	30	200
3.0	221.65	293.38	365.11	584.24	60	400	30	200

Table 7.2: Buffer Space Requirements of FLOW 1 and FLOW 2 for the Setup in Table 7.1.

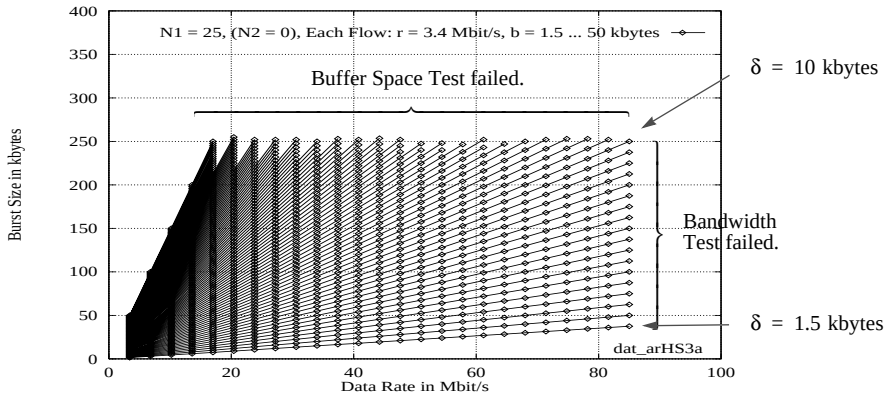
Table 7.2 provides the buffer space requirements for the sum of FLOW 1 and FLOW 2 $(\delta^1 + \delta^2, r^1 + r^2)$ in all LAN switches along the data path. These are complementary to the results in Table 7.1 since their computation was based on the same setup. The first column and the last four columns in each table are thus identical. We find that although the combined burst size of FLOW 1 and FLOW 2 is larger than δ^1, the result in Table 7.1 is lower than the corresponding multiple of sS^1. This is because the computation of sS^1 assumes that the data backlog of FLOW 1 is served only after the backlog of FLOW 2 and FLOW 3 have been processed, whereas the combined flow basically receives service instantly after the normal priority network service is interrupted. Furthermore, we can observe that the growth across the three half-duplex switched links is low because condition: $R_{\text{MIN\_N1}} > r^1 + r^2$ holds for all of these links. The buffer space requirements for switch Sw4 however increase substantially due to the low minimum resource share of: $C_s/24$ for each node in the cascaded network segment.

7.3.2 The Admissible Region

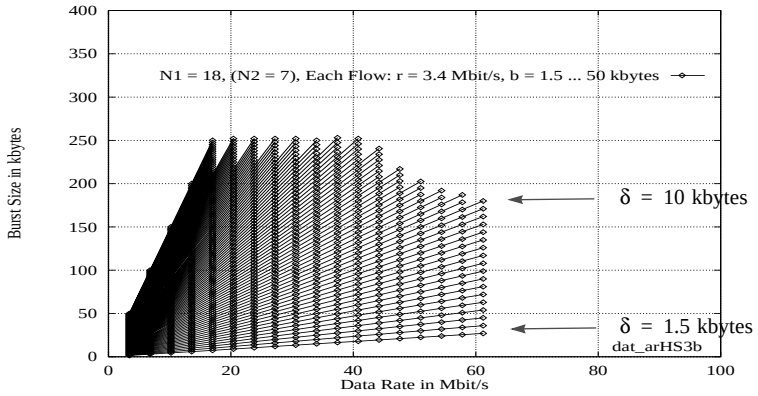
The Admissible Region can be viewed as the two dimensional resource space available for the reservation. We define it as the range of different (δ, r) parameters that lead to a successful admission of a flow to the service. This was motivated by the fact that Controlled Load service users may choose different sets of parameters for the same flow. Some sets may however have a higher chance of being accepted than others. In our experiments for example, the buffer space was often the limiting network resource such that flows were typically rejected by the Buffer Space Test. Selecting lower burst sizes δ resulting in lower buffer space requirements then often allowed the admission of a significantly larger number of flows even though each of them requested a larger data rate *r*.

The Admissible Region of a data flow at node *k* depends on the available resources on the outgoing segment such as the amount of unreserved bandwidth and the spare buffer space on *k*. Since this may vary on subsequent segments in the network e.g. when the available or the allocated resources differ substantially, a different admissible region can typically be defined for each segment along the data path. Finding the optimum resource parameters that satisfy the requirements of the flow but which are also most appropriate for all segments may thus be hard or even impossible, especially when the allocated resources change dynamically.

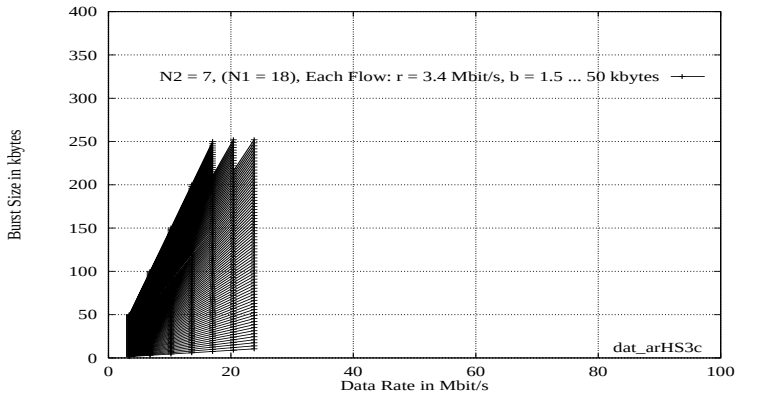
Figure 7.10 illustrates two examples for the Admissible Region of a flow that enters a half-duplex switched link at Node 1 (for the topology, see for example Figure 7.8). For this we performed a number of admission control tests. However, instead of admitting a single flow with different (δ, r) parameters, we admitted several homogeneous flows with a fixed set of parameters until we reached the allocation limit. Afterwards, we repeated the admission control using flows with the same data rate but a different burst size, and so on for a range of different δ values. In the diagrams in Figure 7.10, each admitted flow is represented by a mark. Flows with the same flow parameters are interconnected by a straight line. The results of all tests reflect the admissible region.



(a) Admissible Region on Node 1 in Example 1
(Maximum: 25 Flows allocated on Node 1 and 0 on Node 2).



(b) Admissible Region on Node 1 in Example 2
(Maximum: 18 Flows allocated on Node 1 and 7 on Node 2).



(c) Admissible Region on Node 2 in Example 2
(Maximum: 18 Flows allocated on Node 1 and 7 on Node 2).

Figure 7.10: Two Examples for the Admissible Region on a single Half-Duplex Switched Link.

The data rate requested for each flow in the admission control was: r = 3.4 Mbit/s. The burst size δ was varied from: 1.5 kbytes to 50 kbytes with an incremental step of 512 bytes[1].

The first diagram (a) in Figure 7.10 shows the results for the case that all real-time flows entered the link at Node 1. The admissible region is limited by the link capacity and the output buffer space in Node 1. The maximum number of flows admitted was 25 which corresponds to a total allocated bandwidth of 85 Mbit/s. This could however only be achieved for flows with burst sizes of $\delta \leq 10$ kbytes. For larger values, the number of flows and thus the total allocated bandwidth decreased significantly due to insufficient buffer space at Node 1. For the maximum tested burst size of: $\delta = 50$ kbytes, only 5 reservation requests passed the admission control. This was expected since Theorem 7.2 basically adds up the initial burst sizes of all flows without considering limiting constraints such as topology information or a statistical multiplexing between different flows. This simplified the calculus in Section 7.2.2 but may also result in high upper bounds. Whenever the sum of the burst sizes reached the maximum buffer space, which was 256 kbytes in this setup, all following reservation requests were rejected by the Buffer Space Test.

In the second experiment, we also reserved resources for flows entering the link from Node 2. The diagrams (b) and (c) in Figure 7.10 show the admissible region from the point of view of Node 1 and Node 2, respectively. The resource reservation used the same parameters as listed for Example 1. During the admission control, we first reserved resources alternately on both nodes until a total of 14 flows were admitted on the segment. Afterwards we only added flows on Node 1 until a reservation request was rejected by either the Bandwidth- or the Buffer Space Test.

As in the first example, a total of: $25 \cdot 3.4$ Mbit/s flows can be admitted on the link. The maximum of 18 flows on Node 1 is achieved for the same range of burst sizes ($1.5 \text{ kbytes} \leq \delta \leq 10 \text{ kbytes}$) as found for Example 1. Each of the 7 flows entering the link at Node 2 however may have a larger burst size (up to about 36 kbytes) since they may use the entire output buffer space available at this node. This shows that, a larger total burst size can be admitted on the network when the reservations are distributed across both nodes (or across several nodes in a cascaded network). The optimum is achieved for homogeneous bandwidth shares. If however a larger capacity than the fair bandwidth share is reserved as performed in Example 2, then more buffer space needs to be reserved due to potentially longer queuing delays. This was discussed in the previous section. The same basic characteristics as exhibited in Figure 7.8, can thus also be observed for the special case in diagram (b). It remains to remark in this context that we did not investigate the admissible region for the Guaranteed service because the corresponding admission control allocated the peak data rate for all real-time flows on the network.

1. The following parameters were additionally used in the admission control: (1) a time frame of: TF = 20 ms, (2) a per-packet overhead of: D_{pp_HD} = 8.555 µs and an interrupt time of: D_{it_HD} = 252.67 µs for the half-duplex switched link, (3) a utilization factor of: f = 1.0, (4) a minimum service rate of: $R_{MIN\_N1} = C_s/2$, (5) an ouput buffer space of 256 kbytes for both nodes on the link, and (5) a fixed packet size of 1383 bytes for all data packets of all flows admitted. The buffer space of 256 kbytes corresponds to the default memory that is available for each high priority queue in our prototype LAN switches. The example packet size of 1383 bytes was chosen because this is the average packet size of all data packets in the *MMC2* application trace analysed in Section 4.2.1.

7.3.3 Delay and Loss Characteristics in the 1L1S Test Network

In the following three sections, we discuss measurement results received for the end-to-end delay and the packet loss rate in three different network topologies. All experiments were based on the trace driven measurement approach described in Section 3.2.2. This used the application traces: *MMC1*, *MMC2*, *OVision*, and the source model traces: *POO1* and *POO3*, whose characteristics we discussed in Section 4.2.1 and Section 4.2.2, respectively. Furthermore, the measurement methodology applied to determine the packet delay and loss rate were reported in Section 3.6 and Section 3.7.

Table 7.3 and Table 7.4 show the source- and the token bucket parameters used for the above test traces in all three network topologies. For each trace, we carried out six different measurements. These differ in respect to the resource parameters allocated for each flow and the location where data flows entered the test network. The information in Columns 2 to 4 was taken from Table 4.1 and Table 4.2 for ease of reference. The last four columns list the resources allocated for a flow at the link layer. We always selected the token bucket parameters (δ, r) such that a large number of flows could be admitted. This was based on several initial experiments which showed that for our test traces, low bandwidth utilizations led to low packet delays despite of the larger burst size δ available for all admitted flows. The worst case delays were typically achieved with burst sizes in the order of a few kbytes, depending on the data rate and burstiness of the flow, because this also allowed a large number of flows to be admitted.

Test	Trace	Source	Average Data Rate generated in Mbit/s	Per-Flow Resources allocated.			
				Data Rate r in Mbit/s	Burst Size δ in kbytes	Max. Rate-Reg. Queue in Pkts.	*pcnt* in Pkts. (TF = 20ms)
1a	MMC2	JPEG Video	2.611	3.4	10.5	153	20
1b	MMC2	JPEG Video	2.611	3.4	10.5	153	20
1c	MMC2	JPEG Video	2.611	3.4	10.5	153	20
1d	MMC2	JPEG Video	2.611	3.4	25.4	144	29
1e	MMC2	JPEG Video	2.611	3.4	25.4	144	29
1f	MMC2	JPEG Video	2.611	3.4	25.4	144	29
2a	MMC1	JPEG Video	2.973	3.1	10.0	40	19
2b	MMC1	JPEG Video	2.973	3.1	10.0	40	19
2c	MMC1	JPEG Video	2.973	3.1	10.0	40	19
2d	MMC1	JPEG Video	2.973	3.1	23.0	31	26
2e	MMC1	JPEG Video	2.973	3.1	23.0	31	26
2f	MMC1	JPEG Video	2.973	3.1	23.0	31	26
3a	OVision	MPEG-1 Video	1.286	1.8	6.0	137	11
3b	OVision	MPEG-1 Video	1.286	1.8	6.0	137	11
3c	OVision	MPEG-1 Video	1.286	1.8	6.0	137	11
3d	OVision	MPEG-1 Video	1.286	1.8	15.8	129	20
3e	OVision	MPEG-1 Video	1.286	1.8	15.8	129	20
3f	OVision	MPEG-1 Video	1.286	1.8	15.8	129	20

Table 7.3: Source and Token Bucket Parameters for the Application Traces *MMC1*, *MMC2* and *OVision*.

Column 7 in both tables lists the maximum length of the rate regulator queue at the source node. Whenever this limit was exceeded, arriving data packets were dropped. In all experiments using the traces 1 - 4, this however never occurred. Packet loss was only observed in the POO3 tests which we thus marked with an asterisk (*). The loss can be explained with the infinite variance of the Pareto sources which occasionally generated several hundreds of data packets in a single ON interval.

Test	Model	Average per-flow data rate in Mbit/s	Peak / Average Rate Ratio	Per-Flow Resources allocated.		
				Data Rate r in Mbit/s	Burst Size δ in kbytes	Max. Rate-Reg. Queue in Pkts
4a	POO1	0.321	2	0.66	1.25	1
4b	POO1	0.321	2	0.66	1.25	1
4c	POO1	0.321	2	0.66	1.25	1
4d	POO1	0.321	2	0.60	5.0	1
4e	POO1	0.321	2	0.60	5.0	1
4f	POO1	0.321	2	0.60	5.0	1
5a	POO3	0.262	10	0.44	1.25	430 (*)
5b	POO3	0.262	10	0.44	1.25	430 (*)
5c	POO3	0.262	10	0.44	1.25	430 (*)
5d	POO3	0.262	10	0.44	2.50	430 (*)
5e	POO3	0.262	10	0.44	2.50	430 (*)
5f	POO3	0.262	10	0.44	2.50	430 (*)

Table 7.4: Source and Token Bucket Parameters for the Pareto Sources.

Since this does not reflect the behaviour of any application known to us, especially when we consider the average packet generation rate of 10 and the average data rate of 0.262 Mbit/s (see the POO3 source characteristics in Table 4.2), we believe that cutting the extreme tail of the pareto distribution actually led to more realistic results in this case. While investigating the Pareto source model, we further observed that packet loss is likely to occur as long as resources are allocated close to the average data rate (as performed in the POO3 tests). Note here that the rate regulator queue length does not grow linearly with the number of flows when these are aggregated and sent by a single network node. In this case, we found a strong decrease of the buffer space requirements due to the statistical multiplexing.

Column 8 in Table 7.3 shows the packet count which was used by the admission control for each application flow. The listed results were achieved with the Time Window algorithm and a time frame of: $TF = 20\,\text{ms}$. This algorithm was however not used for Pareto (POO1 and POO3) flows. Instead the admission control computed the packet count based on the fixed packet size (1280 bytes) specified for these flows in Table 4.2. This removed the overhead typically introduced by the Time Window algorithm and enabled a resource allocation up to the capacity limit of the network.

In this section, we discuss the measurement results received for a single Level-1 cascaded segment which we denoted as the *1L1S Test Network*. The topology is shown in Figure 7.11. The Measurement Client was connected to switch *Sw1* (sending LAN adapter card) and to hub *H1* (receiving

LAN adapter card). High priority cross traffic was sent by the nodes 2 - 12. Node 13 generated best effort traffic equivalent to more than 80 Mbit/s (not rate regulated) such that the Level-1 cascaded segment was always overloaded. This used 1500 byte packets to achieve the worst case impact.

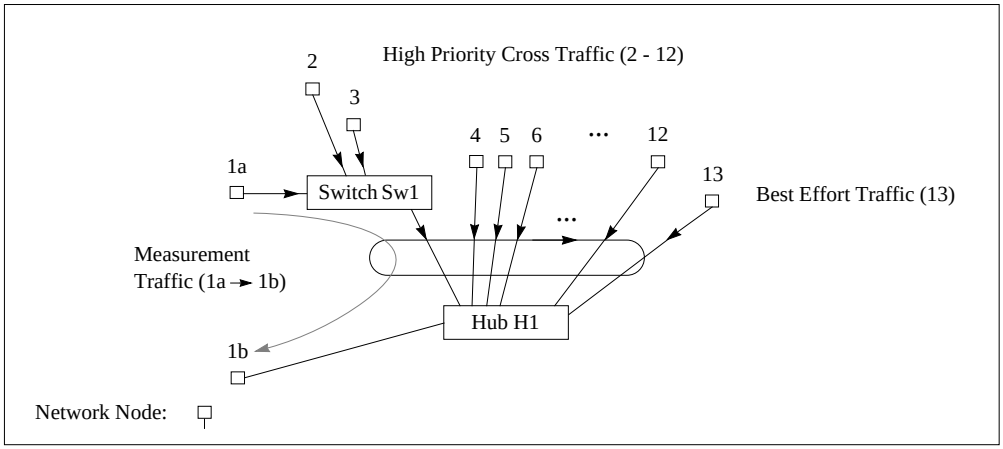


Figure 7.11: Measurement Setup in the 1L1S Test Network.

In all experiments, we measured the end-to-end delay of a single flow sent by the Measurement Client. Note that this did not include the delay in the rate regulator since we were interested in the characteristics of the queuing delay in the network. The measured delay was thus basically introduced in the high priority output queue of Sw1 at the entrance to the shared segment. Additionally, we measured: (1) the packet loss rate of the total traffic that entered the segment through Sw1, and (2) the average high priority data rate. The former was computed based on the packet drop counter of the switch port (for details, see the MIB counters in Table 3.1). This covered the traffic from the Measurement Client and the traffic from the nodes 2 and 3. To determine the average high priority data rate, we set appropriate filter entries in switch Sw1 such that a copy of all high priority data packets was also forwarded to a particular output port (not shown). The best effort traffic was directed to another port (also not shown). Both of them differed to the ports connecting the Measurement Client and the High Priority Traffic Clients at node 2 and node 3. This ensured that the results of the MIB counter *ifOutOctet* in Table 3.1 could be used for computing the average data rate, and prevented any undesired interference between the flows.

The details of the flow distribution in the test network and the measurement results are shown in Table 7.5 and Table 7.6. In all experiments, only homogeneous flows were admitted[1]. The maximum for this is provided in Column 3. Each result represents the allocation limit for the resources (δ, r) specified in Table 7.3 and Table 7.4. More specifically: all results for the tests: *a* - *c* in Col-

1. The following parameters were used for the admission control: (1) a time frame of TF = 20 ms, (2) a per-packet overhead of: D_{pp_L1} = 10.109 µs and an interrupt time of: D_{it_L1} = 261.92 µs corresponding to 100 m UTP cabling as used on the Level-1 cascaded segment during the experiments, (3) a minimum service rate of: $R_{\mathrm{MIN\_N1}} = C_s / m$ for each node with reservations on the segment, where: m = 10, and (4) a buffer space of 256 kbytes in switch Sw1.

umn 3 correspond to the bandwidth limit for the setup (additionally flows were rejected by the Bandwidth Test), whereas the results for the tests: *d* - *f* reflect the maximum buffer space in Sw1 (additionally flows were rejected by the Buffer Space Test).

Test	Trace	Number of Flows admitted	Topology Information			Measured Parameters						
			Number of Flows sent *downstream : upstream*	Number of Flows sent from Nodes:		High Priority Data Rate in Mbit/s	Pkt. Loss Rate in %	Ave. Delay in ms	90.0 % in ms	99.0 % in ms	Max. Delay in ms	Ave. Packet Size in Bytes
				2, 3	4 ... 12							
1a	MMC2	21	19 : 2	9, 9	2 x 1	55.870	0	1.240	2.385	5.585	11.555	1383
1b	MMC2	21	16 : 5	8, 7	5 x 1	55.516	0	1.203	2.285	5.285	13.205	1383
1c	MMC2	21	12 : 9	6, 5	9 x 1	56.045	0	1.230	2.335	5.415	12.565	1383
1d	MMC2	10	9 : 2	4, 4	2 x 1	29.203	0	1.019	1.395	3.065	9.535	1383
1e	MMC2	12	7 : 5	3, 3	5 x 1	32.315	0	1.089	1.545	3.975	14.815	1383
1f	MMC2	14	5 : 9	2, 2	9 x 1	37.646	0	1.245	2.005	5.935	16.645	1383
2a	MMC1	23	21 : 2	10, 10	2 x 1	68.801	0	1.773	4.035	7.775	14.215	1357
2b	MMC1	23	18 : 5	9, 8	5 x 1	68.797	0	1.734	3.925	7.715	13.185	1356
2c	MMC1	23	14 : 9	7, 6	9 x 1	68.809	0	1.841	4.335	8.355	13.863	1356
2d	MMC1	12	10 : 2	5, 4	2 x 1	35.883	0	1.077	1.645	4.085	10.075	1356
2e	MMC1	13	8 : 5	4, 3	5 x 1	38.880	0	1.102	1.685	4.585	11.865	1357
2f	MMC1	15	6 : 9	3, 2	9 x 1	44.864	0	1.367	2.565	7.085	18.315	1356
3a	OVision	40	31 : 9	15, 15	9 x 1	51.481	0	0.758	0.955	1.455	6.455	1333
3b	OVision	40	22 : 18	11, 10	9 x 2	51.346	0	0.779	1.005	1.675	6.635	1333
3c	OVision	40	13 : 27	6, 6	9 x 3	51.089	0	0.798	1.015	2.055	7.785	1332
3d	OVision	21	12 : 9	6, 5	9 x 1	27.339	0	0.712	0.825	1.015	2.035	1333
3e	OVision	26	8 : 18	4, 3	9 x 2	33.251	0	0.727	0.865	1.125	4.675	1332
3f	OVision	32	5 : 27	2, 2	9 x 3	40.558	0	0.742	0.885	1.305	6.115	1332

Table 7.5: Measured Packet Delay and Loss Rate for the Application Traces: *MMC1*, *MMC2* and *OVision* in the Level-1 Cascaded Test Network.

Test	Model	Number of Flows admitted	Topology Information			Measured Parameters					
			Number of Flows sent *downstream : upstream*	Number of flow sent from nodes:		High Priority Data Rate in Mbit/s	Pkt. Loss Rate in %	Ave. Delay in ms	90.0 % in ms	99.0 % in ms	Max. Delay in ms
				2, 3	4 ... 12						
4a	POO1	129	93 : 36	46, 46	9 x 4	46.026	0	0.696	0.845	1.085	2.085
4b	POO1	129	57 : 72	28, 28	9 x 8	44.995	0	0.706	0.865	1.175	2.825
4c	POO1	129	21 : 108	10, 10	9 x 12	45.839	0	0.707	0.855	1.165	2.305
4d	POO1	76	40 : 36	20, 19	9 x 4	26.185	0	0.672	0.765	0.935	1.485
4e	POO1	102	30 : 72	15, 14	9 x 8	34.620	0	0.686	0.805	1.035	1.785
4f	POO1	128	20 : 108	10, 9	9 x 12	45.077	0	0.705	0.845	1.145	2.145
5a	POO3	201	138 : 63	70, 67	9 x 7	68.845	0	0.895	1.145	3.695	19.035
5b	POO3	201	93 : 108	46, 46	9 x 12	68.234	0	0.992	1.285	5.835	21.225
5c	POO3	201	48 : 153	25, 22	9 x 17	68.726	0	1.007	1.325	6.145	21.175
5d	POO3	136	73 : 63	36, 36	9 x 7	45.241	0	0.713	0.875	1.175	12.375
5e	POO3	161	53 : 108	26, 26	9 x 12	54.185	0	0.725	0.965	1.395	16.315
5f	POO3	186	33 : 153	16, 16	9 x 17	61.515	0	0.783	1.025	1.635	17.725

Table 7.6: Measured Packet Delay and Loss Rate for the Pareto Sources in the Level-1 Cascaded Test Network.

Column 4 shows the ratio of the flows sent through switch Sw1 (*downstream*) into the Level-1 cascaded segment versus the number of flows arriving at Sw1 (*upstream*) after traversing the shared segment. The sum of both is always equal to the number of flows admitted. The columns 5 and 6 provide detailed information about how many flows entered the network at each node. In Test 1a for example, each of the nodes 2 and 3 sent 9 flows into the network (9, 9). If we take the Measurement Client into account then we have 19 downstream high priority MMC2 flows. Furthermore, the network included 2 active High Priority Traffic Clients each passing a single flow (2 x 1) into the shared segment. These were located at two of the nine (4 - 12) nodes directly connected to hub H1. The remaining 7 of these nodes were inactive. In contrast, in Test 1f, only two flows entered the test network from nodes 2 and 3 (2, 2), whereas each of the nodes: 4 - 12 passed a single flow into the Level-1 cascaded segment (9 x 1). By additionally considering the Measurement Client, we obtain: $1 + 2 + 2 + 9 \cdot 1 = 14$ MMC2 flows that were in the network in this experiment.

The remaining columns contain the parameters measured. This includes: (1) the average data rate of the total high priority traffic, (2) the packet loss rate measured at the entrance to the Level-1 cascaded network, (3) the average-, the 90.0, 99.0 percentile, and the maximum packet delay recorded by the Measurement Client, and (4) the average packet size of all high priority traffic (application traces only). For each test, the measurement interval was 30 minutes with an additional warm-up time of 2 minutes.

In all measurements, we did not observe a single packet loss in the network. Furthermore, the average delay is in the order of 1 ms, which was however expected considering the observations made in Section 4.3.3 for a loss free data transmission. Both results represents a sufficient quality for a Controlled Load service. The highest average delay was measured in the MMC1 tests: 2a - 2c (~1.7 - 1.8 ms). In these tests, we could however also observe a high average high priority data rate on the network (~ 68 Mbit/s for a total bandwidth reservation of: $23 \cdot 3.1$ Mbit/s $= 71.3$ Mbit/s). Even though the MMC1 trace is less bursty than for example the MMC2 trace (as shown in Section 4.2.1) we measured a lower average delay for the latter. This is because we allocated more resources for each individual MMC2 flow which led to a lower high priority data rate on the network and thus to a lower average packet delay. Furthermore, the results for both traces are higher than the results achieved with the Pareto sources.

The lowest average delays (~0.7 ms) were received in the POO1 tests (4a - 4f) in which we allocated resources equivalent to the peak data rate. For the measurements 4a - 4c for example, this implied a total bandwidth allocation of: $129 \cdot 0.66$ Mbit/s $= 85.14$ Mbit/s after all flows had been admitted. A network bandwidth higher than this was only reserved for POO3 sources in the tests 5a - 5c in Table 7.6 ($201 \cdot 0.44$ Mbit/s $= 88.44$ Mbit/s).

In the POO3 tests we measured the highest maximum delays (12.4 - 21.2 ms). A long tail in the distribution of the results can be identified. This is similar to the characteristics observed in Figure 4.15 for this source model. In contrast, the results received in the MMC1 and MMC2 measurements are typically significantly lower despite of the higher average delays measured for these trace files.

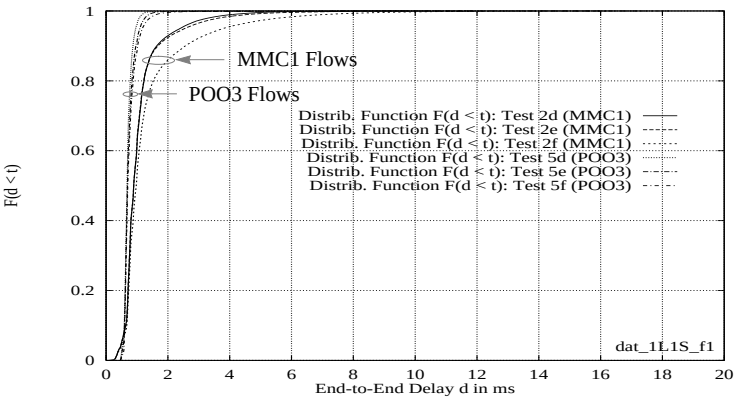


Figure 7.12: Distribution Function for Tests 2d - 2f (MMC1) and 5d - 5f (POO3) in the 1L1S Test Network.

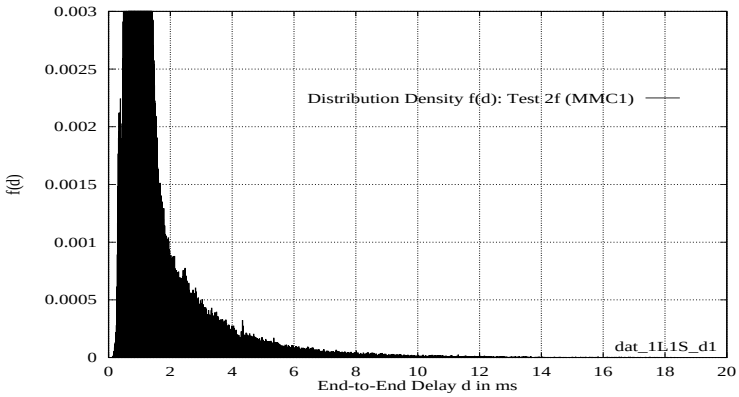


Figure 7.13: Distribution Density corresponding to Test 2f (MMC1, 1L1S Topology) in Figure 7.12.

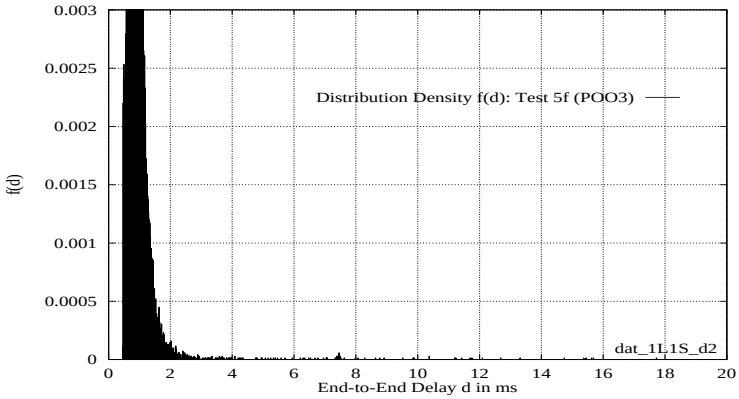


Figure 7.14: Distribution Density corresponding to Test 5f (POO3, 1L1S Topology) in Figure 7.12.

To illustrate the differences we plotted the delay distribution function of the tests: 2d - 2f (MMC1) and 5d - 5f (POO3) in Figure 7.12. Figure 7.13 and Figure 7.14 show the corresponding distribution density for Test 2f and Test 5f. Similar graphs are obtained for the other results. These are however omitted here. We selected the tests: 2f and 5f for illustration because they caused the largest maximum delays in the bridged test topology discussed later in Section 7.3.5.

7.3.4 Delay and Loss Characteristics in the 1HDL Test Network

In the second set of experiments, we measured the performance parameters across a single half-duplex switched link. This was to investigate whether and how a half-duplex switched network topology changes the delay characteristics observed in the Level-1 cascaded network. The *1HDL Test Network* which was used for these measurements is illustrated in Figure 7.15. The Measurement Client was connected to the switches Sw1 and Sw2. High priority cross traffic was sent by the High Priority Traffic Clients located at the nodes 2 - 9 (connected to Sw1) and the nodes 10 and 11 (connected to Sw2). Node 12 additionally overloaded the test link with best effort traffic.

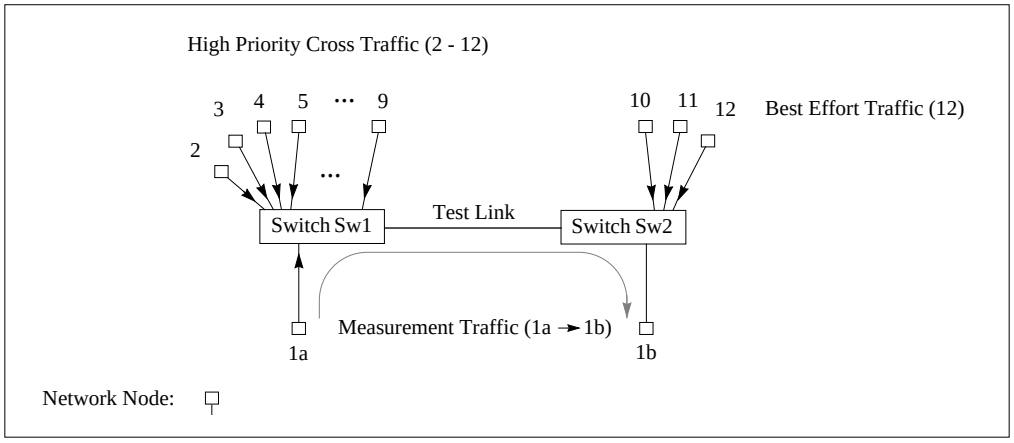


Figure 7.15: Measurement Setup for the Half-Duplex Switched Link.

In all experiments, we measured the same performance parameters as described for the Level-1 cascaded network. This was based on the same measurement methods and the same setup. The admission control only differed in: (1) the topology specific parameters used for the per-packet overhead (D_{pp_HD} = 8.555 µs) and the interrupt time (D_{it_HD} = 252.67 µs), and (2) the minimum service rate of: $R_{MIN\_N1} = C_s/2$ considered for each of the two LAN switches on the test link (m = 2). Both switches had buffer space of 256 kbytes for all output ports. Furthermore, the characteristics of the best effort traffic generated by node 12 in Figure 7.15 were identical to those used in the experiments reported in the previous section.

The details of the flow distribution in Figure 7.15 and the measurement results are shown in Table 7.7 for the application traces and in Table 7.8 for the Pareto sources. Both tables are organized in the same way as Table 7.5 and Table 7.6 in the previous section. We thus shorten the following

description and basically only discuss the differences. The number of flows admitted for Controlled Load service can again be found in Column 3. Since a flow may enter the test link from either switch Sw1 or switch Sw2, Column 4 provides the ratio for the flow distribution.

Test	Trace	Number of Flows admitted	Topology Information			Measured Parameters						
			Number of Flows sent from: Sw1 : Sw2	Number of Flows sent from Nodes: 2 ... 9	Number of Flows sent from Nodes: 10, 11	High Priority Data Rate in Mbit/s	Pkt. Loss Rate in %	Ave. Delay in ms	90.0 % in ms	99.0 % in ms	Max. Delay in ms	Ave. Packet Size in Bytes
1a	MMC2	22	22 : 0	(7), 2	0, 0	59.120	0	1.407	2.865	6.325	13.275	1383
1b	MMC2	22	17 : 5	(2), 2	2, 3	58.775	0	1.322	2.605	5.735	12.025	1383
1c	MMC2	22	11 : 11	(3), 1	5, 6	59.127	0	1.018	1.615	3.745	10.515	1383
1d	MMC2	20	10 : 4	(2), 1	2, 2	37.972	0	1.173	1.775	4.395	10.485	1382
1e	MMC2	20	10 : 7	(2), 1	3, 4	45.656	0	1.477	2.895	6.745	15.525	1383
1f	MMC2	20	10 : 10	(2), 1	5, 5	53.086	0	1.690	3.525	8.025	19.365	1383
2a	MMC1	24	24 : 0	(2), 3	0, 0	71.775	0	1.743	3.835	6.855	13.545	1356
2b	MMC1	24	18 : 6	(3), 2	3, 3	71.820	0	1.643	3.665	7.235	16.615	1356
2c	MMC1	24	12 : 12	(4), 1	6, 6	71.901	0	1.160	2.125	4.535	11.745	1357
2d	MMC1	15	11 : 4	(3), 1	2, 2	44.933	0	1.219	2.155	5.265	14.425	1356
2e	MMC1	19	11 : 8	(3), 1	4, 4	56.825	0	1.517	3.175	7.155	15.655	1356
2f	MMC1	22	11 : 11	(3), 1	5, 6	65.917	0	1.947	4.435	9.335	18.985	1356
3a	OVision	42	42 : 0	(6), 5	0, 0	54.025	0	0.797	0.985	1.895	6.395	1332
3b	OVision	42	32 : 10	(3), 4	5, 5	54.125	0	0.786	0.975	1.885	6.785	1332
3c	OVision	42	21 : 21	(6), 2	10, 11	53.937	0	0.752	0.915	1.465	6.595	1332
3d	OVision	24	20 : 4	(5), 2	2, 2	30.750	0	0.721	0.825	1.015	5.655	1333
3e	OVision	30	20 : 10	(5), 2	5, 5	38.556	0	0.731	0.855	1.105	5.965	1333
3f	OVision	36	20 : 16	(5), 2	8, 8	46.741	0	0.750	0.885	1.385	7.375	1332

Table 7.7: Measured Packet Delay and Loss Rate for the Application Traces: *MMC1*, *MMC2* and *OVision*, across 2 LAN Switches interconnected by a single Half-Duplex Switched Link.

Test	Source	Number of Flows admitted	Topology Information			Measured Parameters					
			Number of Flows sent from: Sw1 : Sw2	Number of Flows sent from Nodes: 2 ... 9	Number of Flows sent from Nodes: 10, 11	High Priority Data Rate in Mbit/s	Pkt. Loss Rate in %	Ave. Delay in ms	90.0 % in ms	99.0 % in ms	Max. Delay in ms
4a	POO1	132	132 : 0	(19), 16	0, 0	47.206	0	0.703	0.825	1.035	2.125
4b	POO1	132	102 : 30	(17), 12	15, 15	47.246	0	0.702	0.835	1.075	2.395
4c	POO1	132	66 : 66	(9), 8	33, 33	46.084	0	0.684	0.795	1.025	2.415
4d	POO1	60	50 : 10	(7), 6	5, 5	20.244	0	0.660	0.715	0.835	1.325
4e	POO1	80	50 : 30	(7), 6	15, 15	27.121	0	0.663	0.725	0.875	1.455
4f	POO1	100	50 : 50	(7), 6	25, 25	36.164	0	0.667	0.745	0.925	1.635
5a	POO3	202	202 : 0	(26), 25	0, 0	67.203	0	1.059	1.045	12.045	20.405
5b	POO3	204	154 : 50	(20), 19	25, 25	68.527	0	1.078	1.095	11.795	26.915
5c	POO3	204	104 : 100	(19), 12	50, 50	69.754	0	0.839	1.005	2.975	18.625
5d	POO3	141	101 : 40	(16), 12	20, 20	45.541	0	0.705	0.805	1.055	12.445
5e	POO3	171	101 : 70	(16), 12	35, 35	58.819	0	0.733	0.855	1.285	13.685
5f	POO3	201	101 : 100	(16), 12	50, 50	69.283	0	0.840	1.005	3.345	25.475

Table 7.8: Measured Packet Delay and Loss Rate for the Pareto Sources across 2 LAN Switches interconnected by a single Half-Duplex Switched Link.

The first number in Column 5 (in brackets) specifies the number of flows that entered the network at node 2, whereas the second defines the number sent from each of the other nodes (3 - 9) connected to Sw1. This differentiation was required since the total number of flows could typically not be evenly distributed amongst all nodes. Column 6 shows the number of flows sent by node 11 and node 12. The setup for Test 1a thus included 7 MMC2 flows from node 2, and two from each of the nodes: 3 - 9. If we additionally consider the single flow sent by the Measurement Client, we get: $1 + 7 + 7 \cdot 2 = 22$ for the number of flows used in this experiment. In Test 1f for example, we had 2 flows from node 2, 1 flow from each of the nodes: 3 - 9, and 5 flows from node 10 and node 11. This resulted in 20 MMC2 flows in the test network. The measurement results are shown in the Columns 7 - 12. Column 13 additionally provides the average packet size measured for the application traces.

Comparing the total number of flows admitted on the half-duplex switched link with the results received for the Level-1 cascaded network, we find that except with the setups in the tests 4d - 4f, a larger number of Controlled Load flows could be supported in the switched topology. This can typically be achieved due to the higher network capacity available (see for example Figure 4.10). The largest bandwidth reservations across the half-duplex switched link were made in the measurements 5b and 5c in which we allocated: $204 \cdot 0.44 \text{ Mbit/s} = 89.76 \text{ Mbit/s}$. Under certain conditions, the admission control might however not be able to reach the utilization achieved in the Level-1 cascaded network. In the POO1 tests 4d - 4f, this was caused by the comparatively large burst size δ requested for each POO1 flow and the buffer space limit of 256 kbytes in switch Sw2. Both led to early rejections from the Buffer Space Test since all high priority traffic from the nodes 10 and 11 in Figure 7.15 did have to enter the test link at switch Sw2. In contrast, in the Level-1 cascaded test network, the upstream high priority flows were distributed amongst the nodes: 4 - 12. These however had a larger total high priority output buffer space than switch Sw2.

In all experiments reported in Table 7.7 and Table 7.8, we did not observe any packet loss for high priority traffic. The loss measurements included all flows that entered the test link at switch Sw1. Since their total number was always at least as high as the total number of flows from node 11 and node 12, it is very likely that no loss did occur at the output queue of switch Sw2. We can further observe that the results for the average delay and the 90.0 percentile differ only marginally from the results received in the Level-1 cascaded network. These results can however not easily be compared because of the different total numbers of flows admitted and the different flow distributions used. Nevertheless the differences are typically in the order of less than 0.5 ms for the average delay and less than 2 ms for the 90.0 percentile which is negligible for existing real-time applications such as voice conferencing with an end-to-end delay budget of over 100 ms.

The results for the 99.0 percentile and the maximum delay differ more significantly. The largest delays were measured in the POO3 tests (12.4 - 25.5 ms). Note the result for test 5f (25.475 ms) which is higher than the maximum observed in Figure 4.11 for a switch with 256 kbytes buffer space. This can be explained by the different experimental setup used.

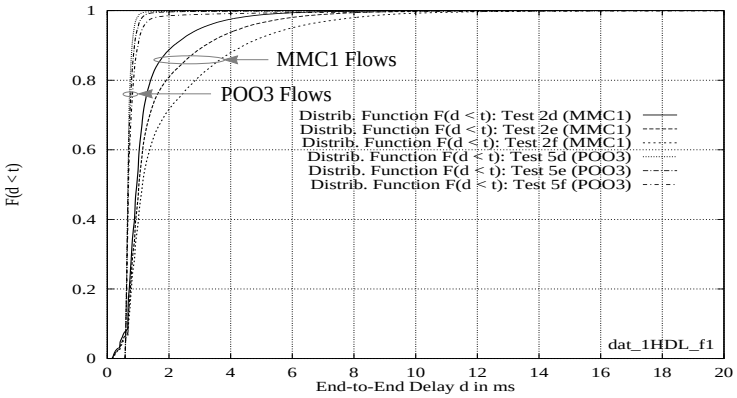


Figure 7.16: Distribution Function for Tests 2d - 2f (MMC1) and 5d - 5f (POO3) in the 1HDL Test Network.

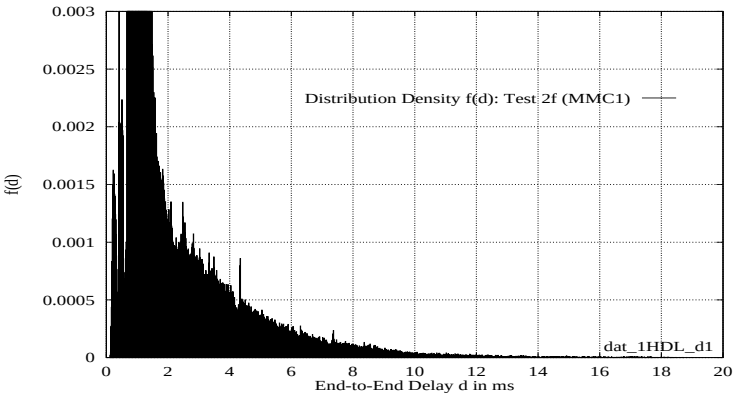


Figure 7.17: Distribution Density corresponding to Test 2f (MMC1, 1HDL Topology) in Figure 7.16.

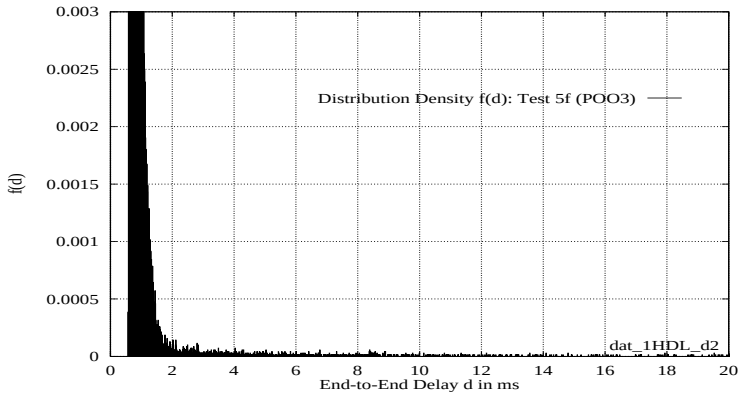


Figure 7.18: Distribution Density corresponding to Test 5f (POO3, 1HDL Topology) in Figure 7.16.

In Section 4.3.3, all traffic entered the test link at the same switch, whereas in test 5f the half-duplex switched link was loaded from both switches. In the worst case, when the output queues of both switches in Figure 7.15 are full and both receive the same service ($C_s/2$), then the maximum delay may be as high as twice ($\sim 2 \cdot 23 = 46$ ms) the result observed in Figure 4.11. We can thus expect long maximum delays in bridged networks, provided: (1) this consists of bridges interconnected by half-duplex switched links, (2) the links are traversed in both directions, and (3) the traffic is bursty over long time scales.

Figure 7.16 shows the distribution function for the MMC1 Tests 2d - 2f and the POO3 Tests 5d - 5f. The corresponding distribution density for the tests 2f and 5f are provided in Figure 7.17 and Figure 7.18. A comparison with the results in Figure 7.12, Figure 7.13 and Figure 7.14 shows that the results of both topologies have the same basic characteristics. We believe that the delay differences are mainly caused by the different flow distributions used in the two test networks.

7.3.5 Delay and Loss Characteristics in the 4HDL Test Network

In the following section we discuss the measurement results received in a bridged test network consisting of five LAN switches interconnected by half-duplex switched links. The network which we denoted as the *4HDL Test Network* is illustrated in Figure 7.19. The upper part of the picture shows the network topology, the lower part the data flows during the experiments. The end-to-end packet delay was measured by the Measurement Client whose two LAN adapter cards were connected to switch Sw1 and switch Sw5, respectively. Controlled Load flows entered the test network at the nodes: 2 - 13 (cross traffic) and at the Measurement Client (measurement traffic).

All flows from the nodes: 2, 4, 7 and 10 traversed two half-duplex links along the data path of the measurement traffic as illustrated in Figure 7.19. In contrast, the flows sent from the nodes: 3, 5, 8 and 11 only travelled across a single link downstream with the measurement traffic. Upstream Controlled Load flows were generated at the nodes: 6, 9, 12 and 13 and also only forwarded across a single half-duplex switched link. The particular data path of each flow was enforced by addressing the corresponding data packets with a unique multicast address and installing appropriate filter entries in the LAN switches. Best effort traffic was sent by node 14. It traversed the entire upstream data path from switch Sw5 to switch Sw1 and had the same characteristics as in the previous experiments.

In this topology, we measured the packet loss rate at the output queue of the switches: Sw1 (to link L1), Sw2 (to L2), Sw3 (to L3) and Sw4 (to L4). In switch Sw1 for example, this included the flows from the nodes: 2, 3 and the measurement traffic, in switch Sw4, this detected loss of data packets from the nodes: 7, 10, 11 and from the Measurement Client. The average high priority data rate was recorded for link L4. The Measurement Client measured the end-to-end delay for all data packets of a single flow traversing the entire test network from switch Sw1 to switch Sw5. The measurement methods for these parameters were identical to those used in the 1L1S- and the 1HDL test networks.

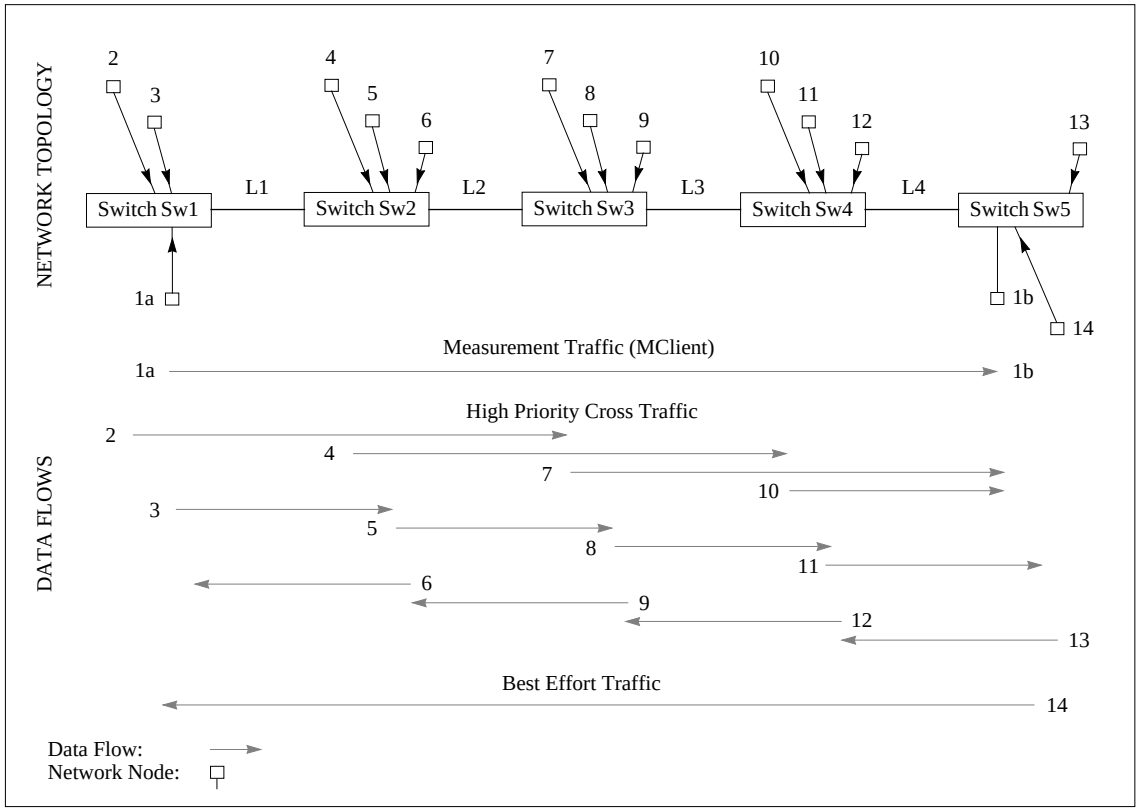


Figure 7.19: Measurement Setup in the 4HDL Test Network.

Since the test network had a half-duplex switched topology, the admission control could use the same topology specific parameters (D_{pp_HD}, D_{it_HD}, R_{MIN_N1}, $m = 2$) as used for the single link (1HDL) test network. In contrast to the numerical example in Section 7.3.1 (see the discussion for Figure 7.9), we however did not increase the buffer space allocation for flows traversing several switches in the experiments reported in this section. Instead, the admission control treated all Controlled Load flows as if these had entered the test network at the corresponding local switch, neglecting any traffic distortions along the data path already traversed by these flows. This led to the same bandwidth- and buffer space reservations on all half-duplex links in the bridged topology.

Such an allocation strategy assumes that the statistically distributed use of buffer space in switches can compensate for the traffic distortions introduced in the network. This was motivated by several observations. First, we found in our experiments, that whenever resources are conservatively allocated at the edge of the bridged network, then the packet loss rate in the core of the network was null or negligible. We believe that this was due to statistical multiplexing. Second, feedback effects which may additionally distort the traffic characteristics can not occur. Third, in real LANs we do not expect reservations to be made up to the capacity limit of the network. More realistic utilization factors of: $f \leq 0.9$ however are likely to enforce sufficient spare bandwidth such that packet loss within the network is eliminated or at least significantly reduced (provided we sufficiently restrict the Controlled Load traffic at the entrance to the bridged LAN).

Test	Trace	Number of Flows admitted (L1, L2, L3, L4)	Topology Information					Measured Parameters					
			Number of Flows sent: *downstream : upstream*	Number of Flows sent from Nodes:			Cross-Switch Reservation in Mbit/s	High Priority Data Rate on L4 in Mbit/s	Pkt. Loss Rate in %	Ave. Delay in ms	90.0% in ms	99.0% in ms	Max. Delay in ms
				2, 4, 7, 10	3, 5, 8, 11	6, 9, 12,13							
1a	MMC2	22	22 : 0	7	(14),7	0	27.20	58.357	0	3.541	6.975	12.195	23.635
1b	MMC2	22	17 : 5	8	(8), 0	5	30.60	58.623	0	3.746	7.525	12.845	26.385
1c	MMC2	22	11 : 11	5	(5), 0	11	20.40	59.016	0	2.670	5.105	9.545	21.735
1d	MMC2	20	10 : 4	4	(5), 1	4	17.00	37.272	0	2.588	4.645	8.675	20.525
1e	MMC2	20	10 : 7	4	(5), 1	7	17.00	45.972	0	3.595	6.925	12.395	25.445
1f	MMC2	20	10 : 10	4	(5), 1	10	17.00	54.065	0	4.249	8.395	14.705	30.535
2a	MMC1	24	24 : 0	11	(12),1	0	37.20	71.789	0	5.448	9.965	15.215	25.065
2b	MMC1	24	18 : 6	6	(11),5	6	21.70	71.791	0	6.655	12.535	19.445	33.555
2c	MMC1	24	12 : 12	5	(6), 1	12	18.60	71.875	0	5.660	11.015	17.094	31.885
2d	MMC1	15	11 : 4	5	(5),0	4	18.60	44.887	0	3.158	6.125	10.695	19.815
2e	MMC1	19	11 : 8	5	(5),0	8	18.60	56.936	0	4.988	9.995	16.795	33.165
2f	MMC1	22	11 : 11	5	(5),0	11	18.60	66.104	0	6.393	12.655	20.145	36.625
3a	OVision	42	42 : 0	20	(21),1	0	37.80	53.973	0	1.561	1.965	3.655	12.195
3b	OVision	42	32 : 10	11	(20),9	10	21.60	54.320	0	1.679	2.195	4.285	14.135
3c	OVision	42	21 : 21	10	(10,0	21	19.80	53.519	0	1.668	2.135	3.865	13.595
3d	OVision	24	20 : 4	8	(11),3	4	16.20	30.245	0	1.361	1.575	1.835	8.085
3e	OVision	30	20 : 10	8	(11),3	10	16.20	38.143	0	1.442	1.715	2.225	8.025
3f	OVision	36	20 : 16	8	(11),3	16	16.20	46.225	0	1.571	1.935	3.085	12.855

Table 7.9: Measured Packet Delay and Loss Rate for the Video Sources across 5 LAN Switches and 4 Half-Duplex Switched Links.

Test	Source	Number of Flows admitted (L1, L2, L3, L4)	Topology Information					Measured Parameters					
			Number of Flows sent: *downstream : upstream*	Number of Flows sent from Nodes:			Cross-Switch Reservation in Mbit/s	High Priority Data Rate on L4 in Mbit/s	Pkt. Loss Rate in %	Ave. Delay in ms	90.0% in ms	99.0% in ms	Max. Delay in ms
				2, 4, 7, 10	3, 5, 8, 11	6, 9, 12,13							
4a	POO1	132	132 : 0	65	(66), 1	0	43.56	47.535	0	1.363	1.625	1.955	7.955
4b	POO1	132	102 : 30	50	(52), 1	30	33.66	47.105	0	1.383	1.655	2.015	7.815
4c	POO1	132	66 : 66	32	(33), 1	66	21.78	48.278	0	1.421	1.705	2.085	7.735
4d	POO1	60	50 : 10	20	(29), 9	10	12.60	20.536	0	1.217	1.355	1.535	7.995
4e	POO1	80	50 : 30	20	(29), 9	20	12.60	26.641	0	1.272	1.455	1.685	7.445
4f	POO1	100	50 : 50	20	(29), 9	50	12.60	36.593	0	1.326	1.545	1.815	7.665
5a	POO3	202	202 : 0	90	(111),21	0	44.44	66.786	0	5.048	16.415	38.375	65.755
5b	POO3	204	154 : 50	50	(103),53	50	22.44	67.738	0	7.174	23.855	48.485	88.405
5c	POO3	204	104 : 100	50	(53), 3	100	22.44	69.872	0	5.482	16.755	48.555	78.315
5d	POO3	141	101 : 40	50	(50), 0	40	22.44	46.478	0	2.214	1.855	24.835	52.875
5e	POO3	171	101 : 70	50	(50), 0	70	22.44	57.192	0	3.289	5.025	36.705	63.985
5f	POO3	201	101 : 100	50	(50), 0	100	22.44	69.536	0	6.129	19.325	54.125	88.015

Table 7.10: Measured Packet Delay and Loss Rate for the Pareto Sources across 5 LAN Switches and 4 Half-Duplex Switched Links.

Furthermore, our analysis of the buffer space requirements does not consider all the properties of the medium access. In reality, nodes are served in round-robin order by the network. In our test switches, all Controlled Load flows encountered FIFO queueing within the high priority output

queue. If we additionally consider the use of the worst-case results for the per-packet overhead and the interrupt time, then we can assume that the admission control will typically compute pessimistic bounds for the data throughput and the buffer space requirements within the network.

Table 7.9 and Table 7.10 contain the results measured in the 4HDL Test Network. Figure 7.20, Figure 7.21 and Figure 7.22 show the distribution function and the distribution density for selected MMC1 and POO3 tests. These represent the equivalent graphs to: (1) Figure 7.12, Figure 7.13 and Figure 7.14 in Section 7.3.3 (the 1L1S Test Network), and (2) Figure 7.16, Figure 7.17 and Figure 7.18 in Section 7.3.4 (the 1HDL Test Network).

The numerical results in Table 7.9 and Table 7.10 are organized in a similar way as the results discussed for the 1HDL Test Network. There are only a few minor differences which we clarify in the following. Column 3 shows the number of flows admitted on *each* of the four half-duplex switched links. The results are identical to those received for the single link (1HDL) test network. The columns 4 - 8 contain informations about the flow distribution in the network. This uses the same notation as explained for Table 7.7 and Table 7.8 in the previous section. In Test 1a (MMC2) for example, 7 flows entered the network at each of the nodes: 2, 4, 7 and 10. The setup additionally included 14 flows sent by node 3 and 7 flows generated at each of the nodes: 5, 8 and 11. The nodes: 6, 9, 12, 13 did not pass any flows into the test network in this experiment. Considering: (1) the single flow sent by the Measurement Client, and (2) the path information for the bridged test network in Figure 7.19, we have: 1 + 14 + 7 = 22 flows for link L1, and: 1 + 7 + 7 + 7 = 22 flows for the links: L2, L3 and L4. Switch Sw2 however only forwarded the 7 flows from node 2 and the single flow from the Measurement Client onto link L2. This resulted in a Cross Switch Reservation of: $8 \cdot 3.4$ Mbit/s $= 27.2$ Mbit/s from link L1 to L2. The same result is received for all other switches. It is thus listed in Column 8 in Table 7.9 and Table 7.10.

In contrast, Test 1f included: (1) 4 flows from each of the nodes: 2, 4, 7 and 10, (2) 5 flows from node 3 and 1 flow sent by each of the nodes: 5, 8, 11, and (3) 10 flows generated at each of the nodes: 6, 9, 12 and 13. This led to 20 flows on each half-duplex switched link in this experiment. For the Cross Switch Reservation, we receive: $(1 + 4) \cdot 3.4$ Mbit/s $= 17.0$ Mbit/s in this case.

The columns 9 - 14 show the results measured. We however omitted the results for the average packet size in Table 7.9 since these were basically identical to those in Table 7.7. As in the previous experiments, the measurement interval for each test was 30 minutes with an additional warm-up time of 2 minutes.

In all tests in the 4HDL test network, we did not detect a packet loss of a Controlled Load data packet in any of the four LAN switches. It is again likely that this was also the case for the high priority traffic forwarded upstream in the experiments because this traffic was never higher than the total traffic forwarded downstream. We could however not explicitly measure the loss characteristics for this traffic since our test LAN switches do not differentiate between high- and normal priority data packets dropped. Packet loss along the upstream data path however always occurred for best effort traffic which led to large results in the corresponding packet drop counters (*ifOutDiscards*).

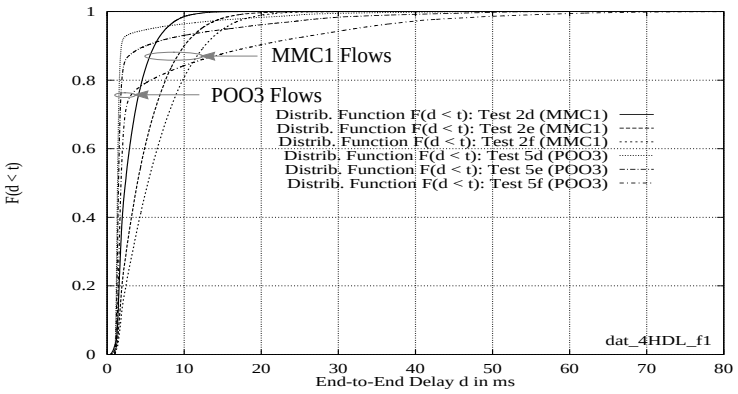


Figure 7.20: Distribution Function for Tests 2d - 2f (MMC1) and 5d - 5f (POO3) in the 4HDL Test Network.

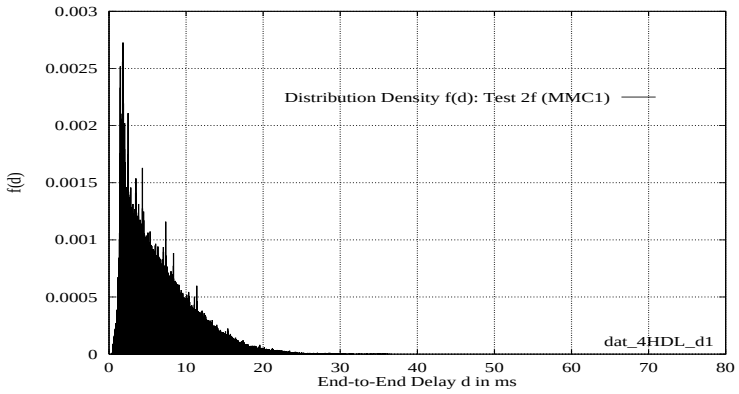


Figure 7.21: Distribution Density corresponding to Test 2f (MMC1, 4HDL Topology) in Figure 7.20.

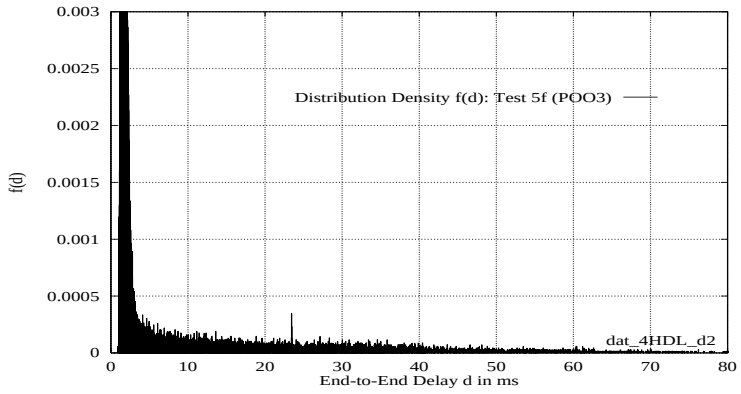


Figure 7.22: Distribution Density corresponding to Test 5f (POO3, 4HDL Topology) in Figure 7.20.

Looking at the results in Column 11, we can observe that in some tests, the average delay increased significantly (MMC2, MMC1, POO3) in comparison to the single link (1HDL) topology, whereas this did not occur to the same extent with other sources (OVision, POO1). We believe that the results are in a way comparable since the corresponding experiments were based on the same flow distributions on all links. The experimental setups in both test networks only differed slightly due to the cross traffic reservations and the smaller number of network nodes for each link in the 4HDL topology. The largest growth of the average delay can be observed in the POO3 tests (214 - 630%) even though the absolute results (2.2 - 7.2 ms) are still in the order of the values received for MMC1 (3.2 - 6.7 ms). For MMC1 and MMC2 we achieved growth rates of: 159 - 388% and 120 - 183%, respectively. In contrast, the results in the POO1 tests only increased by: 84 - 108%.

Even though the average delays received in the MMC1 and POO3 tests are similar, their distributions nevertheless differ significantly. The results for MMC1 in Figure 7.20 are only distributed over a short time range. We measured maximum delays of just: 19.8 - 36.6 ms. These are caused by the short range burst behaviour which we already identified for this trace in Section 4.2.1. Even though the MMC2 trace is burstier than the MMC1 trace, this did not have a substantial impact in the experiments. Setups including larger numbers of admitted flows e.g. caused by fewer resources allocated for each individual flow might however lead to higher delays and a different distribution.

In contrast to this, the results obtained for the POO3 tests exhibit a strong tail in their distribution with maximum delays of: 52.9 - 88.4 ms. Even for the 99.0 percentile we still have values between: 24.8 - 54.1 ms. Considering the distribution in Figure 7.22, one can expect even larger maxima when the experiments are repeated with longer measurement intervals than 30 minutes. A similar behaviour could already be observed for the single link test network in Figure 7.18. All recorded results are however still significantly smaller than the theoretical maximum (~184 ms) of the queuing delay in the 4HDL test topology. The maximum delays obtained with each of the other four traces are much smaller than the results achieved with the POO3 sources. The lowest values were measured in the POO1 tests (7.4 - 7.9 ms). In these tests, resources were however also reserved at peak data rate (tests a - c) or very close to the peak rate (tests d - f).

We further found that in all three test topologies, the results received for each source type exhibit the same fundamental characteristics (e.g. short range versus long range distribution). This can for example be observed in the graphs shown for MMC1 Test 2f (Figure 7.14, Figure 7.17, Figure 7.21) and POO3 Test 5f (Figure 7.14, Figure 7.18, Figure 7.22).

The impact of the network topology and the flow distribution on the service parameters is hard to quantify. We expected a more significant distribution of the results for the same source trace than those reported in Table 7.9 and Table 7.10. A conclusion other than this impact is difficult to predict, because of the dependencies between all test setup parameters, can not be drawn from our results. Some of the factors that influence the characteristics are: (1) the resources allocated for each flow, leading to a certain burstiness of the total traffic and to a certain average high priority data rate on the link, (2) the amount of traffic that traverses several switches, (3) the ratio of the flows forwarded

upstream and downstream in respect to the measurement traffic, (4) the number of network nodes with High Priority Traffic Clients connected to each switch, (5) speed mismatches between input and output links (not tested), or (5) the traffic characteristics and the data path of the best effort traffic. During the experiments performed in our test networks, we however found that although these dependencies changed the results, for the test traces, they basically never led to a significant increase of the average delays, persistent packet loss in the network, or consistently higher maximum delays than discussed in this section.

7.3.6 Resource Utilization

Beside a simple service discipline in switching nodes in the network, the low assurance level of the Controlled Load service typically also allows the admission of additional flows compared to the Guaranteed service. Table 7.11 and Table 7.12 show example results for the maximum high priority resource utilization achieved with the Controlled Load service. These were determined for a number of selected test applications in a single Level-1 and Level-2 cascaded network segment. The equivalent results for the same applications using the Guaranteed service can be found in Table 6.4 and Table 6.5.

To allow an accurate comparison, the admission control used the same input parameters as previously discussed in Section 6.5.4. The first four columns in Table 7.11 and Table 7.12 thus have the same contents as their equivalent in Table 6.4 and Table 6.5. They specify: (1) the size of the allocation time frame *TF*, (2) the test application, (3) the data rate *r* allocated for each flow on the segment, and (4) the packet count that was used to estimate the data transmission overhead. The results for the packet count imply a burst size of one maximum sized data packet ($\delta = P_{max}$) as assumed in the tests performed for the Guaranteed service. Even though we could have allocated larger burst sizes for all Controlled Load flows, we decided not to do so, since this had also led to different results for the packet count and would have made a comparison with the results in Table 6.4 and Table 6.5 more complicated. In all experiments, we again admitted homogeneous flows until we received a reject from the admission control. This was always caused by the Bandwidth Test (Theorem 7.1) due to the small burst size requested for each flow.

The results received from the admission control are shown in columns 5 - 8. This starts with the maximum number of flows successfully admitted to the service. Column 6 contains the amount of bandwidth that was allocated after all flows had been admitted. The corresponding maximum high priority network utilization is listed in Column 7. For the details of how this was computed, we refer to Section 6.5.4. The last column shows the difference of the utilizations received for the Controlled Load service and the Guaranteed service. For *vat* (*TF* = 10 ms, *r* = 0.075 Mbit/s) for example, we have a utilization gain of: 29.31% - 5.43% = 23.88% .

When we compare the results received for both services, we find that, as expected, higher resource utilizations could be achieved with the Controlled Load service. In the Level-1 cascaded network, the admission control was able to allocate between: 26.33 Mbit/s and 81.00 Mbit/s using a utiliza-

tion factor of: $f = 1$ in the admission control. This corresponds to resource utilizations of: 29.31% - 88.40%. In the Level-2 cascaded network, we still achieved utilizations between: 17.50% and 78.89%.

Time Frame *TF* in ms	Application	Per-Flow Data Rate allocated in Mbit/s	Packet Count (*pcnt*) measured	Max. Number of Flows admitted (N_{max})	Total Bandwidth allocated in Mbit/s	Maximum High Priority Network Utilization (%)	Utilization Gain in Comparison to the GS in %
10	vat	0.075	2	351	26.33	29.31	23.88
	nv	0.128	3	226	28.93	32.21	23.80
	vic	1.0	5	63	63.00	70.14	32.28
	OVision	1.8	7	37	66.60	74.15	26.05
	MMC	3.0	8	24	72.00	80.17	23.39
20	vat	0.075	4	356	26.70	29.33	20.10
	nv	0.128	4	298	38.14	41.91	27.14
	vic	1.0	6	74	74.00	81.30	27.47
	OVision	1.8	9	42	75.60	83.06	19.78
	MMC	3.0	11	26	78.00	85.69	16.48
40	vat	0.075	5	493	36.98	40.35	24.22
	nv	0.128	6	354	45.31	49.45	25.70
	vic	1.0	10	77	77.00	84.04	17.46
	OVision	1.8	16	43	77.40	84.47	11.78
	MMC	3.0	17	27	81.00	88.40	9.82

Table 7.11: Maximum High Priority Utilization using the Controlled Load Service in a Single Hub Network.

Time Frame *TF* in ms	Application	Per-Flow Data Rate allocated in Mbit/s	Packet Count (*pcnt*) measured	Max. Number of Flows admitted (N_{max})	Total Bandwidth allocated in Mbit/s	Maximum High Priority Network Utilization (%)	Utilization Gain in Comparison to the GS in %
10	vat	0.075	2	187	14.03	17.50	12.35
	nv	0.128	3	122	15.62	19.49	11.98
	vic	1.0	5	44	44.00	54.91	22.46
	OVision	1.8	7	26	46.80	58.40	17.97
	MMC	3.0	8	17	51.00	63.64	14.97
20	vat	0.075	4	193	14.48	17.55	9.64
	nv	0.128	4	174	22.27	27.00	14.12
	vic	1.0	6	57	57.00	69.10	20.61
	OVision	1.8	9	32	57.60	69.83	13.09
	MMC	3.0	11	20	60.00	72.74	10.91
40	vat	0.075	5	287	21.53	25.73	12.10
	nv	0.128	6	218	27.90	33.35	13.46
	vic	1.0	10	61	61.00	72.91	13.14
	OVision	1.8	16	34	61.20	73.15	8.60
	MMC	3.0	17	22	66.00	78.89	7.17

Table 7.12: Maximum High Priority Utilization using the Controlled Load Service in a Level-2 Cascaded Network.

The results obtained for the different applications and time frames exhibit similar characteristics as discussed for the results in Section 6.5.4: (1) lower utilizations are received for low bitrate flows (*vat*, *nv*), whereas we achieved higher results whenever higher-bitrate applications (*vic*, *OVision*, *MMC*) became admitted. (2) Higher utilizations can further be observed for all applications when larger time frames are used in the admission control.

In spite of the discussion in Section 6.5.4, a few additional comments can be made. In comparison to the Guaranteed service, utilization gains between: 9.8% and 32.28% were achieved for the Controlled Load service in the Level-1 cascaded network. For a time frame of 20 ms, these correspond to: 244 *vat*, 193 *nv*, 25 *vic*, 10 *OVision* and 5 *MMC* flows that could additionally be admitted to the service in each of the corresponding tests. For the Level-2 cascaded network, we received utilization gains between 7.17% and 22.46%. All results in Table 7.12 are however lower than those in Table 7.11 due to the lower bandwidth that is available for the resource reservation in a Level-2 cascaded network.

The gain that can be observed for all applications is mainly a result of the average rate allocation performed for the Controlled Load service. For low bitrate flows, the admission control does further not allocate a minimum resource share per time frame as it does to provide Guaranteed service. The maximum resource utilization is still dependent on the time frame (*TF)* because the estimation of the data transmission overhead is based on this parameter. The size of *TF* is however not as critical as in the admission control for the Guaranteed service since the Controlled Load service does not have to provide a delay bound. Larger time frames can thus be selected without a penalty other than the estimation of the network capacity is less pessimistic.

Furthermore, since the packet counts in all tests in Table 7.11 and Table 7.12 are measurement results determined with the Time Window algorithm, all results for the resource utilization include the overhead introduced by the packet count estimation process. As shown in Section 6.5.3, this overhead is significant for low bitrate flows, which explains the lower utilizations received for them in the tests.

The utilization gain is however not achieved without any costs. These are in the higher transmission delays encountered by data packets using the Controlled Load service. The differences can for example be observed by comparing the results in Table 6.2 (Guaranteed service across a Level-2 cascaded network segment) with those in Table 7.5 (Controlled Load service across a Level-1 cascaded segment). Note that both tables only show the results across a single segment. In bridged networks, it can be expected that the average delays provided by both services can hardly be detected by existing applications. The results for the maximum delays may however differ significantly. The Controlled Load service might further occasionally lose a data packet in the network.

7.4 Related Work

In this section, we summarize related work that can be used to enforce Controlled Load quality of service. None of the discussed approaches however was specifically designed for IEEE 802 type LANs (or Demand Priority networks). Instead, most of them assume an ATM network and can thus not easily be reused in shared, or half-duplex switched LANs.

There are many approaches based on the Effective Bandwidth concept. These will be discussed first. As remarked at the beginning of this chapter, the concept of the Effective Bandwidth includes the computation of the bandwidth requirement $C(\varepsilon)$ for a class of flows such that their stationary data arrival rate exceeds $C(\varepsilon)$ with a probability of not more than ε. More formally [GAN91]: $Prob(C(\varepsilon) < R_A) \leq \varepsilon$, where R_A denotes the aggregated data rate and ε the overflow probability. In [GAN91], [AS94], [DJM97] the data traffic arriving at an ATM switch is modelled as having a Normal Distribution. Assuming an average data rate of μ_A and a variance of σ_A^2, then an approximation for the Effective Bandwidth is for example given by [GAN91]: $C(\varepsilon) = \mu_A + \sigma_A \cdot \sqrt{-2\ln(\varepsilon) - \ln(2\pi)}$. The authors of [GAN91] however also remark that the Gaussian assumption does not hold for small numbers of very bursty flows, with high peak rates and long burst periods. This was also found in [AS94] when the arrival rate is approximated using a Poisson Distribution (see for example Fig. 2 and Fig. 3 therein).

In [Floy96], an upper bound on the Effective Bandwidth is derived using Hoeffding bounds. For a set of *n* flows with a peak rate of: R_P^i for each flow $i \in n$, and an average aggregated arrival rate of μ_A for all already admitted flows, the bound is computed using: $C(\varepsilon) = \mu_A + \sqrt{(\ln(1/\varepsilon) \cdot \sum_{i \in n}(R_P^i)^2)/2}$. This is used to enforce Controlled Load service. A new flow ν with the peak rate R_P^ν is admitted when the sum of the Effective Bandwidth $C(\varepsilon)$ estimated for all already admitted flows and the peak rate of the new flow does not exceed the allocation limit *B* of the Controlled Load service ($C(\varepsilon) + R_P^\nu \leq B$). The simulation results presented by Floyd in [Floy96] suggest that an approximation based on the Normal Distribution is generally more accurate than the results derived using Hoeffding bounds. The former however sometimes underestimates the bandwidth requirements which confirms the conclusions drawn in [GAN91]. Floyd further remarks that for traffic aggregations including only 10 flows, an estimation based on the concept of the Effective Bandwidth provides similar results as a peak data rate allocation. Significant statistical multiplexing gains were observed for classes with about 50 flows.

Other schemes can be found in: [KWC93], [GKK95], [GiKe97], [TG97], [Droz97]. The authors of [KWC93] derive the Effective Bandwidth for Markov fluid sources which characterize the traffic as a time-continuous, Markov modulated data stream. In [GKK95], Bayesian decision theory is applied to derive an acceptance threshold which is then used in the admission control. The approximations in [GiKe97] and [TG97] are based on the Chernoff bound. The author of [Droz97] proposes an algorithm that uses a Wavelet-based traffic estimation (similar to a Fourier analysis) to derive the Effective Bandwidth.

Many of the schemes listed so far are measurement-based since they also use input parameters which are estimated using on-line measurements. In [AS94] and [DJM97] for example, the average data rate and the variance of the number of ATM cells arriving within a time interval are measured and afterwards used for the modelling of the traffic distribution. The admission control decisions in [GKK95] and [Floy96] only use measurements of the average aggregated data rate of all flows already admitted. The algorithm additionally requires the peak data rates of all accepted flows which can however be derived from the parameters declared by the corresponding applications at reservation setup. The admission control conditions proposed in [TG97] are only based on measurement information.

In [DLC+95], [CLL+95], [CLH+95], it was shown that the modelling of the data arrival process can be by-passed by measuring an approximation of the large deviation rate function. This was performed based on the observation that, for a single server queue that is served with a constant service rate, the queue-length distribution of the traffic passed through this queue is of the form [DLC+95], [CLH+95]: $Prob(Q > q) \sim \exp -\delta q$, where the slope δ of the distribution is given by the rate function *I(.)*. The parameters Q and q denote the buffer space and the queue length, respectively. Instead of estimating the rate function, it is however more convenient to estimate a transform of it called the *Scaled Cumulant Generating Function* (CGF): $\lambda(\theta)$. The decay parameter δ can then be determined directly from $\lambda(\theta)$ using the relation [DLC+95]: $\delta = sup\{\theta: \lambda(\theta) \leq s \cdot \theta\}$, where s denotes the service rate. To compute the CGF, the authors measured the amount of data passed into the output queue in subsequent time blocks of constant size *T*. Assuming *n* samples and block sums X_i, where $i \in n$, then leads to [DLC+95]:

$$\lambda(\theta) = \frac{1}{T} \cdot \log \frac{T}{n} \sum_{i=1}^{\lfloor n/T \rfloor} e^{\theta X_i} \tag{7.55}$$

for the CGF of the empirical distribution of the block sums. This can be used as an estimator. Experimental results are provided in [CLL+95], [CLH+95]. The main practical difficulties are in determining (1) the block sizes T such that the block sums X_i are independent and identically distributed, and (2) the minimum number of samples required to achieve an accurate estimate. Furthermore, the approach is based on the assumptions that the data arrival rate is stationary (no rate shifts) and does not include long-range dependencies. Other schemes based on Large Deviation Theory can for example be found in: [CT95] and [VeSo97].

Research has also been performed on alternative approaches which do not use the Effective Bandwidth concept. [JDSZ95], [Jami96] propose admission control conditions for Predictive service which could however also be used to provide Controlled Load quality of service. The scheme is based on the CSZ scheduler [CSZ92] which we described in Section 2.3.2. In the following discussion, we focus on the admission control for the Predictive service and neglect the resources allocated for the Guaranteed service. The algorithm is based on the Simple Sum approach, but

additionally uses measurements to increase the resource utilization. Measured are (1) the aggregated load $\hat{\upsilon}$ of all already admitted Predictive service flows, and (2) the delay $\hat{D}$ experienced by the corresponding data packets in the output queue of the switch. A new flow with the token bucket parameters: δ, r is admitted when: (1) the sum of the flow's data rate r and the current load estimate $\hat{\upsilon}$ is lower than the allocation limit B for the service: $r + \hat{\upsilon} < B$, and (2) the admission of the new flow does not violate the delay bound D, where: $D > \hat{D} + \delta / C_l$. The parameter C_l denotes the bandwidth of the outgoing link. The measured parameters: $\hat{\upsilon}$ and $\hat{D}$ are estimated using a fixed-size window algorithm. The authors of [CKT96] extended this work by proposing an adaptive window-algorithm for the parameter estimation. Furthermore, simulation results for a Controlled Load service using an admission control that is only based on the above bandwidth test ($r + \hat{\upsilon} < B$) can be found in [JSD97].

7.5 Summary

In this chapter, we showed how Controlled Load quality of service can be enforced across shared and half-duplex switched Demand Priority networks. We first defined the packet scheduling process and derived the corresponding admission control conditions. The second part included a performance evaluation of the new service.

In contrast to other algorithms whose design aimed at high results for the resource utilization, we focused on simplicity to ensure the lowest possible costs for LAN switches. This was achieved by building the service based on: (1) a simple static priority scheduler in switches, and (2) traffic policing and reshaping mechanisms deployed only at the entrance to the bridged network. The access to the service is restricted by admission control. For this, we used a Simple Sum style approach based on an average rate allocation for all Controlled Load service flows. The Bandwidth Test proves Stability and additionally enables a network administrator to enforce a minimum resource share for the Best Effort service. The test directly follows from the Bandwidth Test derived for the Guaranteed service and thus also considers the Demand Priority protocol overhead. The Buffer Space Test was derived by applying the analysis techniques developed by Cruz in [Cruz91a]. Our calculus however differs by considering: (1) a shared network model, (2) a variable data throughput as can be found in Demand Priority networks, and (3) a minimum guaranteed service rate for each node enforced by the round-robin packet service policy of the network.

The simplicity of the service discipline however also enables strong interactions between different flows using the Controlled Load service in the network. The admission control considers this by reserving additional buffer space in the network. In the performance evaluation, we found that the impact of the cross traffic characteristics on the buffer space requirements of a flow may be significant when the bandwidth allocated for the corresponding network node is higher than its "Fair Bandwidth Share". The highest results were achieved when we reserved large capacities for a single node in a cascaded network that already included a multitude of network nodes with previously accepted reservations. Requests including large bandwidth requirements and burst sizes will thus

have a lower probability of being accepted when the data path contains segments with large cross traffic reservations. In contrast, the impact of the cross traffic is negligible as long as the allocated resources remain below the fair share limit. In the results received for a half-duplex switched link for example, we could observe a large bandwidth region (see Figure 7.8) where the buffer space requirements were completely independent from the burstiness of the cross traffic sent by other nodes on the link.

One general problem we encountered was to select appropriate token bucket parameters (δ, r) for bursty test flows such that the delay in the rate regulator remained low and network resources were not wasted. During the experiments in our test networks, we found that determining the "optimum" typically required several initial tests before the actual measurement. Note that in contrast to the common case, in our experiments the "optimum" denoted the case with the highest queueing delay measured in the network. For the final measurements, we thus always selected token bucket parameters close to the minimum flow requirements such that a large number of flows could be admitted. This typically led to the highest delays. In a real LAN however, finding an appropriate parameter set may be difficult since: (1) initial tests can typically not be performed, or (2) the reservation needs to be made for a flow (e.g. a video source using data compression) whose characteristics are determined by the contents of the data. If the requirements cannot accurately be characterized then overallocating resources in the network is probably inevitable, which we believe is however an acceptable policy in a LAN environment.

To test the Controlled Load service, we performed 30 experiments in each of our three test networks. Each measurement lasted 30 minutes and included a reservation at the allocation limit. In spite of the variety of the test setups including different: (1) network topologies, (2) cross traffic flow distributions, and (3) flow characteristics, the results for the end-to-end packet delay and the packet loss rate reported in this chapter have shown that even with a utilization factor of: $f = 1$ and experimental setups including only data sources with long range dependencies, Controlled Load type service guarantees can be provided by the network. In none of the above experiments, we detected the loss of a single data packet transmitted with the Controlled Load service. Packet loss was only observed for POO3 Pareto sources using an equivalent setup but a significantly longer measurement interval (several hours).

The worst results measured for the average end-to-end delay in the bridged 4HDL Test Network are in the order of a few milliseconds. The corresponding maximum delay and the 99.0 percentile of it may however be significantly higher. These mainly depended on the total burstiness of the traffic and the average high priority network load during the test. It was thus not surprising that the maximas were achieved with POO3 sources because of: (1) the large capacity allocated (maximum: 89.76 Mbit/s) in these tests, (2) the infinite variance of the Pareto sources used, and (3) the shared medium access on each half-duplex switched link in the data path. The latter property may basically lead to maximum delays which are twice as high as those that can be achieved in equivalent experiments including only full-duplex switched links in the data path.

Beside the simpler service discipline used in LAN switches, the Controlled Load service further enables higher maximum resource utilizations compared to the Guaranteed service. We achieved maximum results between: 17.50 Mbit/s and 88.40 Mbit/s, and utilization gains between: 7.17% and 32.28% on a single Level-1- and Level-2 cascaded network segment. These are basically the result of the average rate allocation performed for the Controlled Load service. It remains to emphasize that the resource allocation scheme developed in this chapter was based almost exclusively on pessimistic assumptions. A number of optimizations could thus be explored to increase the resource utilization. One is to use optimistic results for the Demand Priority overhead in the admission control. These could for example be determined by using a less conservative estimation approach than embedded in the Time Window algorithm, or could be based on heuristics when the characteristics of the packet size distribution in the LAN are known. Furthermore, instead of using the concept of the General Multiplexer for LAN switches, more accurate admission control conditions can be derived by taking the specific properties of the medium access or additionally topology information into account. Considering more detailed informations such as: (1) FIFO queueing for all Controlled Load flows, (2) the number of input ports with reservations, and (3) data arrival rates in the analysis will lead to tighter bounds which then enable a higher maximum resource utilization. This however also increases the complexity of the calculus and the probability of buffer overflow in the network.

Chapter 8
Summary and Future Work

8.1 Thesis Summary

In this dissertation we have proved that advanced packet delivery services, in particular the Guaranteed- and the Controlled Load service standardized for a future multi-service Internet can be provided across multi-hub shared and half-duplex switched Demand Priority LANs. The differentiator to the traditional Best Effort service deployed today is the quality of service which is assured by these new services for data packets sent across the network.

Chapter 1 introduced the research area and defined the hypothesis of this dissertation. We first discussed the potential advantages of multi-service networks and identified the packet switching approach as an efficient way of implementing these services. To be able to offer deterministic service guarantees in a packet switching network, a proactive congestion control scheme is needed. Furthermore, end-to-end service guarantees can only be supplied when the service is supported on all intermediate links including LANs within the data path. The research goal of the thesis was to show that Guaranteed- and Controlled Load quality of service can be enforced across shared Demand Priority LANs even when the network is highly utilized or becomes overloaded with Best-Effort traffic. This was achieved by applying admission control and differentiating data packets in the network. Our research was based on two methods: a theoretical analysis and experimental measurements in a test network. The analytical approach was chosen to analyse network performance parameters and to derive the admission control conditions. Measurements were performed to confirm the analytical results and to examine the quality of the new services.

Chapter 2 described the framework for our research. This is the ISPN architecture that has been proposed by the IETF to provide Integrated Services across the Internet. The architecture has three key components which we studied in this chapter: (1) the Integrated Services e.g. the Guaranteed- and the Controlled Load service, (2) the traffic control including the service discipline and the admission control, and (3) the reservation management. In contrast to the Integrated Services, the service discipline and the admission control do not become standardized. While looking at existing solutions proposed for WANs, we argued that most of them would show a poor performance when used in shared medium LANs. Popular approaches based on a sorted priority queue algorithm such as e.g. Weighted Fair Queueing cannot be applied at all. Our particular attention was given to the ISSLL framework for reserving resources across IEEE 802 style LANs. Unlike the mechanisms required in WANs, this framework allows a wide range of mechanisms to built Integrated Services in LANs. This enabled us to make design trade-offs between complexity and efficiency.

Chapter 3 described the measurement methods which we applied in this thesis. To generate realistic traffic patterns in the test network, we used a traffic trace driven approach. The traces required for this were obtained by: (1) monitoring the data output of selected multimedia applications in our test network, and (2) computing them based on Pareto traffic models. The accuracy of the approach was determined by the 1 ms timer granularity of the Traffic Generator. For a single 1.286 Mbit/s MPEG encoded video stream for example, we measured that 99.0 percent of all packet interarrival times differed by an absolute value of less than 0.85 ms from the original trace. This further decreased for traces with a peak to average rate ratio close to the network capacity. For a Pareto source with a peak to average ratio of 90, we observed a 99.0 percentile of just 0.35 ms despite the timer granularity of 1 ms. To measure the data throughput and packet loss rate in the test network, we exploited the standard MIB counters. The end-to-end delay was determined based on a centralistic approach in which the start and finish time of each measurement were taken by the same workstation. The accuracy of the measurement approach enabled us to clearly distinguish the transmission time for a single maximum sized data packet which is equivalent to 120 μs.

In Chapter 4, we investigated the performance characteristics of 802.12 networks in respect to the bandwidth, the packet delay and the packet loss rate encountered by data flows in the network. We could first observe that the data throughput in Demand Priority networks is variable and may significantly decrease for data transmissions that only use small sized packets. In a single hub test network, we measured a performance loss of over 60% for this. Beside the packet size, the data throughput further depends on the topology, in particular the cascading level of the network. The maximum throughput measured for example in a Level-4 cascaded network for data packets of 100 bytes was as low as 17.93 Mbit/s. We further observed a low average packet delay over a load range of over 60 Mbit/s. This was exploited in Chapter 7 to provide Controlled Load quality of service. Packet loss may however occur with an average delay in the order of a few milliseconds. This suggests that traffic control mechanisms within LANs should attempt to control the maximum delay and the packet loss rate instead of the average delay which will be low provided that packet loss can be avoided. Our experiments further showed that additional buffer space within LAN switches improves the loss behaviour. Depending on the traffic characteristics, it may however be impossible or require a substantial amount of memory to completely eliminate packet loss in the network. After the analysis, we discussed several approaches to provide quality of service within LANs and identified low costs as a design goal for our resource reservation schemes introduced in Chapter 6 and Chapter 7.

In Chapter 5, we analysed the details of the data transmission in 802.12 networks and derived worst-case bounds for the signalling overhead. We first observed that the service properties enforced by the Demand Priority protocol are maintained in multi-hub networks and half-duplex switched links. This enabled us to use the same packet scheduling process and the same admission control conditions for all 802.12 network topologies. To describe the signalling overhead, we identified two specific network parameters: (1) the per-packet overhead, and (2) the time it takes to interrupt the normal priority service. Analytical results for both parameters were derived for a UTP and a fibre-

optic physical layer. Based on these results, our admission control was able to accurately determine the minimum available bandwidth in the network. This is essential to provide deterministic service guarantees as required for the Guaranteed service.

Chapter 6 proposed a resource allocation scheme which can be used to provide a Guaranteed service across shared multi-hub and half-duplex switched Demand Priority networks. We defined the packet scheduling process in the network and derived the corresponding admission control conditions which bind the worst-case packet delay. The scheme is based on a time frame concept and was built on top of the 802.12 high priority medium access mechanism. Small delay bounds can be guaranteed by using admission control. Our approach differs from others by: (1) the consideration of the Demand Priority overhead in the admission control, and (2) the meaning of the time frame. In our scheme, the time frame is an upper bound for the queueing and the propagation delay for all data packets using the Guaranteed service. Furthermore, we showed that it is not necessarily the minimum delay bound that can be provided for a node in the network. All other approaches known to us simply ignored the Demand Priority overhead despite its significant impact on the data throughput which we could observe in Chapter 4. Furthermore, in allocation schemes designed for other LAN technologies, the time frame often bounds the medium access time and is thus more comparable with the normal priority service interrupt time in our scheme. Results of experiments performed on test networks with a UTP physical layer and different topologies showed that our network model and the admission control conditions derived from it were accurate. The highest accuracy was found for the single hub network. This decreased for higher cascading levels due to the worst-case assumptions made in our model. To enforce Guaranteed quality of service in bridged networks, our scheme depends on rate regulators and a static priority scheduler within LAN switches. Compared to the traditional FIFO service discipline, this significantly increased the complexity of LAN switches, but was required to ensure a deterministic delay bound and an acceptable level of efficiency. Simplicity and low cost are however maintained in unbridged multi-hub networks since hubs do not have to identify or isolate single flows. The low flow isolation capabilities of the network and the consideration of worst-case conditions in the admission control may lead to a low resource efficiency. For a single hub network for example, we received results between 5.43% and 78.58% for the maximum high priority resource utilization. These might however be acceptable, since any resources allocated but unused can immediately be used to serve Best Effort data packets.

Chapter 7 showed how Controlled Load quality of service can be enforced in Demand Priority networks. This only requires a simple static priority scheduler with two priority levels in LAN switches. Controlled Load data traffic is only policed at the entrance of the bridged network but not within switches. Our approach thus differs significantly from the wide area network model in [BCS94] which requires traffic control mechanisms at each router, but still fits into the ISSLL framework. For admission control, we used a parameter based approach. Unlike the Guaranteed service, which is based on a peak rate allocation, the conditions derived in this chapter also allow the allocation of average data rates for all flows using the Controlled Load service. Furthermore, we consider a shared network model, variable packet sizes and a variable service rate as found in

Demand Priority networks. In contrast to this, existing solutions almost always assume a simple network model including point-to-point links and a constant data throughput as for example provided by ATM. They can thus typically not easily be applied to our environment. To test our scheme, we performed a variety of experiments in different test networks which showed that Controlled Load quality of service can be enforced even when the network is fully loaded, resources are allocated to the allocation limit and the admitted traffic has long range dependencies. During the experiments reported in this chapter we never observed the loss of a single data packet transmitted with the Controlled Load service. We conclude that packet loss in the network will be extremely rare, in particular with more realistic Resource Utilization Factors of: $f < 1$. We also showed that the maximum delays in bridged network can be large when the data sources are bursty over long time scales. In contrast, results measured for the average delays always remained in the order of a few milliseconds which is for example sufficient to support existing time sensitive, but adaptive and loss tolerant applications. We further observed that the delay characteristics are determined by a number of characteristics such as the network topology, the cross traffic characteristics and the flow distribution. We could however not identify a single setup which always led to a significant increase of the average delay, persistent packet loss or consistently higher results for the maximum delay.

When we compare the resource reservation schemes in Chapter 6 and Chapter 7 with other solutions providing Guaranteed- and Controlled Load quality of service, then we find that the main advantages of our approaches are their simplicity and applicability. Both schemes were built on top of the 802.12 high priority medium access mechanism such that no changes to the existing standard are required. The Controlled Load service only depends on a simple static priority scheduler in LAN switches. This will ensure low implementation costs. Furthermore, static priority schedulers will be available in many next generation switch products. Some LAN switches like the one which we used in our experiments can even support it today. The Controlled Load service can thus immediately be deployed, provided network nodes are able to rate regulate flows. For this, only those nodes which use the 802.12 high priority medium access mechanism need to be updated. Our host implementation has however shown that the required traffic control mechanisms can be implemented in software. A solution could thus be distributed as part of a device driver update. Alternatively rate regulators might also become implemented in hardware on LAN adapter cards. For the deployment of the Guaranteed service, basically the same constraints arise if we assume that this service is only used in unbridged multi-hub networks. The support across different network segments requires new LAN switches which are however currently not available.

8.2 Areas for Future Work

Although we believe that we studied the subject in much detail, there are several areas that could still be explored further. First, the Time Window algorithm, though sufficient to prove the overall concept, is rather too simple since it provides a poor estimate for low bitrate flows. More accurate results could probably be achieved by using a more sophisticated estimation process. An adaptive window algorithm similar to the one proposed in [CKT96] for a measurement based admission con-

trol might also be beneficial. Second, the Buffer Space Test of the Controlled Load service could be improved by considering additional network information in the calculus. This was discussed in detail in Chapter 7. Third, we showed that the Controlled Load service enables higher results for the maximum high priority link utilization compared to the Guaranteed service. The average load may nevertheless be low when the traffic is bursty and resources are allocated close to the peak data rate. This could for example be improved by using a measurement based approach. We refer to the related work section in Chapter 7 for possible ideas that could be exploited. We however argue that there is no stringent need for such a scheme because we believe that a large fraction of the traffic in future LANs will still be transmitted using the Best Effort service. Fourth, this thesis only focused on Demand Priority networks. It might however be possible to re-use some of the concepts to enforce quality of service in networks using a different medium access mechanism. In particular, we looked at a Controlled Load service across half-duplex Ethernet links where one of the two nodes on the link used the standard back-off algorithm, whereas the algorithm of the other node was modified. This however requires further research.

Finally, we hope that our work has contributed to a better understanding of the design trade-offs and costs required to provide Guaranteed- and Controlled Load quality of service across shared multi-hub and half-duplex switched Demand Priority networks.

Peter Kim, September 1998

Bibliography

[ACZ92] G. Agrawal, B. Chen, W. Zhao, *Guaranteeing Synchronous Message Deadlines with the Timed Token Protocol*, in Proc. of IEEE Conference on Distributed Computing Systems, pp. 468 - 475, Yokohama, June 1992.

[ACZ93] G. Agrawal, B. Chen, W. Zhao, *Local Synchronous Capacity Allocation Schemes for Guaranteeing Message Deadlines with the timed Token Protocol*, in Proc. of IEEE INFOCOM‘93, pp.186 - 193, San Francisco, March 1993.

[ACZD94] G. Agrawal, B. Chen, W. Zhao, S. Davari, *Guaranteeing Synchronous Message Deadlines with the Timed Token Medium Acess Control Protocol*, in IEEE Transactions on Computers, Vol. 43, No. 3, pp. 327 - 339, March 1994.

[AS94] S. Abe, T. Soumiya, *A Traffic Control Method for Service Quality Assurance in an ATM Network*, in IEEE Journal on Selected Areas in Communications, Vol. 12, No. 2, pp. 322 - 331, February 1994.

[Atki95] R. Atkinson, *IP Authentication Header*, Internet Engineering Task Force, RFC 1826, August 1995.

[BaOf98] M. Baldi, Y. Ofek, *End-to-end Delay of Videoconferencing over Packet Switched Networks*, in Proc. of IEEE INFOCOM’98, pp. 1084 - 1092, San Francisco, March 1998.

[BBB+98] Y. Bernet, J. Binder, S. Blake, M. Carlson, E. Davies, B. Ohlman, D. Verma, Z. Wang, W. Weiss, *A Framework for Differential Services*, Internet Engineering Task Force, Internet Draft draft-ietf-diffserv-framework-00.txt, May 1998.

[BBC+98] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss, *An Architecture for Differentiated Services*, Internet Engineering Task Force, Internet Draft draft-ietf-diffserv-arch-01.txt, August 1998.

[BCS94] B. Braden, D. Clark, S. Shenker, *Integrated Services in the Internet Architecture: An Overview*, Internet Engineering Task Force, RFC 1633, June 1994.

[BeZh96a] J. C. R. Bennett, H. Zhang, *WF^2Q: Worst-case Fair Weighted Fair Queueing*, in Proc. of INFOCOM’96, pp. 120 - 128, San Franciso, March 1996.

[BeZh96b] J. C. R. Bennett, H. Zhang, *Hierarchical Packet Fair Queueing Algorithms*, in Proc. of ACM SIGCOMM’96, pp. 143 - 156, Stanford University, August 1996.

[BFM+96] A. Banerjea, D. Ferrari, B. A. Mah, M. Moran, D. C. Verma, H. Zhang, *The Tenet Real-Time Protocol Suite: Design, Implementation, and Experiences*, in IEEE Transactions on Networking, Vol. 4, No. 1, pp. 1 - 10, February 1996.

[BGK96] F. Baker, R. Guerin, D. Kandlur, *Specification of Committed Rate Quality of Service*, Internet Engineering Task Force, Internet Draft draft-ietf-intserv-commit-rate-srv-00.txt, June 1996.

[BKS97a] F. Baker, J. Krawczyk, A. Sastry, *RSVP Management Information Base using SMIv2*, Internet Engineering Task Force, RFC 2206, September 1997.

[BKS97b] F. Baker, J. Krawczyk, A. Sastry, *Integrated Services Management Information Base using SMIv2*, Internet Engineering Task Force, RFC 2213, September 1997.

[BKS97c] F. Baker, J. Krawczyk, A. Sastry, *Integrated Services Management Information Base Guaranteed Service Extensions using SMIv2*, Internet Engineering Task Force, RFC 2214, September 1997.

[BOP94] L. S. Brakmo, S. W. O'Malley, L. Peterson, *TCP Vegas: New Techniques for Congestion Detection and Avoidance*, in Proc. of ACM SIGCOMM'94, pp. 24 - 35, London, August 1994.

[BPSW95] C. Bisdikian, B. Patel, F. Schaffa, M. Willebeek-LeMair, *The Use of Priorities on Token-Ring Networks for Multimedia Traffic*, in IEEE Network, Vol. 9, No. 6, pp. 28 - 37, December 1995.

[Brad89] B. Braden (Editor Network Working Group), *Requirements for Internet Hosts - Communication Layers*, Internet Engineering Task Force, RFC 1122, October 1989.

[BTW94] J. Bolot, T. Turletti, I. Wakeman, *Scalable Feedback Control for Multicast Video Distribution in the Internet*, in Proc. of ACM SIGCOMM'94, pp. 58 - 67, London, August 1994.

[BZB+97] B. Braden, L. Zhang, S. Berson, S. Herzog, S. Jamin, *Resource ReSerVation Protocol (RSVP) - Version 1 Functional Specification*, Internet Engineering Task Force, RFC 2205, September 1997.

[CFSD90] J. Case, M. Fedor, M. Schoffstall, C. Davin, *Simple Network Management Protocol (SNMP)*, RFC 1157, May 1990.

[ChNa97] H. Chen, K. Nahrstedt, *VGAnet: A Real-Time Transport Protocol Suite for 100VG-Any LAN*, Technical Report UIUCDCS-R-97-2025, University of Illinois, August 1997.

[ChNa98] H. Chen, K. Nahrstedt, *QoS-aware Traffic Shaper for 100VG-Any LAN*, in Proc. of IEEE International Conference on Communications (ICC'98), Atlanta, June 1998.

[CKT96] C. Casetti, J. Kurose, D. Towsley, *An Adaptive Algorithm for Measurement-based Admission Control in Integrated Services Packet Networks*, in Proc. of Workshop for Protocols for High Speed Networks, October 1996.

[CLH+95] S. Crosby, I. Leslie, M. Huggard, J. Lewis, F. Toomey, C. Walsh, *Bypassing Modelling: Further Investigations of Entropy as a Traffic Descriptor in the Fairisle ATM network*, in Proc. of First Workshop on ATM Traffic Management (WATM'95), ENST, Paris, December 1995.

[CLL+95] S. Crosby, I. Leslie, J. Lewis, N. O'Connell, R. Russell, F. Toomey, *Bypassing Modelling: an Investigation of Entropy as a Traffic Descriptor in the Fairisle ATM network*, in Proc. of 12th U.K. Teletraffic Symposium (UKTS 95), February 1995.

[Claf94] K. Claffy, *Internet Traffic Characterization*, PhD Thesis, University of California, 1994.

[Clar95] D. Clark, *Reservations, Service Quality and Equality*, INFOCOM'95 Panel Discussion, Presentation Slides at: ftp://ftp.parc.xerox.com/pub/net-research/infocom95.html/, April 1995.

[Cruz91a] R. Cruz, *A Calculus for Network Delay, Part I: Network Elements in Isolation*, in IEEE Transactions on Information Theory, Vol. 37, No. 1, pp. 114 - 131, January 1991.

[Cruz91b] R. Cruz, *A Calculus for Network Delay, Part II: Network Analysis*, in IEEE Transactions on Information Theory, Vol. 37, No. 1, pp. 132 - 14, January 1991.

[CSZ92] D. Clark, S.Shenker, L. Zhang, *Supporting Real-Time Applications in an Integrated Services Packet Network: Architecture and Mechanism*, in Proc. of ACM SIGCOMM'92, pp.14 - 26, Baltimore, August 1992.

[CT95] C. Chang, J. Thomas, *Effective Bandwidth in High-Speed Digital Networks*, in IEEE Journal on Selected Areas in Communications, Vol. 13, No. 6, pp. 1091 - 1100, August 1995.

[DeBe95] L. Delgrossi, L. Berger (Editors ST2 Working Group), *Internet Stream Protocol Version 2 (ST2) - Protocol Specification - Version ST2+*, Internet Engineering Task Force, RFC 1819, August 1995.

[Deer95] S. Deering, *Reservations or No Reservations ?*, INFOCOM'95 Panel Discussion, Presentation Slides at: ftp://ftp.parc.xerox.com/pub/net-research/infocom95.html/, April 1995.

[DeTr97] M. Decina, V. Trecordi, *Convergence of Telecommunications and Computing to Networking Models for Integrated Services and Applications*, in Procedings of the IEEE, Vol. 85, No. 12, December 1997.

[DJM97] Z. Dziong, M. Juda, L. Mason, *A Framework for Bandwidth Management in ATM Networks - Aggregate Equivalent Bandwidth Estimation Approach*, in IEEE Journal on Selected Areas of Communications, Vol. 5, No. 1, pp. 134 - 147, February 1997.

[DKS89] A. Demers, S. Keshav, S. Shenker, *Analysis and Simulation of a Fair Queuing Algorithm*, in Proc. of ACM SIGCOMM'89, pp. 1 - 12, Austin - Texas, September 1989.

[DLC+95] N. G. Duffield, J. T. Lewis, N. O'Connell, R. Russell, F. Toomey, *Entropy of ATM Traffic Streams: A Tool for Estimating QoS Parameters*, in IEEE Journal on Selected Areas of Communications, Vol. 13, No. 6, pp. 981 - 990, August 1995.

[Droz97] P. Droz, *Wavelet-Based Resource Allocation in ATM Networks*, in Proc. of IEEE Global Telecommunications Conference (Globecom), pp. 833 - 837, Phoenix, Arizona, November 1997.

[Erik94] H. Eriksson, *MBONE: The Multicast Backbone*, in Communications of the ACM, Vol. 37, No. 8, pp. 54 - 60, August 1994.

[FBZ92] D. Ferrari, A. Banerjea, H. Zhang, *Network Support For Multimedia: A Discussion of the Tenet Approach*, Technical Report TR-92-072, University of California at Berkeley, November 1992.

[Ferr90] D. Ferrari, *Client Requirements for Real-time Communication Services*, in IEEE Communications Magazine, Vol. 28, No. 11, pp. 65 - 72, November 1990.

[Ferr95] D. Ferrari, *Reservations or No Reservations*, INFOCOM'95 Panel Discussion, Presentation Slides at: ftp://ftp.parc.xerox.com/pub/net-research/infocom95.html/, April 1995.

[FeVe90] D. Ferrari, D. Verma, *A Scheme for Real-Time Channel Establishment in Wide-Area Networks*, in IEEE Journal on Selected Areas of Communications, Vol. 8, No. 3, pp. 368 - 379, April 1990.

[FiPa95] N. R. Figueira, J. Pasquale, *An Upper Bound on Delay for the VirtualClock Service Discipline*, in IEEE Transactions on Networking, Vol. 3 No. 4, pp. 399 - 408, August 1995.

[Flic96] J. Flick, *Definitions of Managed Objects for IEEE 802.12 Interfaces*, Internet Engineering Task Force, RFC 2020, October 1996.

[FlJa93] S. Floyd, V. Jacobson, *Random Early Detection Gateways for Congestion Avoidance*, in IEEE Transactions on Networking, Vol. 1, No. 4, pp. 397 - 413, August 1993.

[FlJa95] S. Floyd, V. Jacobson, *Link-sharing and Resource Management Models for Packet Networks*, in IEEE Transactions on Networking, Vol. 3, No. 4, pp. 365 - 386, August 1995.

[Floy96] S. Floyd, *Comments on Measurement-based Admission Control for Controlled Load Services*, draft submitted to ACM Computer Communication Review, (also available at: http://www-nrg.ee.lbl.gov/floyd/), July 1996.

[FoLe91] H. J. Fowler, W. E. Leland, *Local Area Network Traffic Characteristics, with Implications for Broadband Network Congestion Management*, in IEEE Journal on Selected Areas in Communications, Vol. 9, No. 7, pp. 1139 - 1149, September 1991.

[Fred94] R. Fredrick, *Experiences with real-time software video compresssion*, in Proc. of Sixth International Workshop on Packet Video, Portland, September 1994.

[G114_96] International Telecommunication Unit (ITU-T), Transmission Systems and Media, General Characteristics of International Telephone Connections and International Telephone Circuits, *Recommendation G.114 - One-Way Transmission Time*, February 1996.

[GaDi97] L. Gautier, C. Diot, *MiMaze, a Multiuser Game on the Internet*, Technical Report 3248, INRIA, Sophia Antopolis, September 1997.

[GAN91] R. Guerin, H. Ahmadi, M. Naghshineh, *Equivalent Capacity and Its Application to Bandwidth Allocation in High-Speed Networks*, in IEEE Journal on Selected Areas in Communications, Vol. 9, No. 7, pp. 968 - 981, September 1991.

[Garr96] M. W. Garrett, *A Service Architecture for ATM: From Applications to Scheduling*, in IEEE Network, Vol. 10, No. 3, pp. 6 - 14, May/June 1996.

[GaWi94] M. W. Garrett, W. Willinger, *Analysis, Modelling and Generation of Self-Similar VBR Video Traffic*, in Proc. of ACM SIGCOMM'94, pp. 269 - 280, London, August 1994.

[GGPS96] L. Georgiadis, R. Guerin, V. Peris, K. Sivarajan, *Efficient Network QoS Provisioning Based on per Node Traffic Shaping*, in IEEE Transactions on Networking, Vol. 4, No. 4, pp. 482 - 501, August 1996.

[GiKe97] R. Gibbens, F. Kelly, *Measurement-based connection admission control*, in Proc. of 15th International Teletraffic Congress, June 1997.

[GKK95] R. Gibbens, F. Kelly, P. Key, *A Decision-Theoretic Approach to Call Admission Control in ATM Networks*, in IEEE Journal on Selected Areas in Communications, Vol. 13, No. 6, pp. 1101 - 1113, August 1995.

[Gole90] S. J. Golestani, *Stop-and-Go Queueing Framework for Congestion Management*, in Proc. of ACM SIGCOMM'90, pp. 8 - 18, Philadelphia Pennsylvania, September 1990.

[Gole91] S. J. Golestani, *Duration-Limited Statistical Multiplexing of Delay-Sensitive Traffic in Packet Networks*, in Proc. of IEEE INFOCOM'91, pp. 323 - 332, Bal Harbor, April 1991.

[Gole94] S. J. Golestani, *A Self-Clocked Fair Queueing Scheme for Broadband Applications*, in Proc. of IEEE INFOCOM'94, pp. 636 - 646, Toronto, June 1994.

[GPS+98] A. Ghanwani, J. W. Pace, V. Srinivasan, A. Smith, M. Seaman, *A Framework for Providing Integrated Services Over Shared and Switched IEEE 802 LAN Technologies*, Internet Engineering Task Force, Internet Draft draft-ietf-issll-is802-framework-04.txt, March 1998.

[GrSp93] J. Grinham, M. Spratt, IEEE 802.12 Demand Priority and Multimedia, in Proc. of 4th International Workshop on Operating Systems Support for Digital Audio and Video, pp. 75 -86, Lancaster, November 1993.

[GVC96] P. Goyal, H. M. Vin, H. Cheng, *Start-time Fair Queuing: A Scheduling Algorithm for Integrated Services Packet Switching Networks*, in Proc. of ACM SIGCOMM'96, pp. 157 - 168, Stanford University, August 1996.

[Hahn87] E. L. Hahne, *Round Robin Scheduling For Fair Flow Control In Data Communication Networks*, PhD Dissertation, Massachusetts Institute of Technology, February 1987.

[Herz96] S. Herzog, *Policy Control for RSVP: Architectual Overview*, Internet Engineering Task Force, Internet Draft draft-ietf-rsvp-policy-arch-01.txt, November 1996.

[HP92a] Hewlett-Packard, *LLA Programmer's Guide*, HP Manual, Part Number: 98194-60534, July 1992.

[HP92b] Hewlett-Packard, *PA-RISC 1.1 Architecture and Instruction Set Reference Manual*, HP Manual, Part Number: 09740-90039, September 1992.

[HP94] Hewlett-Packard, *CASCADE Architecture High-Performance LAN Cards, Hardware / External Reference Specification / Internal Maintenance Specification*, Version 2.0, October 1994.

[HPRG97] S. Herzog, D. Pendarakis, R. Rajan, R. Guerin, *Open Outsourcing Policy Service (OOPS) for RSVP*, Internet Engineering Task Force, Internet Draft draft-ietf-rsvp-policy-oops-00.ps, April 1997.

[ISO93] ISO/IEC, *ANSI/IEEE Standard 802.1D - Media access control (MAC) bridges*, July 1993.

[ISO95] ISO/IEC, *ANSI/IEEE Standard 802.12 - Demand Priority Access Method, Physical Layer and Repeater Specification for 100 Mb/s Operation*, November 1995.

[ISO97a] ISO/IEC, *IEEE P802.1Q/D8 - Draft Standard for Virtual Bridged Local Area Networks*, September 1997.

[ISO97b] ISO/IEC, *IEEE P802.1p/D8 - Supplement to MAC Bridges: Traffic Expediting and Dynamic Multicast Filtering*, September 1997.

[JDSZ95] S. Jamin, P. B. Danzig, S. Shenker, L. Zhang, *A Measurement-based Admission Control Algorithm for Integrated Services Packet Networks*, in Proc. of ACM SIGCOMM'95, pp. 2 - 13, Cambridge MA, August 1995, also (extented version) in IEEE Transactions on Networking, Vol. 5, No. 1, pp. 56 - 70, February 1997.

[Jaco88] V. Jacobson, *Congestion Avoidance and Control*, in Proc. of ACM SIGCOMM'88, pp. 314 - 329, Standford University, August 1988.

[Jain89] R. Jain, *A Delay-Based Approach for Congestion Avoidance in Interconnected Heterogeneous Computer Networks*, in ACM Computer Communication Review, Vol. 19, No. 5, pp. 56 - 71, October 1989.

[Jain90] R. Jain, C*ongestion Control in Computer Networks: Issues and Trends*, in IEEE Network Magazine, Vol. 4, No. 3, pp. 24 - 30, May 1990.

[Jami96] S. Jamin, *A Measurement-based Admission Control Algorithm for Integrated Services Packet Networks*, PhD Dissertation, University of Southern California, August 1996.

[JNP98] V. Jacobson, K. Nichols, K. Poduri, *An Expedited Forwarding PHB*, Internet Engineering Task Force, Internet Draft draft-diffserv-phb-ef-00.txt, August 1998.

[JSD97] S. Jamin, S. Shenker, P. Danzig, *Comparison of Measurement-based Admission Control Algorithms for Controlled-Load Service*, in Proc. of IEEE INFOCOM'97, pp. 973 - 980, Kobe, April 1997.

[KaKa90] C. R. Kalmanek, H. Kanakia, *Rate Controlled Servers for Very High-Speed Networks*, in Proc. of IEEE Global Telecommunications Conference (Globecom), pp. 12 - 20, San Diego, December 1990.

[Kesh91] S. Keshav, *On the Efficient Implementation of Fair Queueing*, in Internetworking Research and Experience, Vol. 2, No. 3, pp. 157 - 173, September 1991.

[Kesh92] S. Keshav, *Congestion Control in Computer Networks*, PhD Dissertation, University of California at Berkeley, September 1992.

[Kesh97] S. Keshav, *An Engineering Approach to Computer Networking: ATM Networks, the Internet and the Telephone Network*, Addision Wesley, 1997.

[Kim96] P. Kim, *LLRMP: a Signalling Protocol for Reserving Resources in Bridged Networks*, in Proc. of OPENSIG'96, New York, October 1996.

[Kim97a] P. Kim, *Deterministic Service Guarantees in 802.12 Networks, Part I: The Single Hub Case*, Hewlett-Packard Technical Report HPL-97-147, December 1997, also (short version) in IEEE Transactions on Networking, Vol. 6, No. 5, October 1998.

[Kim97b] P. Kim, *Deterministic Service Guarantees in 802.12 Networks, Part II: The Cascaded Network Case*, Hewlett-Packard Technical Report HPL-97-148, December 1997, also (short version) in Proc. of IEEE INFOCOM'98, pp. 1376 - 1383, San Francisco, March 1998.

[KWC93] G. Kesidis, J. Walrand, C. Chang, *Effective Bandwidth for Multiclass Markov Fluids and Other ATM Sources*, in IEEE Transactions on Networking, Vol. 1, No. 4, pp. 424 - 428, August 1993.

[LeGa91] D. Le Gall, *MPEG: A Video Compression Standard for Multimedia Applications*, in Communications of the ACM, Vol. 34 No. 4, pp. 47 - 58, April 1991.

[LeWi91] W. E. Leland, D. V. Wilson, *High Time-Resolution Measurement and Analysis of LAN Traffic: Implications for LAN Interconnection*, in Proc. of IEEE INFOCOM'91, pp. 1360 - 1366, Bal Harbor, April 1991.

[Leym96] N. Leymann, *Eine Videokomponente für das Videokonferenzsystem Multimedia Collaboration*, Diploma Thesis, in German, Technical University of Berlin, August 1996.

[LiMo97] D. Lin, R. Morris, *Dynamics of Random Early Detection*, in Proc. of ACM SIGCOMM'97, pp. 127 - 137, Cannes, September 1997.

[LTWW94] W. E. Leland, M. S. Taqqu, W. Willinger, D. V. Wilson, *On the Self-Similar Nature of Ethernet Traffic (Extended Version)*, in IEEE/ACM Transactions on Networking, Vol. 2, No. 1, pp. 1 - 14, February 1994.

[MaMa96] M. Mathis, J. Mahdavi, *Forward Acknowledgement: Refining TCP Congestion Control*, in Proc. of ACM SIGCOMM'96, pp. 281 - 291, Stanford, August 1996.

[Mank90] A. Mankin, *Random Drop Congestion Control*, in Proc. of ACM SIGCOMM'90, pp. 1 - 7, Philadelphia Pennsylvania, September 1990.

[MaZa90] N. Maxemchuk, M. Zarki, *Routing and Flow Control in High Speed Wide Area Networks*, in Proc. of the IEEE, Vo. 78, No. 1, pp. 204 - 221, January 1990.

[McCJ95] S. McCanne, V. Jacobson, *vic: A Flexible Framework for Packet Video*, in Proc. of ACM Multimedia'95, pp. 511 - 522, San Francisco, November 1995.

[McCR91] K. McCloghrie, M. Rose, *Management Information Base for Network Management of TCP/IP-based Internets: MIB-II*, Internet Engineering Task Force, RFC 1213, March 1991.

[MFB+97] A. Mankin, F. Baker, B. Braden, S. Bradner, M. O'Dell, A. Romanov, A.Weinrib, L. Zhang, *Resource ReSerVation Protocol (RSVP) - Version 1 Applicability Statement - Some Guidelines on Deployment*, Internet Engineering Task Force, RFC 2208, September 1997.

[Mill92] D. L. Mills, *Network Time Protocol (Version 3), Specification, Implementation and Analysis*, Internet Engineering Task Force, RFC 1305, March 1992.

[Mill94] D. L. Mills, *Precision Synchronization of Computer Network Clocks*, in ACM Computer Communication Review, Vol. 24, No. 2, pp. 28 - 43, April 1994.

[Minz89] S. E. Minzer, *Broadband ISDN and Asynchronous Transfer Mode (ATM)*, in IEEE Communications Magazine, Vol. 27, pp. 17 - 24, September 1989.

[MJV96] S. McCanne, V. Jacobson, M. Vetterli, *Receiver-driven Layered Multicast*, in Proc. of ACM SIGCOMM'96, pp. 117 - 130, Stanford University, August 1996.

[MMFR96] M. Mathis, J. Mahdavi, S. Floyd, A. Romanow, *TCP Selective Ackknowledgement Option*, Internet Engineering Task Force, RFC 2018, October 1996.

[MoWa96] M. Molle, G. Watson, *100Base-T / IEEE 802.12 / Packet Switching*, in IEEE Communications Magazine, pp. 64 - 73, August 1996.

[Nagl87] J. B. Nagle, *On Packet Switches with Infinite Storage*, in IEEE Transactions on Communications, Vol. COM-35, No. 4, April 1987.

[NBB+98] K. Nichols, S. Blake, F. Baker, D. Black, *Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers*, Internet Engineering Task Force, Internet Draft draft-ietf-diffserv-header-02.txt, August 1998.

[OCon98] N. O'Connell, Private Communication, August 1998.

[OV96] OptiVision Inc., *OptiVision Live MPEG Communication System*, User's Guide, Version 1.2 f, September 1996.

[PaFl95] V. Paxson, S. Floyd, *Wide-Area Traffic: The Failure of Poisson Modeling*, in IEEE Transactions on Networking, Vol. 3, No. 3, pp. 226 - 224, June 1995.

[PaGa93] A. K. Parekh, R. G. Gallager, *A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Single-Node Case*, in IEEE Transactions on Networking, Vol. 1, No. 3, pp. 344 - 357, June 1993.

[PaGa94] A. K. Parekh, R. G. Gallager, *A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Multiple-Node Case*, in IEEE Transactions on Networking, Vol. 2, No. 2, pp. 137 - 150, April 1994.

[Paxs97] V. E. Paxson, *Measurement and Analysis of End-to-End Internet Dynamics*, PhD Dissertation, University of California at Berkeley, April 1997.

[Perl92] R. Perlman, *Interconnections: Bridges and Routers*, Addision Wesley, 1992.

[Plum82] D. C. Plummer, *An Ethernet Address Resolution Protocol*, Internet Engineering Task Force, RFC 826, November 1982.

[Post81a] J. Postel, *Internet Protocol*, Internet Engineering Task Force, RFC 791, September 1981.

[Post81b] J. Postel, *Transmission Control Protocol*, Internet Engineering Task Force, RFC 793, September 1981.

[Post81c] J. Postel, *User Datagram Protocol*, Internet Engineering Task Force, RFC 768, August 1981.

[PrPo87] W.Prue, J. Postel, *Something a Host Could Do with Source Quench: The Source Quench Introduced Delay (SQuID)*, Internet Engineering Task Force, RFC 1016, August 1987.

[RaJa90] K. K. Ramakrishnan, R. Jain, *A Binary Feedback Scheme for Congestion Avoidance in Computer Networks*, in ACM Transactions on Computer Systems, Vol. 8, No. 2, pp. 158 - 181, May 1990.

[Rive92] R. Rivest, *The MD5 Message-Digest Algorithm*, Internet Engineering Task Force, RFC 1321, April 1992.

[Schw97] U. Schwantag, *An Analysis of the Applicability of RSVP*, Diploma Thesis, Institute of Telematics, University Karlsruhe, July 1997.

[SDW92] W. T. Strayer, B. J. Dempsey, A. C. Weaver, *XTP: The Xpress Transfer Protocol*, Addison-Wesley, July 1992.

[ShBr95] S. Shenker, L. Breslau, *Two Issues in Reservation Establishment*, in Proc. of ACM SIGCOMM'95, pp. 14 - 26, Campridge MA, August 1995.

[Shen95] S. Shenker, *Is Admission Control Necessary ?*, INFOCOM'95 Panel Discussion, Presentation Slides at: ftp://ftp.parc.xerox.com/pub/net-research/infocom95.html/, April 1995.

[ShVa95] M. Shreedhar, G. Varghese, *Efficient Fair Queuing using Deficit Round Robin*, in Proc. of ACM SIGCOMM'95, pp. 231 - 242, Cambridge MA, August 1995.

[ShZh93] K. Shin, Q. Zheng, *Mixed Time-Constrained and Non-Time-Constrained Communications in Local Area Networks*, in IEEE Transaction on Communications, Vol. 41, No. 11, pp. 1668 - 1676, November 1993.

[Siga94] K. Sigan, *NetwWare - The Professional Reference*, New Riders Publishing (NRP), Indianapolis, 1994.

[SMM98] J. Semke, J. Mahdavi, M. Mathis, *Automatic TCP Buffer Tuning*, in Proc. of ACM SIGCOMM'98, Vancouver, August 1998.

[SPDB95] S. Shenker, C. Partridge, B. Davie, L. Breslau, *Specification of Predictive Quality of Service*, Internet Engineering Task Force, Internet Draft draft-ietf-intserv-predictive-svc-01.txt, 1995.

[SPG97] S. Shenker, C. Partridge, R. Guerin, *Specification of the Guaranteed Quality of Service*, Internet Engineering Task Force, RFC 2212, September 1997.

[SPW95] S. Shenker, C. Partridge, J. Wroclawski, *Specification of Controlled Delay Quality of Service*, Internet Engineering Task Force, Internet Draft draft-ietf-intserv-control-del-svc-02.txt, November 1995.

[SSC97] M. Seaman, A. Smith, E. Crawley, *Integrated Service Mappings on IEEE 802 Networks*, Internet Engineering Task Force, Internet Draft draft-ietf-is802-srv-mapping-01.txt, November 1997.

[Stei96] R. Steinmetz, *Human Perception of Jitter and Media Synchronization*, in IEEE Journal on Selected Areas in Communications, Vol. 14, No. 1, pp. 61 - 72, January 1996.

[Stev94] W. R. Stevens, *TCP/IP Illustrated, Volume 1*, Addison Wesley, 1994.

[SW97a] S. Shenker, J, Wroclawski, *Network Element Service Specification Template*, Internet Engineering Task Force, RFC 2216, September 1997.

[SW97b] S. Shenker, J. Wroclawski, *General Characterisation Parameters for Integrated Service Network Elements*, Internet Engineering Task Force, RFC 2215, September 1997.

[SZC90] S. Shenker, L. Zhang, D. Clark, *Some Observations on the Dynamics of a Congestion Control Algorithm*, in ACM Computer Communication Review, Vol. 20, No. 5, pp. 30 - 39, October 1990.

[Tane89] A. S. Tanenbaum, *Computer Networks*, Second Edition, Prentice Hall, 1989.

[TG97] D. Tse, M. Grossglauser, *Measurement-based Call Admission Control: Analysis and Simulation*, in IEEE INFOCOM'97, pp. 981 - 989, Kobe, Japan, April 1997.

[VCR98] L. Vicisano, J. Crowcroft, L. Rizzo, *TCP-like Congestion Control for Layered Multicast Data Transfer*, in Proc. of IEEE INFOCOM'98, pp. 996 - 1003, San Francisco, March 1998.

[VeCh95] C. Venkatramani, T. Chiueh, *Design, Implementation, and Evaluation of a Software-based Real-Time Ethernet Protocol*, in Proc. of ACM SIGCOMM'95, pp. 27 - 37, August 1995.

[Venk97] C. Venkatramani, *The Design, Implementation and Evaluation of RETHER: A Real-Time Ethernet Protocol*, PhD Dissertation, State University New York, January 1997.

[VeSo97] R. Vesilo, V. Solo, *Techniques for Adaptive Estimation of Effective Bandwidth in ATM Networks*, in IEEE Global Telecommunications Conference (Globecom), pp. 1344 - 1348, Phoenix, Arizona, November 1997.

[VZF91] D. C. Verma, H. Zhang, D. Ferrari, *Delay Jitter Control for Real-Time Communication in a Packet Switching Network*, in Proc. of Tricomm'91, pp. 35 - 43, Chapel Hill - North Carolina, April, 1991.

[WaCr91] Z. Wang, J. Crowcroft, *A New Congestion Control Scheme: Slow Start and Search (Tri-S)*, in ACM Computer Communication Review, Vol. 21, No. 1, pp. 32 - 43, January 1991.

[WAG+95] G. Watson, A. Albrecht, J. Grinham, J. Curcio, D. Dove, S. Goody, M. Spratt, P. Thaler, *The Demand Priority MAC Protocol*, in IEEE Network, Vol. 9, No. 1, pp. 28 - 34, January 1995.

[WaCr92] Z. Wang, J. Crowcroft, *Eliminating Periodic Packet Losses in the 4.3-Tahoe BSD TCP, Congestion Control Algorithm*, in ACM Computer Communication Review, Vol. 22, No. 2, pp. 9 - 16, April 1992.

[Wall91] G. K. Wallace, *The JPEG Still Picture Compression Standard*, in Communications of the ACM, Vol. 34, No. 4, pp. 31 - 44, April 1991.

[Weis95] A. Weiss, *An Introduction to Large Deviations for Communication Networks*, in IEEE Journal on Selected Areas in Communications, Vol. 13, No. 6, pp. 938 - 952, August 1995.

[WGS97] L. C. Wolf, C. Griwodz, R. Steinmetz, *Multimedia Communication*, in Proceedings of the IEEE, Vol. 85, No. 12, pp. 1925 - 1932, December 1997.

[WhCr97] P. White, J. Crowcroft, *The Integrated Services in the Internet: State of the Art*, in Proceedings of the IEEE, Vol. 85, No. 12, pp. 1934 - 1946, December 1997.

[Wro97a] J. Wroclawski, *Specification of the Controlled-Load Network Element Service*, Internet Engineering Task Force, RFC 2211, September 1997.

[Wro97b] J. Wroclawski, *The Use of RSVP with IETF Integrated Services*, Internet Engineering Task Force, RFC 2210, September 1997.

[WTSW95] W. Willinger, M. S. Taqqu, R. Sherman, D. V. Wilson, *Self-Similarity Through High-Variability: Statistical Analysis of Ethernet LAN Traffic at the Source Level*, in Proc. of ACM SIGCOMM'95, pp. 100 - 113, Cambridge MA, August 1995.

[YHBB97] R. Yavatkar, D. Hoffman, Y. Bernet, F. Baker, *SBM (Subnet Bandwidth Manager): A Proposal for Admission Control over IEEE 802-style networks*, Internet Engineering Task Force, Internet Draft draft-ietf-issll-is802-sbm-04.txt, July 1997.

[ZDE+93] L. Zhang, S. Deering, D. Estrin, S. Shenker, D. Zappala, *RSVP: A New Resource ReSerVation Protocol*, in IEEE Networks, Vol. 7, No. 5, pp. 8 - 17, September 1993.

[Zhan93] H. Zhang, *Service Disciplines For Packet-Switching Integrated-Services Networks*, PhD Dissertation, University of California at Berkeley, 1993.

[Zhan91] L. Zhang, Virtual Clock: *A New Traffic Control Algorithm for Packet-Switched Networks*, in ACM Transactions on Computer Systems, Vol. 9, No. 2, pp. 101 - 124, May 1991.

[Zhan95] H. Zhang, *Service Disciplines for Guaranteed Performance Service in Packet-Switching Networks*, in Proceedings of the IEEE, Vol. 83, No. 10, pp. 1373 - 1396, October 1995.

[ZhBu95] S. Zhang, A. Burns, *An Optimal Sychronous Bandwidth Allocation Scheme for Guaranteeing Synchronous Message Deadlines with the Timed-Token MAC Protocol*, in IEEE Transactions on Networking, Vol. 3, No, 6, pp. 729 - 741, December 1995.

[ZhFe93] H. Zhang, D. Ferrari, *Rate-Controlled Static-Priority Queueing*, in Proc. of IEEE INFOCOM'93, pp. 227 - 236, San Francisco, March 1993.

[ZhFe94] H. Zhang, D. Ferrari, *Rate-Controlled Service Disciplines, in Journal of High Speed Networks*, Vol. 3, No. 4, pp. 389 - 412, 1994.

[ZhKe91] H. Zhang, S. Keshav, *Comparison of Rate-Based Service Disciplines*, in Proc. of ACM SIGCOMM'91, pp. 113 - 121, Zürich, September 1991.

[ZSC91] L. Zhang, S. Shenker, D. Clark, *Observations on the Dynamics of a Congestion Control Algorithm: The Effects of Two-Way Traffic*, in Proc. of ACM SIGCOMM'91, pp. 133 - 147, Zürich, September 1991.