Quality-of-Service Issues in Networking Environments

Burkhard Stiller *
University of Cambridge, Computer Laboratory
New Museums Site, Pembroke Street
Cambridge, CB2 3QR, England, U.K.
Phone: +44 +1223 334476, FAX: +44 +1223 334678
E-Mail: Burkhard.Stiller@cl.cam.ac.uk

Abstract

Quality-of-Service (QoS) issues in networking environments cover various separate areas and topics. They include at least the specification of applications requirements, the definition of network services, QoS models, resource reservation methods, negotiation and transformation methods for QoS, and operating system support for guaranteed services. An embracing approach for handling, dealing with, and supporting QoS in different scenarios and technical set-ups is required to manage sufficiently forthcoming communication and networking tasks. Modern telecommunication systems require an integrated architecture for applications, communication subsystems, and network perspectives to overcome drawbacks of traditional communication architectures, such as redundant protocol functionality, weakly designed interfaces between the end-system and a network adapter, or impossibility of specifying and guaranteeing QoS parameter.

This work contains the discussion of a number of interconnected QoS issues, *e.g.,* QoS mapping, QoS negotiation, QoS-based configuration of communication protocols, or QoS aspects in Asynchronous Transfer Mode (ATM) signaling protocols, which have been dealt with during a one-year research fellowship. This report is not intended to be a complete description of every technical detail, but tries to provide a brief overall picture of the emerging and explosively developing QoS issues in telecommunication systems. Additionally, investigations of some of these issues are undertaken in a more closer detail. It is mainly focussed on QoS mapping, negotiation, and updating in the communication protocol area.

* The author is currently on leave from Universität Karlsruhe, Institut für Telematik, D-76128 Karlsruhe, Germany, and is sponsored by the Commission of the European Communities as a Research Fellow under the Human Capital and Mobility Scheme (RG 19327).

Contents

List of Figures

List of Tables

1 Preface

This work on *Quality-of-Service Issues in Networking Environments* comprises the results of a one-year fellowship granted from the Commission of the European Communities (CEC), sponsored under the Research Grant Number 19327, being part of the Human Capital and Mobility Scheme, and performed at the University of Cambridge, Computer Laboratory, England, U.K.

Because of the broad range of interesting and relevant aspects concerning Quality-of-Service (QoS) issues and related areas, independent sections of this report deal with separate aspects of QoS. Nearly each of them will be assisted by an autonomous report or publication that is included in the appendix if it is not available as another Technical Report at the Computer Laboratory of the University of Cambridge.

The following introduction and motivation will provide the general linkage between sections to clarify the close technical relationships between every regarded research question.

2 Introduction and Outline

Many specific QoS-oriented topics have been tackled in current and important research areas of Quality-of-Service issues in telecommunication systems. In general they comprise at least

- the specification of application requirements,
- the definition and use of network services,
- QoS models,
- resource reservation methods,
- negotiation methods for QoS,
- transformation methods for QoS, and
- operating system support for guaranteed services.

Some of them will be introduced within the following Sections in more detail, while others are presented in a brief survey only.

Firstly, the emerging networking environment, *e.g.,* in terms of the Asynchronous Transfer Mode (ATM), offers a variety of services that can be used by modern multimedia applications. Efficient use of these services is possible due to architectural and procedural features of ATM networks and, especially, due to initial statements of QoS for ATM end-systems and the network, including simple resource reservation tasks per virtual channel.

Protocol support for QoS negotiations and resource reservations in ATM networks is available in terms of various ATM signaling protocols, but sometimes based on traditional signaling systems and fairly outdated assumptions on the network structure or end-system behavior. Therefore, based on a survey, new methods and concepts for modern signaling protocols in an object-oriented environment have been investigated shortly.

Secondly, a survey of QoS issues in existing approaches has been supplied. Classifications for Quality-of-Service parameters and their use allow for a clear view of separated concerns and for a comparison of modelling aspects, intentions, and details of the approach. Three sub-topics have been worked on in closer detail. That is on one hand the development of two negotiation schemes for QoS that can be applied to end-system architectures that offer an extensive application service interface. On the other hand the development deals with

QoS mappings within OSI-based (Open Systems Interconnection) communication protocols. Finally, a framework for handling QoS parameter value updates in a networking environment including inter-dependencies between QoS parameters and system parameters has been designed.

Thirdly, a sufficient support of diverse application requirements can be gained by applying a flexible communication protocol configuration approach to existing communication subsystem architectures. Therefore, an outline of a Protocol Configuration Manager within the Function-based Communication Subsystem (F-CSS) is given and, additionally, the interoperability of different approaches for configuring communication protocols, *e.g.,* with the approach called Dynamic Configuration of Protocols (Da CaPo), has been evaluated.

This work is organized as follows. Subsequent to a brief motivation in Section 3 a presentation of ATM signaling systems and protocols will be given in Section 4, which is enhanced by the Appendix A. Afterwards, a survey on QoS models and concepts is discussed in Section 5. An approach dealing with QoS negotiations is outlined in Section 6, while the QoS mapping problem is focussed on in Section 7 and Appendix B, additionally. Updating issues of QoS and configuration parameters are dealt within Section 8. Protocol configuration aspects are introduced within Section 9 and Appendix C, while interoperability concerns are delineated in Section 10 and Appendix D. Finally, a brief summary and some concluding remarks are presented in Section 11. Placed behind are Abbreviations and an extensive Reference list.

3 Motivation

The support of sufficient Quality-of-Service (QoS) for applications residing in a distributed environment and running on top of high performance networks is a demanding issue. Currently, the areas to provide this support adequately, include communication protocols, operating systems support, and offered network services. A configurable approach of communication protocols offers the needed protocol flexibility to react accordingly on various different requirements. Additionally, a parametrizable operating system for communication purposes and resource reservation methods allow for guarantees of communication services.

Quality-of-Service is one of the main issues in modern telecommunication systems to characterize application requirements, define network services, and apply tariffing models. Nevertheless, the variety of approaches for these issues is overwhelming. Therefore, specific assumptions on applications, networks, the QoS model, and the operating system environment lead to various approaches of providing sufficient services within a telecommunication system. Unfortunately, most of them do not offer a perfect solution for all QoS-relevant networking tasks yet — due to the problem's complexity of a large number of QoS parameters and still unknown, but necessary relationships between various components of the telecommunication system —, but some approaches deal with important and emerging sub-issues already.

4 ATM Signaling Systems and Protocols

The most important prerequisite for any networking task to be fulfilled successfully, is an appropriate network. Dependent on the geographic distance between participants of a joint application, Local Area Networks (LAN), Metropolitan Area Networks (MAN), or Wide Area Networks (WAN) provide a useful basis. Additionally, linked with the detailed requirements of used applications various different networking technologies, such as Token Ring, Ethernet, Fiber Distributed Data Interface (FDDI), Distributed Queue Dual Bus (DQDB), Asynchronous Transfer Mode (ATM), or X.25-Networks, may be applied. Nevertheless, emerging multimedia applications do require an extensive amount of bandwidth, considerably low delays, small delay jitters, and low error rates, which have to be delivered to these applications throughout the network *and* end-system as well. A quite well suited network for modern multimedia applications, concerning the network's access and maintenance methods, is the ATM-based B-ISDN (Broadband Integrated Services Digital Network).

The main aspect covered by signaling systems and protocols for ATM networks concerns the possibility to manage, maintain, and control a user-driven communication between arbitrary ATM end-systems connected to an ATM network. The tasks and procedures defined for, *e.g.,* setting-up an ATM connection, are often very different concerning the relevant specifications of various working bodies (such as ITU-T or ATM-Forum) or certain vendors, although the basics to be done for maintaining ATM connections are always principally similar. The reason for this situation is, that the intentions of working bodies and vendors are different and each one of them follows specific strategies for certain scenarios (such as for point-to-point unicast or point-to-multipoint multicast). Nevertheless, various types of characteristics (such as addressing, multicast groups, or interworking) are requested from applications residing on top of ATM networks. Therefore, the survey identifies for several of selected approaches (such as Q.2931, UNI 3.1, or CMAP) important characteristics and relevant scenarios.

Procedures and features of existing signaling systems and protocols and a brief proposal of high-level functionality and operations for a sufficient ATM-based signaling system may be found in [1] and are included in Appendix A completely.

4.1 QoS Negotiation in ATM Signaling Protocols

Anticipating the discussion on QoS negotiations, within current ATM signaling protocols a QoS negotiation is quite restricted. Therefore, some minor enhancements in messages and information elements, *e.g.,* of the UNI 3.1 or Q.2931 protocols, would lead to a significant improvement of QoS negotiation facilities. Assuming that the *ATM Traffic Descriptor* is capable of specifying not only a single QoS parameter value, but value intervals and ranges, two modifications during the connection establishment and data transfer phase can be done.

Firstly, during the connection establishment the *ATM Traffic Descriptor* could be supplied in the *SET-UP*–message (issued from the end-system and heading towards the first directly connected ATM switch, cf. Figure 9), including provisionally reserved resources. They are defined according to the actual existing QoS parameter fields in the descriptor. Additionally, the inclusion of the ATM Traffic Descriptor in the *CONNECT*–message (a type of acknowledgement on the receive of a set-up request) provides a basis for the relaxation of reserved resources within the ATM network. In general, this modification of the connection establishment phase avoids wasting of unnecessary or, even worse, unwanted resources.

Secondly, during the data transfer phase the introduction of a *NEGOTIATE*–message, which includes an enhanced ATM Traffic Descriptor, can be used for newly required, changed, or to be updated resources. Again, the specification is based on the internal specifications of the ATM Traffic Descriptor. Applying this modification, an on-line adjustment — increase or reduction — of resources is possible. Therefore, a more flexible approach of adapting to very likely changes in a connection is possible, without releasing the connection currently in use and establishing an entirely new connection including the newly requested resources.

5 A Survey of QoS Models and Concepts

Work on QoS models and concepts has been done in various scenarios and within a huge number of projects. Most of them differ according to assumptions, targets, and solution approaches [2].[1] Therefore, the following brief survey of these approaches compares main ideas. QoS modelling includes at least the following aspects:

- structure and type of QoS parameters,
- negotiation of QoS,
- mapping of QoS,
- resource reservation methods,
- methods for updating QoS, and
- methods for monitoring and maintaining QoS.

5.1 Definition and Classification of QoS

Quality-of-Service parameters are integral parts of communication protocols and/or their service specification. For QoS modelling the term *service* is important. The currently applied service definition is based on [3] and is used throughout this Section as follows:

Definition of Service:
Service comprises the aggregate of all capabilities and features, which are available at an interface between two layers, or in general, between two objects that are correlated.

Furthermore, dependent on the standard body, different definitions for QoS are available:

ISO Definition of QoS:
A set of qualities related to the collective behavior of one or more objects [4].

ITU-T Definition of QoS:
The collective effect of service performances which determine the degree of satisfaction of a user of the service [5].

The ISO (International Organization for Standardization) definition assumes an abstract model of objects, while the ITU-T (International Telecommunications Union – Telecommunications Standardization Sector) includes the personal degree of satisfaction of a user. Even more, the detailed QoS parameter specified in different documents of protocols vary as well.

To outline these QoS parameters an initial, three-dimensional classification seems to be useful (cf. Figure 1). It does not deliver a completely overlap-free picture of QoS parameters, but allows for a sufficient grouping of the tremendous variety. The first dimension of QoS parameter distinguishes QoS parameters according to an architectural point of view, while three hierarchical levels may be separated.

Network-dependent QoS — These N-QoS parameters are located in the two lowest levels of the ISO/OSI Basic Reference Model [3] and define capabilities of the network, its topology, and features of the medium itself. Examples may be found in I.356 [6] for ATM networks or for FDDI in [7].

[1] This reference cites a survey on QoS in German.

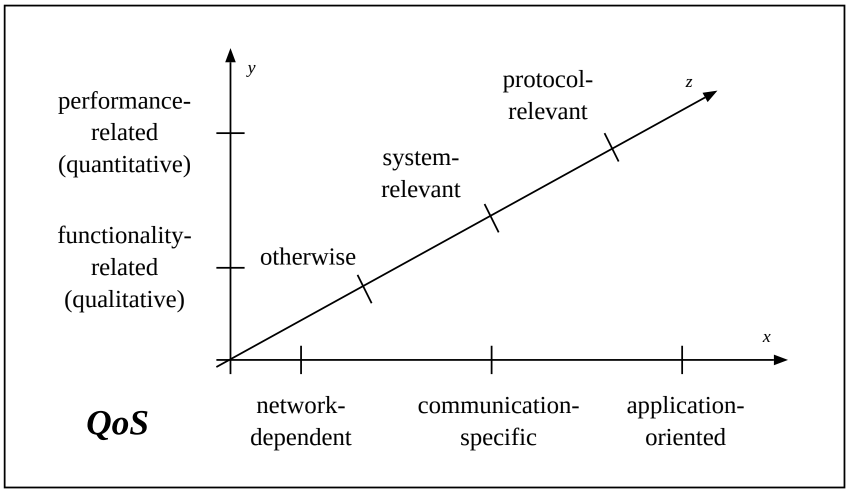

Figure 1: Orthogonal Classification of QoS

Communication-specific QoS — In contrast, C-QoS define the features of processing user data according to communication protocol tasks, such as checksumming, synchronization, delays or throughputs. They comprise at least levels 3 to 7 of the ISO/OSI Basic Reference Model. The "end-to-end"–character of these QoS parameters is important. Examples may be found in TP4 (Transport Protocol Class 4) [8], CLNP (Connectionless Network Protocol) [9], XTP (Xpress Transfer Protocol) [10], or ATM protocols [11].

Application-oriented QoS — Finally, the application-oriented point of view is very fuzzy, but included in A-QoS parameters. It clearly depicts an application perspective only. A number of emerging applications always extends these QoS parameters. They may encompass, *e.g.,* throughput, delay. error rates, frame size, frame rates, or priority.

The second and orthogonal dimension of QoS parameters encompasses the expressiveness of a single QoS parameter related to measurable values or identified functionalities.

Performance-related QoS — All performance-related QoS parameters can be expressed by measurable and numerical values. They are called quantitative QoS parameters as well. Additionally, a unit, such as bit, byte, or s(econd), is attached to the value. Examples are a jitter value of 25 ms or a throughput of 100 Mbit/s.

Functionality-related QoS — In contrast, functionality-related QoS parameters are used to express a qualitative aspect, such as synchronization, security, or priority. Moreover, additional information per functionality-related QoS parameter may be added, such as what type of synchronization or security is required.

Finally, the third dimension includes an implementation versus design point of view.

Protocol-relevant QoS — These QoS parameters include parameters such as window size, retransmission counter, or acknowledgement timer. They are used within a communication protocol and define certain internal details of protocol mechanisms.

System-relevant QoS — System-relevant QoS parameters do include end-system oder intermediate-system relevant parameters, such as memory management methods or scheduling mechanisms. They show a heavy dependence on system activity besides its effects on communicated user data. Additionally, the operating system perspective is quite strong, in contrast to the protocol perspective.

5.2 Modelling of QoS

Modelling of QoS pays notice to service types, which may be applied to one or a group of QoS parameters. The following two Types-of-Service (ToS) are distinguished:

1. *Best Effort Service* and
2. *Guaranteed Service* subdivided into:
   - *Statistical ("soft") guarantee* and
   - *Deterministic ("hard") guarantee.*

Best effort services present the traditional manner of services, normally without any guarantees on QoS parameter values. The system's design does not have to take into account any reservation or negotiation methods. However, guaranteed services require certian prerequisistes. They include, besides resource reservation methods, the possibility to specify QoS, monitor QoS, and react on QoS events. Statistical guarantees allow for a sufficient utilazation of resorces from the providers point of view, while deterministic guarantees define a fixed set of QoS for an application or user.

Overview of QoS Models — As a great variety of QoS models exists, the following list will simply show some of them and cite relevant references. Work in standard bodies includes:

- ISO communication protocols, such as TP4 [8] or CLNP [9],
- QoS Basic Framework [4],
- ITU-T Categories for Services [12],
- ITU-T Recommendation E.800: Quality-of-Service and Dependability Vocabulary [5],
- ITU-T Recommendation I.350: General Aspects of Quality-of-Service and network Performance in Digital Networks, Including ISDNs [13], and
- ITU-T Recommendations for ATM, such as I.356 [6] and I.371 [14].

Furthermore, work on QoS in academia and industrial research centers include:

- OSI'95 [15],
- QoS-A (Quality-of-Service Architecture) [16], [17], [18]
- F-CSS (Function-based Communication Subsystem) [19], [20],
- QoS Broker [21], [22],
- CINEMA (Configurable Integrated Multimedia Architecture) [23],
- QoS Management [24]
- HeiTS (Heidelberger Transport System) [25],
- Extended Integrated Reference Model [26],
- TINA (Telecommunications Information Networking Architecture) [27], and
- ANSA (Advanced Network System Architecture) [28].

Criteria	TP4	ISO	E.800	I.350	OSI 95	F-CSS	QoS-A	Broker
Point of view of								
User	yes	yes	yes	yes	yes	yes	yes	yes
Provider	no	yes	yes	yes	no	no	yes	yes
Use of								
A-QoS	no	yes	yes	yes	no	yes	yes	yes
C-QoS	yes	yes	yes	yes	yes	yes	yes	yes
N-QoS	no	yes	yes	yes	no	yes (7)	yes	yes
Completeness of								
A-QoS	—	no (3)	no	no	—	no (8)	no	no
C-QoS	no	no (3)	no	yes (6)	yes	yes(8)	yes	no (9)
N-QoS	—	no (3)	no	yes	—	no	no	yes (9)
Negotiation of								
Single QoS	no	yes	—	—	yes	yes	yes	yes
Multiple QoS	yes	yes	—	—	yes	yes	yes	yes
Specification of								
Best Effort Service	yes (1)	yes	—	—	yes (1)	yes (1)	yes	yes
Soft Guarantees	no	yes (4)	—	—	no (4)	no (4)	yes	yes
Hard Guarantees	no (2)	yes (5)	—	—	no (5)	no (5)	yes	yes
Methods to deliver								
Best Effort Service	yes (1)	yes	—	—	yes	yes	yes	yes
Soft Guarantees	—	no (4)	—	—	—	—	yes	yes
Hard Guarantees	—	no (5)	—	—	—	—	no	no

Table 1: Comparison of QoS Models

5.2.1 Comparison of QoS Models

Table 1 compares a number of approaches due to their main focus. Characteristic features are listed and the main criteria for the comparison are the point of view (user or provider), the use and completeness of QoS, a negotiation possibility for QoS, the possibility to specify Types-of-Service (ToS), and the definition of methods to really deliver the specified ToS. Some explanatory remarks are added according to the following numbers:

(1) Implicitly, without a detailed specification of a mechanism, the best effort type of service will be delivered.
(2) Valid except for parameters concerning reliability, which allow for a bit error-free delivery of user data applying checksumming and acknowledgements.
(3) The QoS framework does allow for the specification of a *framework* for QoS.
(4) Implicitly, QoS monitoring will be done by the service provider. Specified QoS parameters may be used for resource reservation methods, which are not defined.
(5) Hard guarantees may not be provided without resource reservation methods, which are not defined. QoS parameters may be used for them.
(6) According to ITU-T C-QoS are complete. Nevertheless, on one hand QoS parameter such as synchronization are missing. On the other hand, they may be considered as A-QoS due to the ATM perspective.
(7) Implicitly, the mapping of C-QoS onto N-QoS is done, while evaluating specified configuration conditions.
(8) A-QoS and C-QoS have been integrated into a single category.
(9) C-QoS and N-QoS have been integrated into a single category.

QoS	TP4	OSI	I.350	I.356
Throughput	×	×	×	×
Burstiness	—	(×)	—	×
Delay	×	×	(×)	×
Jitter	—	×	—	×
Response Time	—	×	—	—
Residual Error Rate	×	×	—	—
Transfer Reliability	×	×	×	×
Transfer Failure Probability	×	×	—	×
Elasticity	×	(×)	—	—
Protection	×	×	—	—
Priority	×	×	—	—
Costs	×	×	—	—

Table 2: Comparison of Standardized QoS Parameters

QoS	OSI 95	F-CSS	QoS-A	Broker
Throughput	×	×	×	—/×
Burstiness	—	—	×	—/×
Packet Size	×	(×)	×	×/×
Rate	—	×	×	×/—
Delay	×	×	×	×/×
Jitter	×	×	×	—/×
Response Time	—	×	—	—/—
Error Control	×	(×)	(×)	—/×
Data Corruption	×	×	—	—/—
Data Loss	×	×	×	×/×
Data Replication	—	×	—	—/—
Ordered Delivery	—	×	—	—/×
Protection	×	×	—	—/—
Priority	×	×	—	—/—
Costs	—	—	×	×/×
Synchronization	—	×	—	×/—
Compression	—	—	×	×/—

Table 3: Comparison of QoS Parameters in Research Projects

5.2.2 Comparison of QoS Parameters

Tables 2 and 3compare some approaches according to their detailed QoS parameters specified and used. Due to a potentially large number of empty entries, only important QoS parameters have been listed. The × denotes the existence of the parameter, while the — denotes its missing. Parenthesized crosses (×) define a directly non-supported parameter, since the requested semantic is provided by other parameters.

5.3 Methods for QoS

Four main methods for QoS can be distinguished: QoS negotiation, QoS mapping, and resource reservation methods as well as methods for delivering QoS, which are similar to communication protocols and its internal functions and mechanisms.

QoS Negotiation — QoS negotiation methods allow for the balance of QoS requirements between two or multiple peers of a unicast or multicast connection. In general, an increase or reduction of a single QoS parameter value is possible. Three different scenarios are identified:

- *Unilateral Negotiation*: This represents the simplest scenario, where the peer will be informed on certain QoS only.
- *Bilateral Negotiation*: A more sophisticated scenario allows for the balance of an initial, *e.g.,* sender-stimulated request according to receiver capabilities.
- *Multilateral Negotiation*: The most complex scenario overlaps many bilateral negotiations and admits multiple arrangements between sender and receiver(s).

QoS Mapping — Mapping of QoS has to be done in various separate steps. The main target remains to provide application requested QoS by the end-system and the network. Therefore, A-QoS have to be mapped onto C-QoS/N-QoS and onto protocol-relevant parameters. C-QoS themselves have to be mapped onto resources and parameters of the applied operating system and the network resources as well. As it can be seen, QoS parameter, internal paramater, and resources have to be handled homogeneously. Various approaches, stating a very specific situation (concerning certain applications, protocols, operating systems, and networks) have been undertaken [29], [30], [31], [32], [33], [34].

Resource Reservation — Methods to reserve resources apply to mainly two areas: the local end-system (normally the operating-system) and indermediate-systems (normally the network). In general, the tasks encompass the reservation of resources, such as memory (buffer), bandwidth, processing time (CPU), or scheduling mechanisms for certain application requirements.

Since these methods do form an quite extensive reserach area, only the operating system support will be dealt with briefly in Subsection 5.4. Network-wide approaches include. *e.g.,* Internet-Stream Protocol, Version II [35], Resource Reservation Protocol [36], Realtime Channel Allocation Protocol [37], Session Reservation Protocol [38], or Flow Protocol [39].

Delivering QoS — Finally, every mentioned method above is irrelevant if a communication protocol is not available. Therefore, the primary tasks of protocol processing does deliver some requested QoS to the application, such as reliability issues, *e.g.,* in terms of bit-error free user data. Obviously, a huge number of protocols exists and deal with certain QoS. Additionally, configuration approaches of communication protocols handle felxible QoS deliveries as well.

5.4 Operating System Support for QoS

Additionally, the support of sufficient networking performance relies, as mentioned above, on the operating system as well. An excellent overview of various projects and scheduling mechanisms may be found in [40]. Especially, the scheduling schemes applied to networking tasks are crucial for guaranteeing QoS parameter values.

In addition, resource models are developed to allow for the description of schedulable resources that have to be shared or exclusively used by different users. One approach has been developed to integrate scheduling mechanisms, resource administration, and QoS parameter mapping [31]. Further operating system support is done for continuous media in a real-time environment [41] and within the PEGASUS project, to provide a kernel that allows for guarantees of processing and scheduling times for multimedia application streams [42].

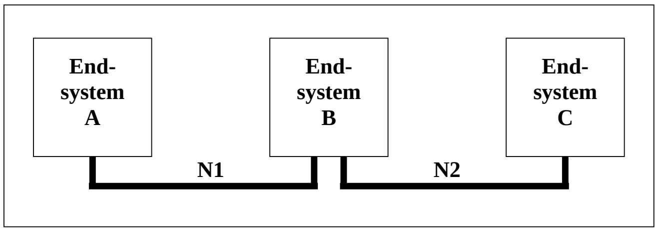

Figure 2: An Example for Diverse Networks Connected to a Single End-system

The manner to structure an operating system according to processor reservations is discussed in [43], while dynamic renegotiation of these reservations are supported.

6 QoS Negotiation

Negotiation schemes are applied in distributed systems to adapt participating peer requirements of end-systems to each other. Especially, the negotiation of QoS parameters between peers needs to be done for, *e.g.,* initial settings of window and buffer sizes, or bandwidth allocation. Even more, the exchange and acceptance of configuration descriptions (cf. Section 10) in a multicasting environment have to be considered. Negotiations may implement different scenarios and may be supported by different methods according to their intended involvement of two or multiple peers.

Therefore, two opposite negotiation schemes, an optimistic and a pessimistic one, have been proposed for experimentation [44].[2] Main issues for this development included an exchange and negotiation of configuration descriptions *and* QoS parameters due to a flexible approach for communication subsystem protocols. 2-way or 3-way handshakes between peers are sufficient in a 1:1–communication relation (connection). If 1:n– or in general n:m–communication relations are required, an enhanced approach is necessary. Therefore, the following areas had to be taken into consideration for the negotiation scheme:

- diverse network connectivity,
- addressing and reacheability,
- acknowledged administrative operations,
- simultaneous update requests,
- resources, and
- algorithms for comparisons.

6.1 Diverse Network Connectivity

The negotiation scheme has to deal with a very simple scenario as depicted in Figure 2. Two networks *N1* and *N2* are connected to end-system *B*, while *B* wants to establish an 1:2–communication relation to *A* and *C*. Different features and QoS of both, networks as well as user requirements in the end-systems, have to be balanced to provide a sufficient multicast service for *B*.

[2] Due to a publication of these issues in German, a brief summary of the QoS negotiation concept is given here, instead of an inclusion of the entire document in the Appendix.

	Active End-system	Passive End-system
Addressing	explicit (list of single addresses)	implicit (group address)
Group Size	static and known	dynamical and unknown
Priority	uniform and high	uniform and low

Table 4: Classification of End-systems

6.2 Addressing and Reacheability

The type of addressing used does influence the negotiation schemes, since all peers hidden behind a group address might not be known explicitly by the originator of a communication. In this case no statement on the number of involved peers can be given and, therefore, the adjustment of QoS parameters and configuration descriptions has to be done without this knowledge. Additionally, the probably highly dynamic behavior of groups may lead to frequent joinings and leavings of members.

Table 4 proposes a simple classification for end-systems involved in multicast communication scenarios, to overcome the problems of explicitly unknown peers in a group. Reacheability is always provided when lists of single addresses are maintained. Therefore, dependent on the importance of end-systems in a communication relation they have to act as active *or* passive end-systems.

6.3 Acknowledged Administrative Operations

Within negotiation schemes the time consumed for balancing various requirements are important. In this case acknowledged administrative operations, such as connection set-up or tear-down, even stretch the time for providing a final result. For that reason, it is eminent to integrate all necessary acknowledgements during a negotiation into a single message. Nevertheless, omitting an acknowledgement in general is not possible, since a guarantee of uniform views of the communication relation has to be provided.

6.4 Simultaneous Update Requests

Updates may include the re-calculation of QoS parameter values or the initiation of a re-configuration (cf. Subsection 8). Reasons for their applicability include changes in the network, such as congestions or failure of links, or explicit requests of applications for already existing communication relations. Monitoring of the current status in connected end-systems might detect changes, which may lead to a re-calculation or re-configuration request. If many end-systems participate, a huge number of almost simultaneous update requests will be issued. This problem can be solved, if in a binary back-off strategy the issuing of an update request will be delayed, while listening for already sent-off requests.

6.5 Resources

Resources for defining negotiation schemes in a networking environment encompass in general two separate parts:

- global resources include the initial application requirements and the current status of the network;
- local resources include end-system specific available memory, processor time, and applicable protocol functions and mechanisms.

Global resources are prerequisites to be met completely or slightly changed, and local resources will be negotiated to meet, support, or provide global resources. Obviously, local resources differ according to the end-system. Therefore, mainly two types of balancing the resources are possible:

(a) All relevant resource information will be collected and balanced after the collection, or
(b) every end-system proposes an update and these results will be compared and applied afterwards.

Either type requires a type of central moderator for handling the collections or dealing with a comparison.

6.6 Algorithms for Comparisons

Remotely generated configuration descriptions or QoS parameter updates for a current situation have to be evaluated to determine the final result of the update. For that reason, a centralized system — the moderator — has to compare incoming proposals and has to generate the final result. The comparison of configuration descriptions needs to apply graph algorithms for the comparison of flat graphs. Since this takes too long in general terms of processing times, the comparison of nodes within the graph will be undertaken only. This is similar to a functional comparison of protocol functions included in the configuration description.

6.7 Design of Negotiation Schemes

Based on the classification of end-systems in Table 4, active end-systems are allowed to vote, while passive end-systems do not have any method of expressing their needs. This scenario will be improved by the introduction of a panel chair (moderator), who is always an active end-system and is responsible for balancing incoming update requests.

A panel discussion presents a feasible comparison with a real-world scenario. Assuming that participants in a panel discussion are listeners (passive end-systems) or panelists (active end-systems), the panelists are well known persons *p* (single addresses *p*) and the listeners are unknown individuals of the discussion *foo* (group address *foo*). Therefore, the panel discussion is sufficiently determined in its organizational structure and may be transmitted, *e.g.,* via MBone [45].

The Optimistic Negotiation Scheme — The basic assumption of an optimistic negotiation scheme is the fact that a moderator will receive QoS parameter values or configuration results that match every peer's requirements. According to Figure 3, the moderator calculates a value and distributes it to every peer. An acceptance check will be run at every peer. Active peers transmit a reply, passive peers do not. If all active peer-replies include a positive answer, the moderator initiates a done command containing the agreed upon QoS parameter values or configuration results. Passive peers do not reply and wait for the done command in a positive case or dismiss themselves from the communication in a negative case.

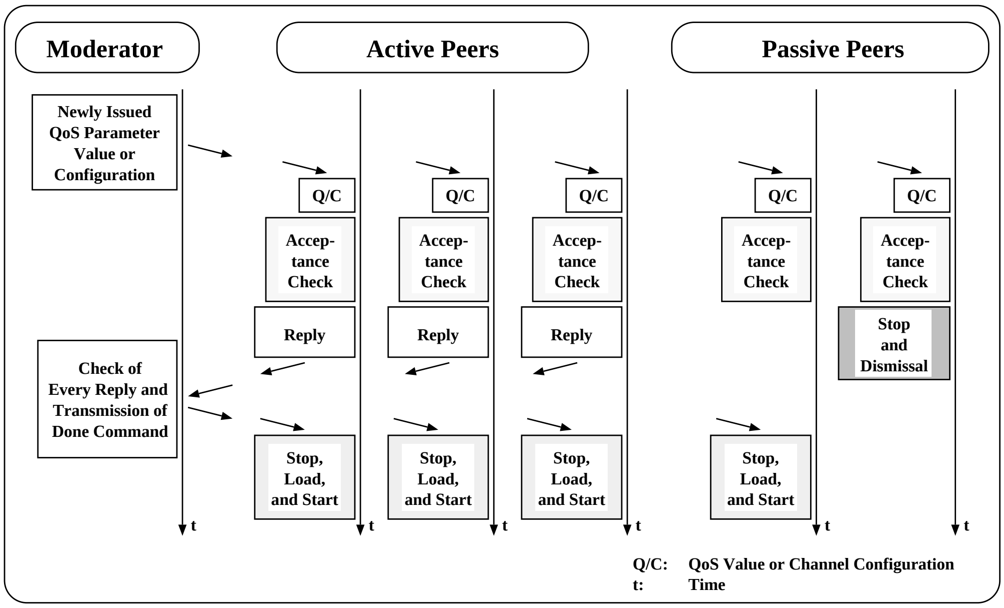

Figure 3: Optimistic Negotiation Scheme

Pessimistic Negotiation — Instead of a quite similar assumption of common requests from all peers, the pessimistic negotiation scheme collects from every active peer local information on resources or QoS parameter values to be used for calculating a moderator-driven conclusion according to Figure 4. This result will be distributed afterwards to every active and passive peer and, since a common agreement already has been reached, active peers accept the result for initiating their local configuration or set-up the new configuration. Passive peers act differently, while accepting the result after a positive acceptance check or dismissing themselves otherwise.

Either peer has to implement both alternative negotiation schemes, since in the case of a failure of the optimistic approach the pessimistic one has to be used afterwards to calculate a final result. Both schemes need to have a centralized moderator. This does not limit the usage of these approaches, because the final result has to be unique for every participating active peer. Any decentralized or distributed approach would need a large number of two-by-two negotiations that cause a large amount of negotiation time and lead to unsuitable efforts.

7 QoS Mapping

As it has been motivated within earlier Sections and during the QoS survey in Section 5, the need of generally applicable mapping functions between different layers of communication protocols is obvious. These transformed QoS information are used for system-internal scheduling techniques, bandwidth reservation, or resource management in general. Especially for an OSI-based protocol stack, a number of generalized mapping functions have been proposed to allow for a sufficient mapping between various OSI layers.

The document on these issues in the OSI-based environment may be found in [33] and has been added entirely in Appendix B as well. In the current status it includes the protocol-

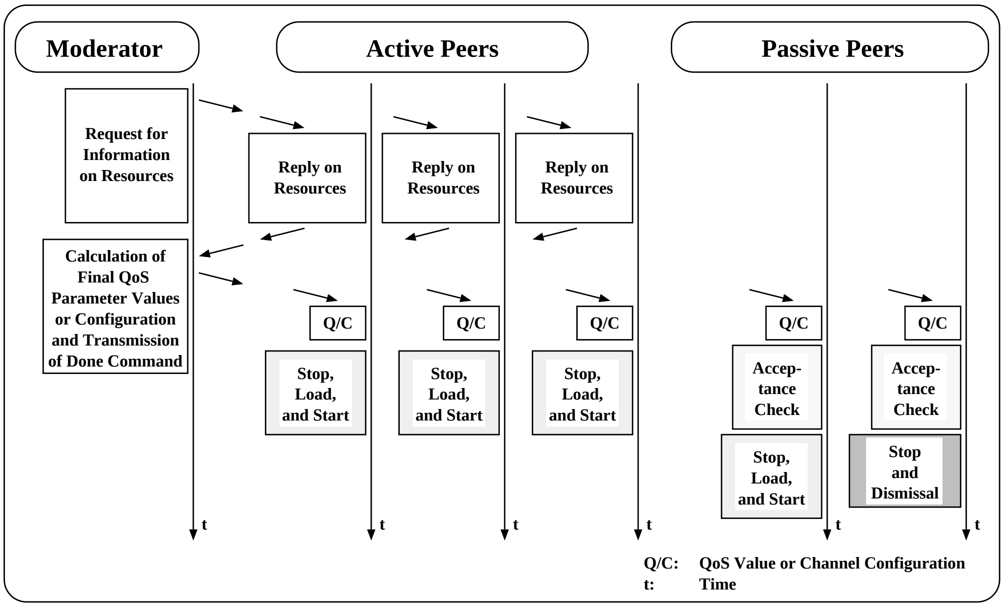

Figure 4: Pessimistic Negotiation Scheme

relevant QoS parameters and mapping only. Further extensions will include a QoS model for the end-system resources, such as memory, processing power, scheduling technique, bus bandwidth, etc.

8 Updating QoS and Configuration Parameters

Communication protocols and operating systems have to be parametrized using internal configuration parameters, such as window sizes, retry counters, or scheduling mechanisms, that rely closely on requested application-oriented or network-dependent QoS, such as bandwidth or delay. Moreover, these internal parameters have to be re-calculated from time to time due to network changes (such as congestion or line break-down) or due to application-specific alterations (such as enhanced bandwidth requirements or increased reliability) to adjust a temporary or semi-permanent "out-of-tune" service behavior.

On one hand, traditional protocols are not well suited for appropriate use on gigabit networks [46]. On the other hand, they do not support mechanisms for flexibly adapting to new environments. Since well established applications of the last approximately two decades did not require various different services, former end-system architectures did and still do not offer the required *service flexibilities*, necessary for modern applications (such as tele-conferencing, tele-learning, virtual reality, or in general multimedia applications). These flexibilities may be supported by configurable communication protocols. A real-time video application requires different protocol functionality, *e.g.,* jitter control and synchronization, than a reliable file-transfer, *e.g.,* acknowledgements and checksumming, besides common functionality for both, such as data packet generation and data transmission.

Therefore, a suitable configuration of a communication protocol can be determined by certain QoS parameters specified by an application. Approaches, such as the proposed flexible protocol configurations [19], [47], and [48], require the tasks of updating their QoS parameter values. However in general, other areas of QoS-oriented work (such as modern protocols [10], new architectures [49], enhanced service interfaces [19], [15], and operating system support [41], [31] (cf. Subsection 5.4)) have to be regarded in an integrated manner, providing a suitable solution for QoS guarantees within an entire networking environment.

Supposing that these approaches solve the lack of service flexibility (an "off-line" problem) and offer solutions for guaranteeing QoS in the end-system and network, the "on-line" situation of adapting configuration parameters sufficiently according to newly arising environmental behaviors still remains open. That is in particular the adaptation of services during run-time of a communication protocol. In fact, any of these alterations can be made explicitly visible (*e.g.,* by newly issued application requests) or are implicitly detected within the end-system by a monitor, which monitors end-system visible states of the network, such as a link congestion — resulting in a dropping of throughput and increased delay — or increased bit error rates on the links. An appropriate reaction to this detrimental behavior and specific solution for some cases is to keep the requested level of QoS parameter values in the end-system by applying a QoS-driven update of parameters. These *configuration parameters* are an inherent constituent of communication protocols and stimulate the increase or decrease of certain transport-related or network-dependent QoS parameter values, finally, adjusting the communication subsystem's behavior according to initial application-requested QoS.

Therefore, a rule-based evaluation and QoS updating framework for configuration parameters in a networking environment has been developed. Details may be found in [50]. The resulting "rulework" can be used within highly dynamic environments in a communication subsystem that offers the possibility to specify for every QoS parameter both, a bounding interval of values and an average value.

9 QoS and Protocol Configuration

Protocol configuration includes, as mentioned above, the selection and configuration of a communication protocol according to application requirements, network prerequisites, and end-system status. As it has been described, these requirements, prerequisites, and status may be expressed in QoS parameters and corresponding values. Therefore, QoS issues and protocol configuration are closely related.

Especially in the Function-based Communication Subsystem (F-CSS) [19] an enhanced application service interface is offered, allowing for the specification of various QoS-parameters that are used to configure a sufficiently application-tailored communication protocol. Details on the service interface, its QoS parameters, and service types may be found in [20].

9.1 PROCOM

The Protocol Configuration Manager (PROCOM) forms the main part of F-CSS, handling QoS requests and providing a description of an application-tailored protocol. An illustration on PROCOM's tasks may be found in [51] and has been added additionally to the Appendix C as well.

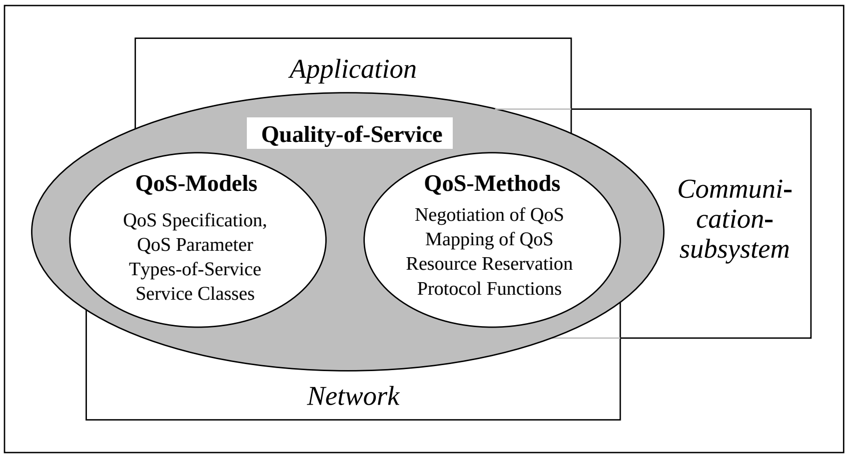

Figure 5: Summary and Structure of QoS Aspects

9.2 CogPiT

Furthermore, as a complete summary of ideas, solution approaches, applied solutions, prototypical implementation details, and preliminary performance results CogPiT (Configuration of Protocols in TIP) illustrates the completed project between the author's home institution and SIEMENS München, Zentralstelle für Forschung und Entwicklung (Research and Development), Germany.

The idea of protocol configuration has been applied to a transport protocol TEMPO being part of the Transport and Internetworking Package (TIP) and its QoS structure under UNIX. The detailed document on protocol configuration in TIP may be found as [52].

10 Interoperability of Configuration Approaches

If application requirements in terms of Quality-of-Service (QoS) are used to automatically configure communication protocols, different configuration systems (such as the Function-based Communication Subsystem, F-CSS and the Dynamic Configuration of Protocols, Da CaPo, approach) have to interoperate to allow for a consistent and correct protocol processing. Therefore, the applied structure of protocol building blocks of either system has to be logically equivalent. Additionally, the data packets to be exchanged have to provide a common format. A general scheme for negotiating the configuration description has been proposed applying protocol graphs as a convenient description technique.

The document on interoperability issues in configuration-based approaches may be found in [53] and has been added completely to the Appendix D as well.

11 Summary and Concluding Remarks

Quality-of-Service has to be dealt within an integrated manner, while considering users, applications, communication systems, end-systems, intermediate-systems, networks, and providers. The summarizing Figure 5 depicts an overview of these inter-relationships in an abstract and structured manner.

Outlook — In the near future at least three different tasks and questions will be relevant concerning QoS aspects. Firstly, standardized management methods, *e.g.,* the Performance or Configuration Managenemt, have to be applied to negotiation schemes, monitoring, and maintenance of QoS. Secondly, tariffing of communication costs is an emerging area, especially in the public environment of telecommunication providers. Thirdly, the problem of mapping application-oriented QoS parameters onto communication-specific and system-relevant QoS parameters needs to be tackled. A solution is this area — in addition to parametrizable operating systems and resource reservation methods in the network — will lead towards guaranteed services throughout entire networks and end-systems as well.

12 Acknowledgements

A one-year fellowship is challenging and seems to be quite long in the very beginning and early stage of work. Unfortunately, time passed very fast, since the environment provided interesting people, modern technical equipment, and exciting research areas as well. I have met various people in research, technical support, and administration at the Computer Laboratory that are individuals on their own, with different habits and characters, but all of them linked into Computer Science or its support.

Although the missing *good morning* or a simple *hi*–type of reply, while meeting people for the first time at the beginning of a day in the Computer Laboratory's buildings remains still quite unsatisfactory, some others always had the time to exchange some words of welcome or a brief talk about various issues. This made the inter-personal communication easier and motivated for the day's work and targets.

In a somewhat arbitrary order the following people of the Systems Research Group and the Computer Laboratory in general provided an excellent basis for technical discussions, for sharing experiences in telecommunications, for getting technical and administrative support, and for having fun in numerous common gatherings: my colleague Kobus van der Merve sharing the office for the time being in Cambridge and always proofreading documents, Simon Crosby, Xiaobao Cheng, Steve Pope, Kook-Jin Nam, Shaw Chuang, Robin Fairbairns, Scarlet Schwiderski, Jennifer Li Kam Wa, Daniel Gordon, John Bates, and Chai-Keong Toh. Library support was given in an unselfish and very helpful manner by Lewis Tiffany and Paola Bishop. Demanding conversations on classical music and life had been held with them additionally.

I am also indebted to Margaret Levitt, who explained the first important details on working in the Computer Laboratory, extended by Graham Titmus' and Martyn Johnson's help in system questions. Finally, thanks go to the Computer Laboratory, in person of Derek McAuley, Ian Leslie, and Head of Department Professor Roger Needham for running the administration of my stay.

Last but not least I am very grateful to the Commission of the European Communities for granting the fellowship to be fulfilled at the Computer Laboratory at the University of Cambridge.

A A Survey of UNI Signaling Systems and Protocols for ATM Networks

The following document has been published in:

ACM Computer Communications Review
Vol. 25, No. 2, April 1995; pages 21 – 33.

A Survey of UNI Signaling Systems and Protocols for ATM Networks

Burkhard Stiller [3]
University of Cambridge, Computer Laboratory
New Museums Site, Pembroke Street
Cambridge, CB2 3QR, England, U.K.
Phone: +44 +1223 334476, FAX: +44 +1223 334678
E-Mail: Burkhard.Stiller@cl.cam.ac.uk

Abstract

The main aspect covered by signaling systems and protocols for ATM networks concerns the possibility to manage, maintain, and control a user-driven communication between arbitrary ATM end-systems connected to an ATM network. The tasks and procedures defined for, e.g., *setting-up an ATM connection, are often very different concerning the relevant specifications of various working bodies (such as ITU-T or ATM-Forum) or certain vendors, although the basics to be done for maintaining ATM connections are always principally the same. The reason for this situation is, that the intentions of working bodies are different and each one of them follows specific strategies for certain scenarios (such as for point-to-point unicast or point-to-multipoint multicast). Nevertheless, various types of characteristics (such as addressing, multicast, or interworking) are requested from applications residing on top of ATM networks.*

Therefore, this survey of signaling systems and protocols for ATM networks identifies for several of selected approaches (such as Q.2931, UNI 3.1, or CMAP) important characteristics and relevant scenarios. Furthermore, a table-based comparison of some approaches has been added.

A.1 Introduction

Emerging high-speed networks for wide areas, such as the ATM (Asynchronous Transfer Mode)-based B-ISDN (Broadband-Integrated Services Digital Network), do require in an irregular meshed topology appropriate signaling protocols and systems to allow for the correct management of user-to-user communications, *e.g.,* between digital voice devices, IP (Internet Protocol) hosts, or LAN (Local Area Networks) interworking devices. The general picture for the interoperation of communication protocols in an ATM environment providing user-to-user communications is stated in the B-ISDN Protocol Reference Model [54]. *Signaling* — being part of the ATM control and management plane as well — is the critical issue of interoperability in wide area ATM networks. It is necessary for finally transferring user data between ATM end-systems within well defined ATM connections while dynamically generating outgoing call set-ups and clearings, accepting incoming call set-ups and clearings, processing status maintenance and notification tasks, and running connection control issues according to user requirements for modern services. The underlying ATM concept of virtual path identifiers and virtual channel identifiers [55] is "materialized" into a set of supporting signaling procedures and messages (exchanged between ATM switches and ATM end-systems as well) that are summarized as *call and connection control* issues.

[3] The author is currently on leave from Universität Karlsruhe, Institut für Telematik, D-76128 Karlsruhe, Germany, and is sponsored by the Commission of the European Communities as a Research Fellow under the Human Capital and Mobility Scheme (RG 19372).

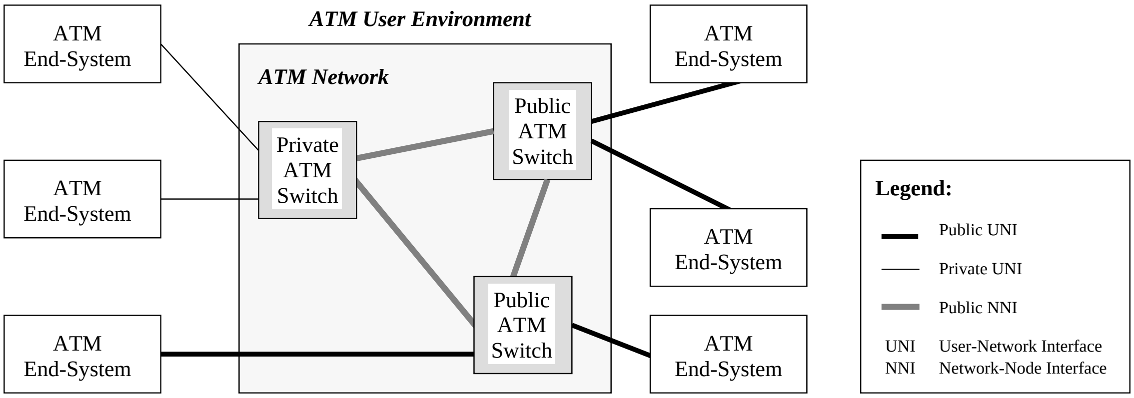

Figure 6: A Common Scenario for Interconnected ATM Switches and ATM End-systems

A network scenario as presented in Figure 6 is quite common, where private (corporate) and public ATM switches are interconnected. Two different types of *User-Network Interfaces (UNI)* are distinguished. Private UNIs are located between ATM end-systems and private ATM switches. Instead, public UNIs are located between ATM end-systems and public ATM switches. Public ATM switches may define a public ATM network, where private ATM switches define a private ATM network. A main difference involves the administrative responsibility for all domains. While private domains may follow a local management scheme, which offers only dedicated interconnection points to the outside world, public domains have to offer well-defined interconnection points and signaling systems as well. Additionally, the *Network-Node Interface (NNI)* between private or public ATM switches is not visible to any ATM end-system or user.

A wide range of signaling systems for ATM-based networks has been developed, in each case leading to the special focus of various working bodies or vendors. Most of todays signaling protocols as well as generalized signaling systems are still in the progress of definition or standardization. The main focus throughout this survey lays on the *User-Network Interface (UNI) Signaling* — besides *Network-Node Interface (NNI) Signaling* — without regarding ATM Adaptation Layer Signaling or Physical Network Signaling. In summary, UNI signaling specifies tasks and procedures between an ATM end-system and the next directly connected ATM switch.

In the N-ISDN (Narrowband-Integrated Services Digital Network) environment the International Telecommunications Union – Telecommunications Standardization Sector (ITU-T) Recommendation Q.931 [56] has been introduced as an UNI signaling system. Nevertheless, this *Digital Subscriber Signaling System No. 1 (DSS 1)* could not be applied to ATM-based B-ISDN networks directly, since it did not include schemes for dealing with virtual paths and virtual channels. Therefore, the *Digital Subscriber Signaling System No. 2 (DSS 2)* in the Recommendation Q.2931 [57] (as well as extensions, such as multicast [58] and Quality-of-Service parameter negotiation [59], [60], [61]) has received final preparation at the ITU Study Group 11/Working Party 2, Question 15 "Updating and enhancement of ISDN user-network call control protocols". Additionally, the ATM-Forum is in progress of aligning the standardization process with market opportunities and it produces pre-standardization specifications that are supported by multiple vendors, network operators, and users. The ATM-Forum signaling system is specified as UNI Specification 3.0 [62], now being updated in UNI Specification 3.1 [63]. These specifications are based on subsets and extensions of the above mentioned ITU-T Broadband Access Signaling Protocol Recommendations. An example of a vendor specific approach is SPANS (Simple Protocol for ATM Network Signaling) as Fore's interim signaling protocol for ATM LANs [64], [65]. Finally, examples that have been developed in a university environment (Washington University, St. Louis, U.S.A.), are the "Connection Management Access Protocol" [66], [67] or the "MultiService Network Architecture Connection Management" from the University of Cambridge, England, U.K. [68], [69].

The target of this survey is the identification of commonalities and differences between a number of regarded signaling systems and protocols, while focussing on their main features, characteristics, and procedures. Therefore, this survey is organized as follows. Providing a common basis on important issues, a concise introduction into signaling and into the ATM-based B-ISDN is presented in Section A.2. Afterwards, Section A.3 focusses on an overview of several signaling systems and protocols while contrasting differences in a table-based comparison. Finally, concluding this survey, Section A.4 proposes briefly important issues and characteristics for future signaling systems and protocols.

A.2 Signaling in the ATM-based B-ISDN

Relevant protocols for ATM are defined within a generic protocol architecture, the *B-ISDN Protocol Reference Model* [54], which contains orthogonal dimensions for different layers and various functions. Horizontal layers encompass:

- *Physical Layer*, specifying media technology-dependent issues;
- *ATM Layer*, including ATM specific functions, such as the generation of ATM cell headers, the (de-)multiplexing of cells, and mapping functions for identifiers;
- *ATM Adaptation Layer (AAL)*, including AAL specific functions, such as segmentation/reassembling of higher layer protocol data units into ATM cells, and convergence functions; and
- *Higher Layers* for application specific functions.

The vertical structure defines the *User Plane*, the *Control Plane*, and the *Management Plane*. User and control plane use identical physical and ATM layers respectively, but use different AAL and higher layers. Especially the user plane is responsible for transmission of user data, where the control plane handles all relevant issues on *Signaling*. This is the set-up, maintenance, and clear of calls and connections, while supporting different kinds of unicast, multicast, broadcast, and multipeer communication scenarios, negotiation and renegotiation of QoS (Quality-of-Service) parameters during set-up, admission control functions, ongoing QoS monitoring during the data transfer phase, and routeing of set-up requests through the network. Finally, the management plane encompasses relevant functions to interact and coordinate between user and control plane activities.

A.2.1 The Virtual Path and Virtual Channel Concept

Cells are small 53 byte long units of transportation. The relevant format is defined in I.361 [70]. They are exchanged between an *originator* or "calling party" and a *responder* or "called

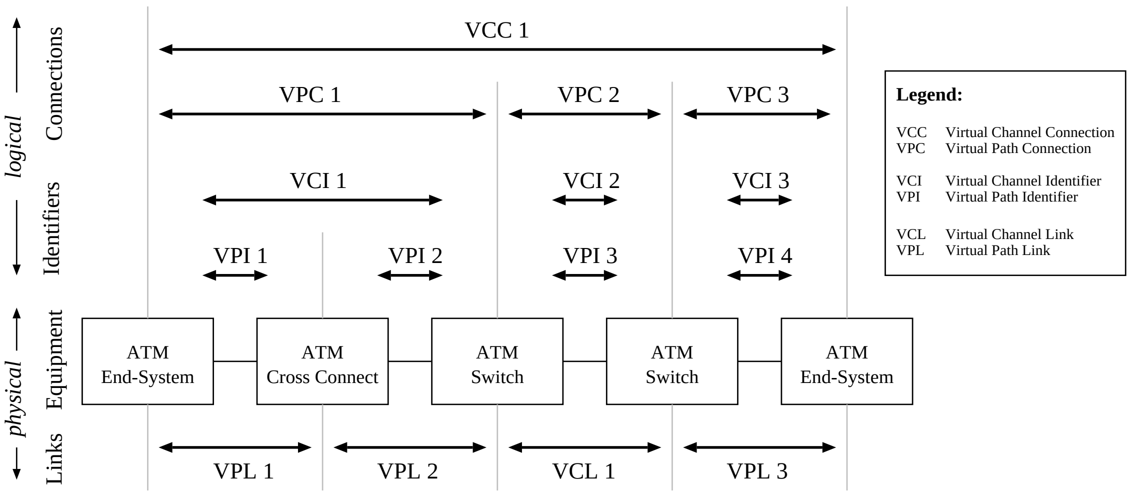

Figure 7: Links, Equipment, Identifiers, and Connections

party" that are identified by ATM addresses. Various addressing formats (such as IP addressing [71], public ISDN network E.164 addressing [72], or OSI NSAP addressing [73]) may be used for ATM. The ATM address is divided into an address and a sub-address [62], [74].

A *Virtual Channel (VC)* describes a unidirectional logical ATM connection between two or more ATM end-systems, which has some amount of reserved resources, *e.g.,* bandwidth. In contrast, a *Virtual Path (VP)* defines a number of virtual channels as a unit of observation for unidirectional traffic between an ATM end-system and an ATM switch. Connections relying on either type are logically equal. A *Virtual Channel Link (VCL)* is bounded by ATM switches on either side, which can be uniquely referred to by a *Virtual Channel Identifier (VCI)*. Instead, a *Virtual Path Link (VPL)* defines a physical end-to-end connection between either an ATM end-system or an ATM switch or ATM cross connect respectively (cf. Figure 7). Any VPL is identified by a *Virtual Path Identifier (VPI)*. Finally, one or multiple interconnected virtual channel links are regarded as a *Virtual Channel Connection (VCC)* and one or multiple interconnected virtual path links are regarded as a *Virtual Path Connection (VPC)*. An *ATM Cross Connect* — also known as a *VP Switch* — switches VPIs only, while an *ATM Switch* — also referred to as an *VC(+VP) Switch* — switches VCIs and VPIs. Nevertheless, throughout this survey both types of switches are regarded as ATM switches. *ATM connections* are referred to as switched physical or virtual channels between ATM end-systems. Finally, a *call* defines an abstract issue of an originator to set-up or maintain one or multiple connections to one or multiple responders. Therefore, the call includes more application specific details such as multicast or multipeer requirements [75].

A.2.2 Key Issues in an ATM Signaling Scenario

Signaling Protocol Architecture – A necessary prerequisite for transferring signaling messages is a reliable network service. Therefore, a SAAL (Signaling ATM Adaptation Layer) based on AAL type 5 [76] has been defined in the Recommendations Q.2100 [77], Q.2110 [78], Q,2120 [79], and Q.2130 [80]. *E.g.,* Q.2931 [57] runs on top of SAAL, which in turn runs on top of the ATM layer. Different types of AAL may be used as well, *e.g.,* SPANS [64] runs on top of either AAL type 3 or type 5. Moreover, the UNI signaling may be processed on top of a TCP/IP (Transmission Control Protocol/Internet Protocol) [81], [71] stack and may be used for transmitting signaling messages, to ensure the reliable transfer of these messages, as it is done by the Q.93B-based Netcomm signaling protocol [82]. Finally, signaling protocols may not require reliable signaling message transfer services at all, when the application itself is responsible for dealing with errors, such as corrupted, lost, or duplicated signaling messages.

For Network-Node Interface signaling in B-ISDN a specific B-ISUP (Broadband-ISDN User Part) ([83], [84], [85], [86], [87], and [88]) based on the ISUP (ISDN User Part) ([89], [90], [91], and [92]) of the *Signaling System No. 7 (SSN 7)* — defined completely in Q.700-Q.782 [93] — is located on top of the MTP 1, 2, or 3 (Message Transfer Part) [94] of SSN 7. This regular layer 3 of SSN 7 runs on top of the already mentioned SAAL. Finally, meta signaling is defined in Q.142x [95] and done directly inside the ATM layer, as an internal extension of the ATM layer.

Signaling Features and Tasks – Any signaling system has to hide completely any internals of the network and its topology from the user perspective. That is an important reason for the definition of the User-Network Interface, where the user only recognizes access methods that are handled with the "first" connected ATM switch, but has no knowledge on, *e.g.,* how to reach the potential receiver somewhere connected to the network. In general, signaling in ATM has to be regarded within three different layers:

- Layer 1 signaling is done between physical hardware devices;
- Layer 2 signaling allows for the interconnection of Interworking Units, *e.g.,* between LANs, using layer 2 endpoints; and
- Layer 3 signaling supports the establishment of calls and connections between digital voice devices, IP or IPX hosts, relying on layer 3 endpoints.

Therefore, UNI signaling in the B-ISDN environment is located in layer 3 and defines peer functions for the network connection control. Furthermore, signaling issues may be separated into two distinct control areas:

- *Connection (Bearer) Control* defines procedures to set-up or initialize features of the user data connection, *e.g.,* the ATM connection, or the process of connecting that type of connection.
- *Call Control* defines procedures for maintaining the connection itself, *e.g.,* associating specific VPIs or VCIs with a calling user, calling of any destination, or clearing VPI/VCI tables.

Two important tasks exist in any signaling scenario: network-dependent and service-dependent signaling tasks. The specific target of signaling within ATM networks includes two main ATM *network-dependent tasks*, only:

- Set-up, maintenance, and clear of VCCs and VPCs and
- negotiation of traffic characteristics.

Service-dependent tasks are in general completely independent of any specific network feature and are relevant for signaling systems. But they are *not* compulsory for signaling systems and *may* only be integrated in the signaling system definition approach, *e.g.:*

- The definition of multicast and multipeer scenarios and its direct support;
- the symmetric or asymmetric behavior of connections; or
- QoS parameter negotiation (such as bandwidth, latency, or error rates) for services.

Unfortunately, in case of ATM, QoS is related to services *and* the network as well, *e.g.,* any service requirement regarding throughput has to be mapped onto the network specific ATM peak cell rate, sustainable cell rate, or maximum burst size. Therefore, a complete signaling system *shall* include network-dependent tasks *and* service-dependent tasks as well.

Virtual Signaling Channels – In ATM-based B-ISDN networks *Out-of-Band Virtual Signaling Channels (VSC)* are used. They form the main basis for processing signaling tasks, *e.g.,* transmitting and receiving signaling control messages. In general, three different types of VSCs are distinguished:

- Meta Virtual Signaling Channel (MVSC),
- Broadcast Virtual Signaling Channel (BVSC), and
- Point-to-Point Virtual Signaling Channel (PVSC).

MVSCs are always bidirectional and are used to establish BVSCs or PVSCs as necessary. These in turn, are used to signal all types of ATM connection signaling messages between different ATM end-systems and ATM switches. BVSCs use to be unidirectional, while PVSCs are bidirectional. At least, one VSC of either type is mandatory and, therefore, permanent for each single UNI. However, one static VSC, regardless of which type, has to be assigned at a defined (pair of) VPIs and VCIs initially for each ATM end-system. *E.g.,* for a PVSC the VCI is always set to 5 and the VPI is always non-zero, while a MVSC is defined by a VCI set to 1 and the VPI equals always non-zero, if the signaling is done with remote networks. Otherwise (the VPI equals zero), user signaling is processed with the local switch [70], [96].

Comparison of ATM Signaling with X.25 Signaling – The manner of setting-up an ATM-connection is briefly compared with existing connection establishment procedures. Since a reliable ATM transport service for signaling messages (cf. above) is necessary, ATM signaling messages can not be sent directly into the network via the ATM layer. Instead, in a traditional signaling system and protocol X.25 [97] a "call request message" is delivered directly to the end-system via the reliable network layer, which is used for user data transmissions as well.

Additionally, X.25 signaling is defined between end users. In contrast, an ATM end-system transmits a set-up message to the next ATM switch and this switch will proceed with the necessary set-up tasks between following ATM switches or any final responder ATM end-system.

However, the introduction of a UNI between ATM end-systems and ATM switches is quite similar to the abstraction of the DTE/DCE (Data Terminal Equipment/Data Circuit-Terminating Equipment) concept in X.25, since a user access is processed between the ATM end-system and the ATM switch. But in contrast to X.25, each ATM connection may be additionally specified with a certain set of characteristics (such as QoS parameters), which in turn may be negotiated with the ATM network.

Organization	Standard	Content	Remarks
ITU-T ITU-T	Q.700 series Q.930/Q.931	NNI Signaling N-ISDN Voice Standard	SSN 7 DSS 1
ITU-T	Q.93B	Basic Call	Preliminary DSS 2
ITU-T	Q.2931 Q.298x Q.29xy	Point-to-Point Point-to-Multipoint Various additional features	DSS 2 DSS 2 DSS 2
ETSI	DE/SPS-5024 DE/SPS-5034	Basic Call Supervisory Call	
ATM Forum	UNI Spec 3.0 UNI Spec 3.1	Point-to-Point and Point-to-Multipoint Point-to-Point and Point-to-Multipoint	
Fore Systems	SPANS	Point-to-Multipoint	For ATM LANs
Netcomm	Q.93B-based	Point-to-Point	
NetExpress	FAST Select	Fast Call Set-up procedure	
Bellcore	EXPANSE	Multimedia Services	
AT&T Bell Laboratories	GSP	Connection Management	For various environ.
Washington University	CMAP	Multipoint-to-Multipoint	
University of Melbourne	gNET	Point-to-Multipoint	For ATM LANs
University of Cambridge	MSNL-CM	RPC-based Message Transfer	For Fairisle

Table 5: Overview of Several Signaling System Drafts, (Pre-)Standards, and Definitions

A.3 Overview and Comparison of Signaling Systems and Protocols

Existing signaling systems and protocols are listed in an overview in Table 5, which is not complete, but presents several relevant approaches. The Figures 10 and 11 include a table-based comparison of several approaches, while depicting a number of important capabilities of signaling protocols.

The ITU-T Recommendations for the Signaling System No. 7 (SSN 7) in the Q.700-series [93] encompass the N-ISDN version of an NNI signaling system. Recommendations for the voice standard in N-ISDN are defined in Q.930 [98], Q.931 (similar to I.451) [56], and Q.93B [99]. The Digital Subscriber Signaling System No. 2 (DSS 2) for B-ISDN in the final text of Q.2931 [57] and the draft text of Q.298x [58] is based on older versions (as Q.93B and Q.931). ETSI (European Telecommunications Standards Institute) is in progress of defining a UNI signaling protocol, which is subdivided into *basic call* [100] and *supervisory call* [101] issues. The ATM-Forum defined the UNI Specification 3.0 [62], now being updated in UNI Specification 3.1 [63], which is not interoperable with the former version. A proposal for UNI 4.0 is under discussion [102]. These specifications are based on subsets and extensions of the ITU-T Broadband Access Signaling Protocol Recommendations. Additionally, examples of vendor specific signaling protocols are SPANS [64], [65], a Q.93B-based protocol [82], and FAST Select [103]. Furthermore, Bellcore worked on EXPANSE in support of complex multimedia services [104], and Bell invented GSP (Generic Signaling Protocol) [105], [106]. Finally, university work has been driven by project requirements, such as with CMAP (Connection Access Management Protocol) [66], [67], [107], gNET [108], and MSNL-CM (MultiService Network Layer-Connection Management) [68], [69].

Class	Type	Q.2931/Q.298x	UNI 3.0/3.1
Point-to-Point	Set-up	Alerting	—
		Call proceeding	Call proceeding
		Connect	Connect
		Connect acknowledge	Connect acknowledge
		Set-up	Set-up
	Clearing	Release	Release
		Release complete	Release complete
	Managing	Notify	—
		Status	Status
		Status enquiry	Status enquiry
		—	Restart
		—	Restart acknowledge
Point-to-Multipoint	Joining	Add-connection	Add party
		Connection-added	—
		Connection-added acknowledge	Add party acknowledge
		—	Add party reject
	Leaving	Release-connection	Drop party
		Release-connection complete	Drop party acknowledge

Table 6: Messages for Q.2931/Q.298x and UNI3.0/3.1

A.3.1 Signaling Capabilities, Messages, and Procedures

User communications in an ATM network takes place after a set-up of an ATM connection, which is based on signaling messages transmitting set-up request information to one or multiple destinations. Depending on the request, either the network (an ATM switch) or the destination (an ATM end-system) agrees upon the information included in the message. A set-up includes other tasks to be managed, *e.g.,* the selection of an appropriate VP, the mapping of VCIs and VPIs in intermediate switches, resource negotiations between switches, and the call and connection control in the calling and called ATM end-system.

Capabilities – Q.2931/Q.298x, UNI 3.0, and UNI 3.1 include the specification of procedures and messages to achieve capabilities, which define the purpose of signaling in various scenarios. They encompass switched connections for point-to-point as well as point-to-multipoint connections that support traffic having symmetrical or asymmetrical bandwidth requirements. A statically pre-defined out-of-band signaling channel allows for maintaining these connections with uni- or bidirectional traffic in terms of user-/network-driven joining or leaving additional partners to/from an existing single connection call. Class X, A, and C ATM data link layer services (ATM transport services) are supported, while public UNI addressing formats for unique identifications of ATM endpoints are used. Furthermore, error recovery, end-to-end compatibility parameter identification, and signaling interworking with N-ISDN (Q.29xy only) are provided. Furthermore, Q.2931 runs on top of SAAL [77], while UNI 3.1 needs any type of reliable network service.

Messages and Information Elements – A message acts as a pre-defined function for exchanging signaling information, that is further refined into information elements. Depending

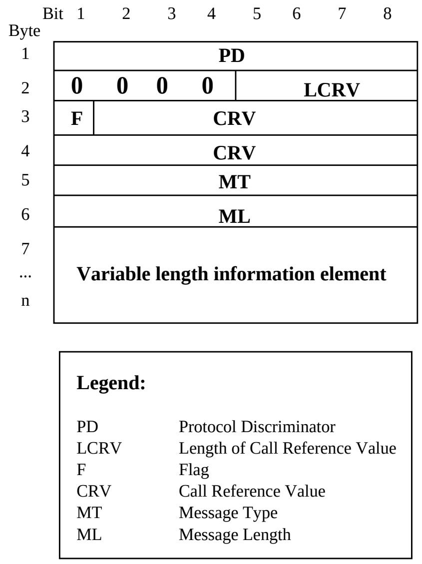

Figure 8: The Structure of Signaling Messages Defined out of Information Elements

on the message different directions (from the originator or from the responder of a call) degrees of significance (local, access, dual, and global) are distinguished. Table 6 lists defined messages of two classes and five types within Q.2931/ Q.298x and UNI 3.0/3.1. The structure of each message, possibly consisting out of several variable length information elements, is defined in Figure 8. The *Protocol Discriminator (PD)* distinguishes a specific message from other messages of different protocols that might be used, such as user-network call control messages as defined in Q.931, Q.93B, or X.25. The *Call Reference* consists out of a length field (LCRV), a call reference flag (F), and the value (CRV) itself. It is used — instead of an end-to-end significance — as a local identification of the call. The call reference value is fixed for the lifetime of a call at the originating side. Since multiple calls may be processed within one virtual signaling channel, the call reference value distinguishes messages from different calls. The flag defines, whether the message is being sent from the side that originated the call reference or from the responder. The *Message Type (MT)* specifies the type of the message (cf. Table 6) being sent. Obviously, the *Message Length (ML)* defines the length of the following one or multiple variable length information elements.[4]

Every single information element contains information that are necessary for the network and/or the responder, relating to the processed call. Certain coding rules for the information elements are defined. A general information element format includes identifiers for information elements, such as “Broadband-Sending Complete”, “ATM Traffic Descriptor”, or “Quality-of-Service Parameter”. *E.g.,* the “ATM Traffic Descriptor” contains the forward peak cell rate and the backward peak cell rate as defined in I.371 [14]. Reasons for a message and relevant diagnoses are transmitted by “Cause”. The “End-to-End Transit Delay” information element defines the selection of the end-to-end transit delay, while the “Connection Identifier” allows for the specification of the controlled connection. A “Number” defines a network addressing,

[4]It is important to recognize that multiple information elements may be used in a single message. *E.g.,* a “Set-up” message at least contains an “ATM Traffic Descriptor”, the “Called Party Number”, and the “Calling Party Number”.

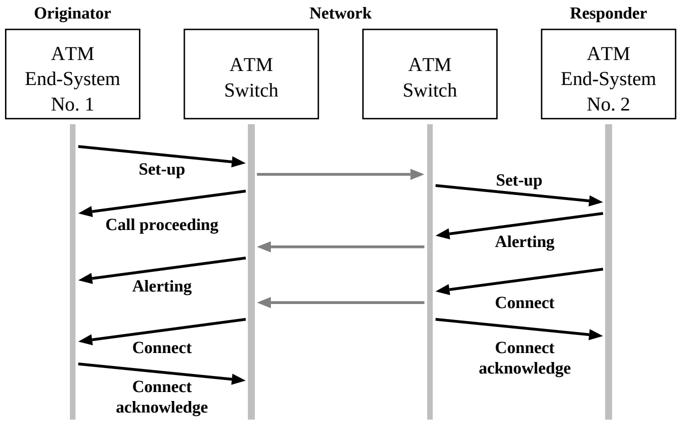

Figure 9: A Simple ATM Connection Set-up Using Messages

whereas a "Subaddress" defines full addressing. The designation of the desired transit network is specified within the "Transit Network Selection".

Example Procedure: ATM Connection Set-up – The use of some signaling messages is explained in a simple point-to-point scenario (cf. Figure 9), where originator and responder act differently. The originator initially requests a call by transmitting the "Set-up" message. If the call can be accepted, either in the "Call proceeding" or in the "Connect" message the newly allocated VPI/VCI is determined for the initiating ATM end-system No. 1. Otherwise, the "Release complete" message heading for the originator of the call determines that the network or the responder ATM end-system No. 2 are unable to connect. A responder receives a "Set-up" message for an offered call. An "Alerting" and a "Connect" message are transmitted accepting the incoming call or the "Release complete" message is used to determine a rejection. Finally, between each end-system and the directly connected switch the "Connect acknowledge" is exchanged.

CMAP Capabilities – The CMAP (Connection Management Access Protocol) [66], [67], [107] has been defined to manage complex multipoint connection scenarios in an ATM-based network environment. CMAP specifies different kinds of access procedures for set-up, clearing, and monitoring multipoint calls and connections that are handled as a distributed object throughout the ATM network. The main contribution of CMAP capabilities is a great variety of signaling procedures that are used within multipoint connections defining a dynamic call model, where *e.g.,* multiple connections may be added dynamically as a call is set-up. This can happen originator-driven or responder-driven, where the responder might or might not be member of that call. A number of procedures — operations in CMAP terminology — are defined as call procedures to set-up, modify, join, leave, clear, and trace calls, and as connection procedures to set-up, join, or leave connections. Managing procedures allow for the request of status information, alerting, resetting, and reporting. Furthermore, characteristics of calls (such as type, accessibility, originator identification, modifiability, or connection lists) and of connections (such as bandwidth, permissions, or notification) can be defined initially or during a two-way handshake negotiation phase.

Call Management Capabilities				Q.29xy (0)	UNI 3.0	UNI 3.1	UNI 4.0 (7)	SPANS	CMAP	gNET	MSNL-CM
Scenario	Addressing	Private UNI		no	yes	yes (6)	yes	yes	yes	yes	yes
		Public UNI		yes	yes	yes	yes	no	yes	no	no
	Unicast			yes	yes	yes	yes	yes	yes	yes	yes
	Multicast			yes	yes	yes	yes	yes	yes	yes	yes (10)
	Multipeer			no	no (1)	no (1)	yes	no	yes (9)	no	yes (10)
Special Features	Traffic descriptors			yes	yes (5)	yes	yes	-	(yes)	-	-
	Client registration			no	yes	yes	yes	-	yes	-	-
	Trace			yes	yes	yes	yes	-	yes	-	-
Set-up	Number of connections per call			1 (4)	1	1	multiple	1	multiple	1	1
	ATM transport service		Class A	yes	yes	yes	yes	-	-	yes	-
			Class C	yes	yes	yes	yes	-	-	yes	-
			Class D	-	no (2)	no (2)	yes	-	-	-	-
			Class X	yes	yes	yes	yes	-	-	-	-
	QoS negotiation			no (3)	no (3)	no (3)	yes	no	yes (4)	yes	no
Maintaining	Static			yes	yes	yes	yes	-	yes	-	yes (10)
	Dynamic	Joining	User-driven	yes	yes	yes	yes	-	yes	-	yes (10)
			Network-driven	-	-	-	-	-	-	-	-
		Leaving	User-driven	yes	yes	yes	yes	-	yes	-	yes (10)
			Network-driven	-	yes	yes	yes	-	no	-	yes (10)
Clearing	User-driven			yes	yes	yes	yes	-	yes	-	yes (10)
	Network-driven			-	yes	yes	yes	-	no	-	yes (10)

Note (0): This includes ITU-T Recommendations Q.2931 and Q.298x.
Note (1): The use of a Multicast-Server as a mapping system will provide multipeer functionality.
Note (2): The use of a Connectionless-Server with transport service classes C or X will provide a class D service.
Note (3): A rudimentary negotiation scheme is availabe: if the network can not support the QoS, it rejects the call.
Note (4): Only within point-to-point calls.
Note (5): Terminology in UNI 3.1 is set as "user cell rate", while referring to the content of a traffic descriptor.
Note (6): UNI 3.1 includes a completely updated and newer version of addressing features than defined in UNI 3.0.
Note (7): Entries in this column are of preliminary status, since the UNI version 4.0 is still under discussion.
Note (8): Any negotiation takes place dynamically for bandwidth requirements.
Note (9): Source descrimination is done via different VPs.
Note (10): Done in higher layers of the MultiService Network Architecture

Legend:

yes	existing feature
no	non existing feature
-	no statement possible

Figure 10: A Comparison of Call Management Capabilities

A.3.2 Selected Comparisons

UNI 3.0 versus UNI 3.1 – UNI 3.0 is based on Q.93B and is extended with certain features for addressing multicast issues. UNI 3.1 is the result of an update[5] of UNI 3.0 to become compatible with Q.2931 in most parts, including certain multicast features as defined in Q.298x. However, UNI 3.0 is not compatible with UNI 3.1 due to detailed changes in the signaling part. Important changes are the refinement of the Domain Specific Part of an E.164 Address and of the "General Message Format and Information Element Encoding" (mainly the variable length information elements of "ATM Adaptation Layer Parameters", "Broadband High Layer Information", "Broadband Low Layer Information", "Called Party Subaddress" , "Called Party Number", "Calling Party Subaddress" , "Calling Party Number", and "Quality-of-Service Parameter") and their corresponding procedures and states.

Q.2931 versus UNI 3.1 – A comparison of the Q.2931 Recommendation and UNI 3.1 identifies *non-supported issues* and *extensions*. *E.g.,* UNI 3.1 does not provide the messages "Alerting" and "Notify", and includes non-similar information elements "Information", "OAM Traffic Descriptor", "Notification Indicator", and "End-to-End Transit Delay" as well as their corresponding procedures and states. Extensions in UNI 3.1 encompass information elements concerning the point-to-point multipoint procedures, which are defined separately

[5] A further collection of new features for another updated UNI version, now called UNI 4.0, has been proposed already in an ATM-Forum signaling subgroup meeting [102].

Connection Management Capabilities				Q.29xy (0)	UNI 3.0	UNI 3.1	UNI 4.0 (5)	SPANS	CMAP	gNET	MSNL-CM
Scenario	Permanent connection	Single		no	no	no	yes	-	-	-	-
		Multiple		no	no	no	yes	-	-	-	-
	Switched connection	Single		yes	yes	yes	yes	-	yes	yes	yes
		Multiple		yes	yes	yes	yes	-	yes	yes	yes
	Symmetric			yes	yes	yes	yes	-	no (7)	-	-
Special Features	QoS support	Bandwidth	Symmetrical	yes	yes	yes	yes	no	yes	yes	-
			Asymmetrical	yes	yes	yes (1)	yes	yes (3)	yes	yes	-
		Delay/Error rate		-	-	-	yes/-	-	-/yes (6)	yes/-	-
	Data flow directions	Unidirectional		yes	yes	yes	yes	yes	yes	yes	yes
		Bidirectional		yes	yes	yes	yes	no	yes	no (9)	yes
	Signalling channel (predefined, static)	In-band		no	no	no	no	no	yes	yes	no
		Out-band		yes	yes	yes	multiple	yes	no	(no)	one
	Broadcast signalling			no	no	no	-	-	yes (8)	yes	-
	Meta signalling			no	no	yes	yes	-	yes	-	yes
	N-ISDN interworking			yes	no	no	yes	-	no	-	-
Maintaining	Static			yes	yes	yes	yes	yes	yes	yes	yes
	Dynamic	Adding	Single	yes	yes (2)	yes (2)	yes	yes	yes	yes	yes
			Multiple Sequential	yes (4)	yes (2)	yes (2)	yes	yes (4)	yes (4)	yes (4)	yes
			Multiple Parallel	yes (4)	yes (2)	yes (2)	yes	yes (4)	yes (4)	yes (4)	yes
		Dropping		yes	yes (2)	yes (2)	yes	(yes)	yes	yes	yes

Note (0): This includes ITU-T Recommendations Q.2931 and Q.298x.
Note (1): In the backward direction the bandwidth equals zero for point-to-multipoint connections.
In the forward direction the bandwidth is identical to all destinations in point-to-multipoint connections.
Note (2): The dynamic behavior is initiator-driven, which is the originator of a point-to-multipoint connection.
Note (3): Since of unidirectional flow of data, an asymmetrical bandwidth support is possible, only.
Note (4): There is no explicit mentioning of either sequential or parallel adding of additional connections.
Note (5): Entries in this column are of preliminary status, since the UNI version 4.0 is still under discussion.
Note (6): The error rate is defined in this case as "cell loss rate".
Note (7): Receive or transmit direction may have different characteristics set individually.
Note (8): A type of "surrogate" has been introduced, defining a certain type of proxy.
Note (9): AAL users may establish bidirectional communication scenarios.

Legend:

yes	existing feature
no	non existing feature
-	no statement possible

Figure 11: A Comparison of Connection Management Capabilities

in the Q.298x Recommendation. Minor but *important differences* comprise (1) the set-up message. The "Called Party Number" and the "Connection Identifier" are mandatory, while the maximum lengths of these parameters are specified. Since the "Connection Identifier" is missing, appropriate error handling procedures are defined. The retransmission of a set-up is defined as optional. Additionally, only two combinations of type of number and addressing plan identifications according to E.164 are supported. (2) For each supported information element the maximum length and the number of valid occurrences are defined. (3) For coding QoS parameters the network specific standard is used. The selection procedures for these parameters are extended with error procedures for any unsupported combination of these parameters.

A.3.3 A Table-based Comparison of Signaling Systems and Protocols

Based on the above mentioned proposals, standards, and recommendations as well as from previous works such as [109], [110], [111], [112], [113], [114], [115], and [116] the table-based comparison has been developed. A main distinction can be initially used for signaling systems:

- Call management and
- Connection management.

These two categories are based on the connection and control issues defined in Subsection A.2.2, using a slightly modified term of management that involves only call and connection related features respectively. This is an important criteria for B-ISDN, since in a multicast environment a call may need a set of connections to different users, *e.g.,* for different media, such as video and text.

Figure 10 summarizes the area of **call management capabilities**.[6] A *scenario* is described by identifying different addressing schemes (such as IP, E.164, or NSAP) that identify unique ATM endpoints and various communications types (such as unicast, multicast, or multipeer). *Special features* include traffic descriptors that are used to specify the type of traffic the application will hand over to the network, a registration of clients that can be used for the integration of a data base for registered end-system's addressing information, and a trace of a call that allows for tracking of the actual physical and virtual path links used within a certain call.

Dependent on the phases of establishing, maintaining, and clearing a call, various differences are identified. During *set-up* time the number of connections per call defines the supported maintenance effort to be handled within the state machines of a signaling protocol. Different ATM transport services (data link layer services in the ISO/OSI Basic Reference Model [3]) can be supported. Additionally, the possibility to (re-)negotiate QoS parameters is important for traffic characteristic variations over the lifetime of a call. *Maintaining* issues concern the static or dynamic behavior of a call. Finally, any *clearing* procedure of a call can be initiated by users or the network.

Figure 11 summarizes the area of **connection management capabilities**. Relevant features encompass a *scenario*, dependent on the type of connection used. Certain *special features* are distinguished. QoS support is defined for bandwidth requirements that are symmetrical or asymmetrical. Further QoS parameters, such as delay or error rates, can be provided. The directions of the data flow is uni- or bidirectional. Signaling support is reflected in the signaling channels provided. A pre-defined and static signaling channel may be used as out-of-band or in-band. Furthermore, broadcast signaling channels and meta signaling can be provided. Finally, interworking with N-ISDN defines upward compatibility with narrowband systems.

Issues of *maintaining* a connection may have different characteristics as well. Distinguishing between static and dynamic connection management, the dynamics are refined in terms of different scenarios for joining or leaving connections. Adding or dropping single or multiple connections (sequential or in parallel) within an existing call is supported.

A.4 Conclusions

Important features of recent standardization work on signaling systems and protocols (such as capabilities, procedures, messages, and information elements) have been briefly described to allow for an insight in signaling tasks. The table-based comparison identifies at a capability-based level main differences of several regarded approaches. Additionally, the following list includes analyzed aspects for a generic signaling system comprising a broader capability- and services-based view of important signaling issues:

- Support of switched and permanent channel connections, including single and multiple connections per call in (multi-)point-to-(multi-)point communication scenarios.
- Definition of basic signaling functions via similar signaling protocol messages, information elements, and procedures for each communication scenario.

[6] In this Figures a *yes* states the existence of the feature, while a *no* states the non-existence. A dash "-" indicates that no reliable information has been identified in the inspected references and no statement can be given.

- Support of uni/bidirectional and symmetric/asymmetric connections concerning QoS requirements, such as bandwidth or error rates for each communication scenario.
- Provision of information for resource negotiation and reservation during connection/call set-up (admission control) and renegotiation during data transfer, including resource monitoring, such as for QoS parameters.
- Using a highly dynamic call model, including detailed characterizations of calls, such as accessabilities or identifications of originator.
- Multiple simultaneous call and connection set-ups for each communication scenario.
- Support of simultaneous joining/leaving of one or multiple calls/connections into/from existing ones (user-driven and network-driven), while supporting mandatory and optional network connections.
- Using a final call and connection clearing.
- Support of all ATM transport services as well as other data link services.
- One statically defined virtual meta signaling channel, providing the facility to dynamically establish broadcast- and/or point-to-point-virtual signaling channels, or associating a pre-defined static point-to-point virtual signaling channel.
- Support of public and private UNI addressing formats for unique identification of ATM endpoints (such as E.164, OSI-NSAP, IEEE 802.6, or IP addresses).
- Definition of client registration mechanisms/ services for exchange of addressing information across UNIs.
- Specification of error detection and recovery (such as alerting and resetting).
- Provision of end-to-end parameter compatibility and identification, *e.g.,* for QoS parameters or signaling parameters, including target platform independent (such as from the operating system or the equipment used) encoding rules for all parameters.
- Independent specification of the signaling system and protocol of the underlying reliable network service (such as with AAL type 3, AAL type 5, SAAL or TCP/IP).
- Independence of the requested service (*e.g.,* multimedia, client/server, or RPC) and of the used protocol for transmitting user data, while supporting different subscriber equipment.

These generic features reflect the needs for a modern signaling system and protocol that will be able to interwork with a wide range of ATM network scenarios and equipment as well. This catalogue should be taken into account for a future development of a *Global Signaling System Architecture*, which is based on distributed system architectures. It includes general mapping schemes and rules for different signaling message types and message contents; on one hand for the integration and experimentation of many different signaling systems and protocols, and on the other hand for signaling simplifications of complex communication and service scenarios. As the variety of existing signaling systems and protocols does not seem to converge to a single signaling system or protocol, an overlayed global signaling system should allow for an interoperable, different ATM networks embracing approach.

B Hierarchical Mapping of Enhanced QoS Parameters Based on OSI Protocols

The following document has been published at:

3rd IEEE Workshop on the Architecture and Implementation of High Performance Communication Subsystems (HPCS'95)
Mystic, Connecticut, U.S.A., August 23 – 25, 1995; pages 43 – 46.

Hierarchical Mapping of Enhanced QoS Parameters Based on OSI Protocols

Burkhard Stiller [7]
University of Cambridge, Computer Laboratory
New Museums Site, Pembroke Street
Cambridge, CB2 3QR, England, U.K.
Phone: +44 +1223 334476, FAX: +44 +1223 334678
E-Mail: Burkhard.Stiller@cl.cam.ac.uk

B.1 Introduction

Quality-of-Service (QoS) defines the way end-systems and networks deal with the description of applications requirements in support of their communication demands. QoS parameters define specific communication characteristics including different values. Traditionally, these have been used in communication protocol specifications without sufficient relation to the protocol itself, and sometimes only a set of vague and separately defined QoS parameters exists. The problem with this situation is that in layered protocol stacks, such as within the ISO/OSI (International Organization for Standardization/Open Systems Interconnection) Basic Reference Model, no rules are provided for mapping higher layer QoS onto lower layer QoS.

Therefore, this paper proposes a hierarchical QoS mapping approach based on OSI protocols. The approach taken includes enhanced QoS and leads to general mapping functions between QoS parameters.

B.2 Architecture

An OSI protocol stack comprises seven layers as defined in the ISO/OSI Basic Reference Model [3]. Each of them includes a limited set of QoS parameter definitions. Besides desirable consequences, such as efficiency, synchronization, or functional redundancy, the main consequence of these multiple-layer architectures concerns the urgency of QoS mapping between neighboring layers. As depicted in Figure 12(a) for the 7-layer OSI architecture, 7 different QoS mapping functions have to be defined. The application resides on top of the considered protocol stack and has to "map" their requirements into the QoS structure of the highest layer. However, this "mapping" is of a different nature than the following ones, since it involves the initial specification of application requirements within a set of QoS parameters. Therefore, this may be regarded as only a simple specification of application requirements. The other QoS mapping has to take place at every interfaces of layers N and N-1.

A different and more practical approach is presented in Figure 12(b) and significantly reduces the number of QoS mapping functions. In this case, seven ISO/OSI layers have been reduced to three components [49]. Based on this architecture, three relevant groups of QoS mapping functions have been developed (cf. Section B.4). Besides the application itself, the *Application Support* is responsible for application-oriented protocol processing tasks (such as presentation, remote operations, or synchronization), while the *Transport System* is responsible for transport-specific tasks (such as (non-)reliable transport of user data or segmentation).

[7] The author is currently on leave from Universität Karlsruhe, Institut für Telematik, D-76128 Karlsruhe, Germany, and is sponsored by the Commission of the European Communities as a Research Fellow under the Human Capital and Mobility Scheme (RG 19372).

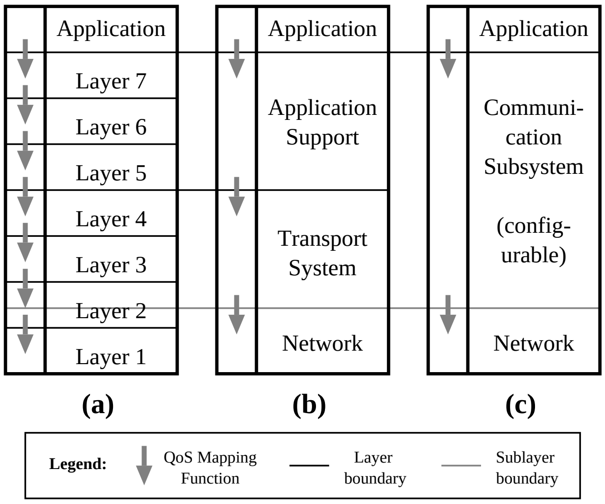

Figure 12: Alternative Architectures

As depicted in Figure 12(c), another possible approach has been proposed in [19], [20], relying on protocol configuration and applying an internal QoS mapping scheme. In this case, application requirements are specified by QoS parameters (which are handled by the application, the protocol, and the end-system) and are finally mapped onto protocol building-blocks and network-dependent parameters.

B.3 QoS Parameter

A general distinction between QoS requested from the application support (application-oriented), QoS maintained from the transport system (transport system-specific), and QoS provided by the network (network-dependent) is useful. These categories are used within the following mapping approach, while *quantitative* QoS parameters specify the performance-oriented behavior of the service and *qualitative* ones specify the functional-oriented issues of the service. Besides the considered OSI protocols of layers 4, 5, 6, [117], [118], [119] and the ITU-T (International Telecommunications Union) parameter definitions for layer 1 and 2 [5], [13], [120] additional useful extensions of QoS parameters — based on [20] and [15] — are added.

Due to the fact that the application, the transport system, and the network will interact by means of well-defined interfaces, a hierarchical structure of these QoS parameters is defined. This is the decomposition of the originally requested *Communication SerVice (CSV)* into separately requestable *Application SerVices (ASV)* in the upper level, which are strengthened by the set of *Application-oriented Quality-of-Service Parameters (AQoS)*. They in turn are supported in the medium level by the set of *Transport SerVices (TSV)* and a set of *Transport Quality-of-Service Parameters (TQoS)*. Additionally, a set of *Network Parameters (NP)* in the lower level of the hierarchy are defined. This hierarchical structure of CSV, ASV, AQoS, TSV, TQoS, and NP will be dealt with separate generic mapping functions, depicting the service-user and service-provider relation.

Application-oriented QoS — Applications specify their requirements in ASV and AQoS, that are provided by the application support [121]. Within the considered OSI protocols, application support is initialized using parameters as defined in Table 7 for ASV and Table 8

ID	ASV Parameter
AS.1	communication relation management
AS.2	data transfer
AS.3	synchronization
AS.4	dialogue control
AS.5	exception handling
AS.6	remote operations
AS.7	remote file access and management
AS.8	remote terminals

Table 7: Application Services

ID	Sub-ID	AQoS Parameter
AQ.1		*representation independence*
AQ.2		*error handling*
	AQ.2.1	data corruption
	AQ.2.2	data loss
AQ.3		*type of service*
	AQ.3.1	normal
	AQ.3.2	expedited
	AQ.3.3	multicast (parties)
	AQ.3.4	isochronous
	AQ.3.5	asynchronous
	AQ.3.6	(un-)confirmed
AQ.4		*dependencies*
	AQ.4.1	independent of token or context
	AQ.4.2	capability
AQ.5		*requestor*
AQ.6		*security*
AQ.7		*parameter types and sizes*
	AQ.7.1	amount of data
	AQ.7.1	throughput
AQ.8		*segmented delivery*
AQ.9		*time restrictions*
	AQ9.1	maximum response time

Table 8: Application Quality-of-Service

for AQoS. The services aspects (ASV) are highly related to the Session, Presentation, and Application Layer according to the Basic Reference Model, while certain performance-oriented aspects (AQoS) are covered as well. These parameters are mainly maintained and provided within the application support.

Transport System-specific QoS — Besides the upper level application point of view the transport system-specific view takes account by transport tasks and protocols. Especially, as depicted in Table 9, transport services (TSV) encompass various parameters that deal with a single or multiple user-data streams. These qualitative parameters characterize a data stream in a coarse-grained manner. The refinement of this information is possible using the transport QoS, as presented in Table 10. A wide range of parameters allows for the specification of additional values (dependent on the specific TQoS such as maximum, average, and minimum) that have to be maintained and provided within the transport system.

Network-dependent QoS — Finally, the lowest level maintains and provides parameters, called *Network Performance (NP)*. Generally available parameters are included in Table 11, while specifically network dependent parameters, *e.g.,* as for an ATM (Asynchronous Transfer Mode) or FDDI (Fiber Distributed Data Interface) network, are exemplified in Table 12.

ID	Sub-ID	TSV Parameter
TS.1		*inter-stream synchronization*
TS.2		*data stream update*
TS.3		*stream multiplexing*
TS.4		*error tolerance*
	TS.4.1	guaranteed ordered delivery
	TS.4.2	data loss
	TS.4.3	replication of data
	TS.4.4	corruption of data
TS.5		*guaranteed group delivery*
TS.6		*data transfer*
	TS.6.1	normal
	TS.6.2	expedited
TS.7		*multiplexing*
		data unit manipulation
TS.8		*maximum data unit size*
TS.9		*segmented delivery*
TS.10		*separation/concatenation*

Table 9: Transport Services

B.4 QoS Parameter Mapping

Combining the above presented views, a set of three *generic mapping functions (GMF)* has been defined. They delineate the operability of assisting communication service requests, by mapping various services and their qualities onto lower level services and qualities. This process has to be managed within the application support and transport system as well. Each GMF is defined in the following list including formal parameters, while the result may have no "output" (denoted by $\emptyset$), or *e.g.,* a single or a set of services and QoS parameters, respectively (identified by *):

ID		**Generic Mapping Function**
1.		$CSV \longrightarrow (ASV^*, AQoS^*)$
2.	a	$ASV \longrightarrow TSV^*$
	b	$AQoS \longrightarrow TSV^*$
	c	$AQoS \longrightarrow TQoS^*$
3.	a	$TSV \longrightarrow \emptyset$
	b	$TQoS \longrightarrow NP^*$

GMF 1 depicts that the user communication service request is characterized by a mandatory set of ASV and AQoS parameters. As mentioned before, this function is "performed" by the application, while specifying the application requirements in the appropriate representation. Depending on the protocol features of the application support, these parameters are directly supported or mapped onto lower level parameters.

Afterwards, some of these parameters are presented to the transport system. Therefore, the second group of GMFs (2.a, 2.b, and 2.c) is distinguished by certain degrees of fine-granular mappings. Sometimes parameters have not to be mapped onto lower level parameters (cf. Table 13), since the service is already provided by the application support (depicted by $\emptyset$).

Finally, the third group of GMFs (3.a and 3.b) maps transport system-specific parameters onto network performance parameters (cf. Table 14). Depending on the physically available network, different NP mappings have to be performed, according to Table 12, since NP parameters may differ significantly.

ID	Sub-ID	TQoS Parameter
TQ.1		*throughput*
	TQ.1.1	burstiness
TQ.2		*delay*
	TQ.2.1	transit delay
	TQ.2.2	connection establishment delay
	TQ.2.3	connection release delay
	TQ.2.4	response time
TQ.3		*rate*
TQ.4		*jitter*
TQ.5		*data corruption*
	TQ.5.1	bit error rate
	TQ.5.2	packet error rate
TQ.6		*data loss*
	TQ.6.1	bit loss rate
	TQ.6.2	packet loss rate
TQ.7		*residual error rate*
TQ.8		*probabilities (prob.)*
	TQ.8.1	connection establishment prob.
	TQ.8.2	connection release prob.
	TQ.8.3	connection abort (resilience) prob.
	TQ.8.4	unacceptable transmission prob.
TQ.9		*costs*
	TQ.9.1	priorities

Table 10: Transport Quality-of-Services

ID	Sub-ID	NP Parameter
NP.1		*throughput*
NP.2		*transmission delay per medium length*
NP.3		*bit error rate*
NP.4		*costs per medium access adapter*
NP.5		*topology specific parameters*
	NP.5.1	transmission frame length
	NP.5.2	number of priority levels
	NP.5.3	capability of isochronous traffic
NP.6		*costs*
	NP.6.1	medium access costs

Table 11: Network Performance

B.5 An Example

The mapping for a 5-party multicast voice communication with dialogue control, an expedited data service, a maximum data loss of 10^{-7}, and a guaranteed 1 Mbit/s user throughput per party is illustrated.

Applying GMF 1 in the first step results in ASV parameters as "data transfer" and "dialogue control". The AQoS parameters are set to "multicast (parties = 5)", "expedited", "isochronous (voice)", "data loss (10^{-7})", and "throughput (1 Mbit/s)", including the specifically specified values in brackets.

In the second step the application support maps parameters that can not be supported directly, to transport system parameters. The ASV "data transfer" is mapped by GMF 2.a(2) (Table 13) onto TSV "data transfer". AQoS "multicast (parties = 5)" is mapped onto TSV "guaranteed group delivery" using GMF 2.b(7) (Table 13).[8] The ASV "dialogue control" will be handled inside the application support, which is depicted by the empty-set

[8] Note, one possibility to implement multicast is to duplicate packets as often as needed within the transport system and transmitting all of them to specified destinations. Better solutions including more application knowledge may result in further parameters that can be handed to the transport system.

Network	Specific NP Parameter	Value
ATM	Cell Delay Variation	negotiable
	Average Cell Rate	negotiable
	Peak Cell Rate	negotiable
	Cell Length	53 Byte
	Cell Error Ratio	fixed
	Cell Loss Ratio	fixed
FDDI	Packet Length	9000 Byte
	Target Token Rotation Time	negotiable
	Bit Error Rate	fixed
Ethernet	Packet Length	1524 Byte
	Bit Error Rate	fixed

Table 12: NP Parameter in Different Networks

symbol $\emptyset$ (GMF 2.a(4), Table 13). The AQoS "isochronous (voice)", "data loss (10^{-7})", and "throughput (1 Mbit/s)" will be mapped directly onto the TQoS "jitter (voice)", "data loss (10^{-7})", and "throughput (1001 kbit/s)" respectively, using GMF 2.c(3), (2), and (4). These values are treated as maximum values. Note that due to a maximum of about 80 bytes per header/trailer information per protocol data unit the throughput provided by the transport system increases.

In the final step TQoS are mapped onto NP, since all TSV will be provided inside the transport system itself (GMF 3.a(1), Table 14). Applying GMF 3.b(5), (7), and (1), "jitter (10 ms)", "data loss (bit error rate = 10^{-7})", and "throughput (1002 kbit/s)" are mapped onto NP "isochronous traffic", "bit error rate", and "throughput", respectively. Note that inside the transport system "voice" has been significantly refined, using voice-specific defaults as physical applicable values (10 ms), while "data loss" has been identified as the bit error rate parameter. An additional increase of the required throughput is again due to the bigger protocol data units carrying protocol control information.

B.6 Conclusions

The presented set of *Generic Mapping Functions* provides guidelines for an appropriate QoS mapping in OSI protocols. Details have to be added for each separate QoS parameter mapping, according to relevant information, such as the protocol data unit size, processing delays of the protocol, or network characteristics as the example depicts. Furthermore, it is important to use resource reservation techniques to provide the requested service in terms of guarantees or statistical bounds. Therefore, it is intended to apply the concept of mapping functions to transform QoS parameters into possibly different representations or values that may be used within resource reservation techniques or scheduling algorithms.

Acknowledgements: I am indebted to G. Solvie for discussing mapping problems and K. van der Merwe for proofreading and commenting on earlier versions of this paper.

GMF 2.a	ASV	TSV
(1)	AS.1	TS.4.1, TS.4.3, (∅)
(2)	AS.2	TS.6
(3)	AS.3	TS.1
(4)	AS.4	∅
(5)	AS.5	∅
(6)	AS.6	TS.6
(7)	AS.7	TS.6
(8)	AS.8	TS.6
GMF 2.b	**AQoS**	**TSV**
(1)	AQ.1	∅
(2)	AQ.2	TS.4
(3)	AQ.2.1	TS.4.2
(4)	AQ.2.2	TS.4.4
(5)	AQ.3.1	TS.6.1
(6)	AQ.3.2	TS.6.2
(7)	AQ.3.3	TS.5
(8)	AQ.3.5	TS.6
(9)	AQ.3.6	TS.4
(10)	AQ.4.1.	∅
(11)	AQ.4.2.	∅
(12)	AQ.5	∅
(13)	AQ.6	∅
(14)	AQ.8	TS.8, TS.9, TS.10
GMF 2.c	**AQoS**	**TQoS**
(1)	AQ.2.1	TQ.5, TQ.5.1, TQ.5.2
(2)	AQ.2.2	TQ.6, TQ.6.1, TQ.6.2
(3)	AQ.3.4	TQ.4
(4)	AQ.7.1	TQ.1, TQ.1.1
(5)	AQ.7.2	TQ.1
(6)	AQ.9.1.	TQ.2.4

Table 13: GMFs 2.a, 2.b, and 2.c

GMF 3.a	TSV	∅
(1)	TS.1 to TS.10	∅
GMF 3.b	**TQoS**	**NP**
(1)	TQ.1	NP.1
(2)	TQ.1.1	NP.1
(3)	TQ.2.x	NP.2
(4)	TQ.3	NP.1
(5)	TQ.4	NP.5.3
(6)	TQ.5.x	NP.3
(7)	TQ.6	NP.3
(8)	TQ.7	NP.3
(9)	TQ.8.x	∅
(10)	TQ.9	NP.5.4, NP.6
(11)	TQ.9.1	NP.5.2, NP.6

Table 14: GMFs 3.a and 3.b

C PROCOM — A Protocol Configuration Manager

The following document has been published at:

1st International Workshop on
High Performance Protocol Architectures (HIPPARCH'94)
INRIA Sophia-Antipolis, France, December 15 – 16, 1994;
Session 3, Paper 8.

PROCOM: A Protocol Configuration Manager in the Function-based Communication Subsystem

Burkhard Stiller [9]
University of Cambridge, Computer Laboratory
New Museums Site, Pembroke Street
Cambridge, CB2 3QR, England, U.K.
Phone: +44 +1223 334476, FAX: +44 +1223 334678
E-Mail: Burkhard.Stiller@cl.cam.ac.uk

C.1 Motivation and Introduction

The variety of upcoming applications in terms of their performance and Quality-of-Service (QoS) requirements is extending today. Next to almost well-known applications, such as teleconferencing, audio- and video-transmissions, even more contemporary ones, such as medical imaging, Video-on-Demand, and interactive tutoring systems, are introduced and applied to existing networks. On the contrary, traditionally data-oriented applications, such as file transfer and remote login, are considerably different in terms of their QoS requirements. The consequences of this evolution effect the architectures of end-systems, *e.g.*, workstations that have to be capable of maintaining all different kinds of multi-media data, and intermediate-systems as well. Since traditional models of protocol processing do not show a well suited behavior [122], [123], solutions for achieving a better behavior have to be developed.

The reasons for this behavior are twofold. On one hand, services offered on top of traditional end-systems are quite restricted in the sense that only very simple QoS parameters can be specified. Examples can be found in existing transport protocols, *e.g.*, TP4 [117], [8] and TCP [124]. They offer a limited number of QoS parameters: a jitter parameter is missing. On the other hand, performance of traditional end-systems is not only dependent on transport protocols specifications [125], moreover, multiple different factors are involved, *e.g.*, protocol processing in software, redundant protocol functionality in multiple layers, and data copying between different address spaces.

To overcome these misbehaviors various approaches are possible. A categorization of these alternative approaches can be done by distinguishing their main targets, while further refining performance issues into implementation-related and architecture-related ones:

- avoiding the service gap,
- avoiding problems of layer-oriented architectures, and
- avoiding the performance bottleneck.

These three different areas of research have been tackled often in separate approaches. Examples for bridging the service gap are the introduction of new services for request/response type of traffic and transaction-oriented traffic [126] or acknowledged datagram services [10]. The 3-component model [49] avoids problems with layered architectures and introduces much broader components to overcome intra-layer communication and to escape multiple sub-layers. Examples for avoiding the performance bottleneck are manifold as well: efficient implementation of protocols ([127], [128], [129], [130], [131] and [132]) on one hand and the development of light-weight protocols, such as VMTP [133] or XTP [10], on the other hand are possible. Furthermore, operating system support for protocol processing has been developed resulting in the design of the *x*-kernel [123].

Nevertheless, an important target of a communication subsystem is to efficiently serve various applications with different requirements on top of a network using appropriate communication protocols. Therefore, a flexible and dynamic approach has been developed. The *Function-based Communication Subsystem (F-CSS)* [19] offers a framework to flexibly configure protocols according to application

[9] The author is currently on leave from Universität Karlsruhe, Institut für Telematik, D-76128 Karlsruhe, Germany, and is sponsored by the Commission of the European Communities as a Research Fellow under the Human Capital and Mobility Scheme (RG 19372).

needs providing a service-integrated communication subsystem architecture based on the functional decomposition of communication protocols into protocol functions and mechanisms. Comparable but different approaches in terms of their main research aspects are ADAPTIVE [48] or Da CaPo [47].

C.2 Communication Protocol Configuration

Regarded as an entire architecture, F-CSS includes certain features and tools [19], [20]. These features, *e.g.*, fine-grained QoS specification, defined by qualitative and quantitative criteria, flexibility, and automated protocol configuration, are supported and provided by tools. These tools are involved in the set-up, maintenance, and termination of so-called data streams and sessions. This Section's focus is rather aligned towards the task of configuring communication protocols after a brief introduction into the communication subsystem environment. This configuration task is processed by the *Protocol Configuration Manager (PROCOM)* and supported by a formal language, called *Function-based Protocol Configuration Language (F-PCL)*, amongst others to specify configuration conditions for separate protocol functions and mechanisms. Additionally, a *System Resource Manager (SYSREM)* is used for maintaining information on local end-system and network resources.

C.2.1 The Communication Subsystem Environment

A communication subsystem environment is composed out of 3 parts. The first part comprises an *Application Service Interface* that is used for defining application requirements in terms of Quality-of-Service parameters. Underneath this interface the *Communication Subsystem Architecture* is embedded including communication protocols. The lowest level defines a *Network Service Interface* specifying available network services.

The Function-based Communication Subsystem is specifically based on a functional decomposed Communication Subsystem Architecture extended with a fine-grained specification of an Application Service Interface. On one hand, this architecture encompasses *Protocol Machines* behaving as active components while being responsible for the protocol processing tasks. On the other hand, a *Protocol Resource Pool* forms the static component containing, *e.g.*, formally specified *Protocol Functions and Mechanisms*. Furthermore, a *Session Model* is used to define a hierarchical structure of user data flows. A *Data Stream* is the logical view of the lower level unidirectional user data flow, *e.g.*, an audio stream or a robotic control information sequence. Each of these unidirectional data streams are supported by a specifically configured protocol machine, consisting out of a number of protocol functions and mechanisms that can be described as *Protocol Machine Graphs*. Multiple data streams may be combined into a *Session* that defines the higher level of the hierarchy. The reason for this concatenation is the possibility to form a unique unit of data streams, such as for inter-stream synchronization requirements.

C.2.2 Configuration Prerequisites

In order to configure communication protocols in the F-CSS environment properly certain prerequisites have to be met. First of all, a precise and unique specification of application requirements must be present. These requirements are specified in terms of Quality-of-Service parameters and values, respectively. Secondly, network and system resources have to be specified as well. Finally, a formal language allows for the definition of configuration conditions and further information that are used within the configuration algorithm.

Application Service Interface – Application requirements strictly determine any required protocol communication service. A sufficient specification of these requirements can be done within two sets of QoS parameters: *Quantitative Criteria* and *Qualitative Criteria* [19]. These two distinct categories are used to define requirements at a different level. Quantitative criteria express application requirements in terms of certain quantities, such as "bits-per-second" or "errors-per-unit". These parameters can be specified by three distinct values, (1) an *average*, (2) a *threshold*, and (3) a *useful* value. These criteria encompass the following ones: throughput, delay, response time, jitter, rate, data corruption, and data loss. Instead, qualitative criteria specify the actual delivery of user data at the destination's Application Service Interface, mostly in terms of attributes, such as "reliability" or "multicast". These terms are specified in a binary manner (yes/no semantics). A number of criteria is defined, such as inter-stream synchronization, session update, ordered delivery of data, guaranteed group delivery, loss of data, replication of data, corruption of data, intra-stream synchronization, syntax selection, security requirements, limited data unit size, and segmented delivery. Additionally, *Application Service Classes* may be defined, depicting a pre-defined general category of application requirements, *e.g.*, "unreliable real-time" or "reliable non real-time" services. Current extensions of the Application Service Interface in terms of intermediate criteria and Types-of-Service, such as best-effort and guaranteed service, is under development [134].

Network and System Resource Parameters – As mentioned earlier, the *System Resource Manager (SYSREM)* maintains information on network resources (such as available error rates, maximum delays, isochronous services, multicast support, or maximum data units size) and on local end-system resources (such as network adapter bandwidth, memory, sequence guarantees, or CPU time). These information are used to restrict possible configurations of communication protocols according to available resources. To secure a useful degree of always updated novelty of these information a monitor may be used to monitor certain parameters as well as quantitative criteria of the Application Service Interface.

Function-based Configuration Language – A formal description technique acts as an auxiliary material concerning the static description of protocol functions and mechanisms as well as their use for dynamic configuration tasks. Therefore, the declarative *Function-based Configuration Language (F-PCL)* has been defined, since the use of formal languages (such as SDL, Estelle, or LOTOS) and graphical representations (such as petri-nets and flow charts) would constitute for these configuration purposes a degree of complexity and size that is not necessary for the entire configuration-oriented specification of protocol functions and mechanisms as well as their inter-relationships. F-PCL supports the specification of multiple protocol functions and mechanisms in terms of configuration conditions, dependencies, and needed resources. These informations are arranged in two levels: the top level for protocol functions (function body) and the lower level for protocol mechanisms (mechanism body). Next to regular identifiers, *e.g.*, for names of protocol functions, so-called *Symbolic Variables* are integrated. These are pre-defined identifiers for each of the above mentioned quantitative and qualitative criteria, such as `DSB_Ordered_Delivery` or `DSQ_Throughput_T` as well as for every single network and system resource parameter, such as `NRQ_Jitter_Max` or `LSQ_Memory`.

Figure 13 exemplifies briefly a specification of a single protocol function out of the Protocol Resource Pool with one mechanism. Important to recognize are the **REQUIRED**- and **FORBIDDEN**-configuration conditions. They are specified as logical expressions that are evaluated during the configuration process to `TRUE` or `FALSE`. Depending on the result, a certain function — if the condition is located in the functions body of F-PCL — or a specific mechanism — if located in the mechanisms body of F-PCL — will be selected as a part of the configured protocol machine. Within these logical expressions some arithmetical expressions may be used as well, including equation or comparison operators. The main ingredients of these expressions form symbolic variables that are defined for each QoS parameter and network and system resource parameter. **SENDER**- and **RECEIVER**-constructs specify the use of a protocol mechanism as part of the sending (*e.g.*, the protocol mechanism "Selective_Retransmission") or the receiving (*e.g.*, the protocol mechanism "Negative_Selective_Acknowledgement") part of the protocol machine.

C.2.3 The Protocol Configuration Manager PROCOM

The Protocol Configuration Manager needs information on application requirements and on resources to be used, which are handed over to PROCOM by the Application Service Interface and the System Resource Manager. Furthermore, PROCOM is dependent on the Protocol Resource Pool and its included protocol function and mechanism specifications. The task of configuration is independent of any underlying hardware architecture used to execute the communication protocol, since the result of a PROCOM configuration is a protocol machine graph — a formal representation of a configured

```
DEFPFT Retransmission
  {
  REQUIRED  (!DSB_Data_Loss) | (NSQ_Bit_Data_Loss > DSQ_Bit_Data_Loss_T ) |
            (DSQ_Packet_Data_Loss_T = 0);
  FORBIDDEN (DSQ_Jitter_T <= ((2 * NRQ_Delay_Max) + PMQ_Expected_Proc_Time_Recv);
  DEFPFM Selective_Retransmission
    {
    SENDER     TRUE;
    RECEIVER   FALSE;
    REQUIRED   NSQ_Bit_Data_Loss > Bit_Error_Rate_Limit;
    FORBIDDEN  FALSE;
    INPUT      SID, DID, sequence_number;
    OUTPUT     SID, DID, buffer_info;
    CONFIGURE  NONE;
    PRED       Negative_Selective_Acknowledgement,
               Positive_Cumulative_Acknowledgement;
    SUCC       Window_Flow_Control, Rate_Flow_Control, NSI_Out_to_below;
    TIME       100 * ySec;
    MEMORY     324 * Byte;
    CODE       ret_sel.lku;
    }
    /* other mechanism alternatives may follow here */
  }
```

Figure 13: An Example for the Specification of a Protocol Function in F-PCL

protocol machine consisting out of multiple protocol mechanisms. PROCOM is implemented by using a configuration algorithm defined by eight separate steps. These steps are shortly outlined as follows.

Step 1 — Configuration Preparation: Data stream requirements expressed in terms of quantitative and qualitative criteria are specified by the application. This specification forms one important basis for the selection of protocol functions and mechanisms. All criteria values are associated with a certain symbolic variables that are used in the F-PCL specification for defining configuration conditions. An example is the following one, where the maximum throughput of the application is set to 15 Mbit/s.

```
DSQ_Throughput_T := 15
```

Step 2 — Determination of Available Resources: PROCOM requests information about available system and network resources from the local SYSREM. An optional request may be handed over to one or multiple remote SYSREMs of the destination(s) — depending on the uni- or multicast communication scenario — to receive information as well[10]. The information provided on the first inquiry at least are needed for step 4 and are made available under symbolic variables, too. An example is the following one, where the maximum bandwidth of the local network adapter is set to 100 Mbit/s and for the remote network adapter is set to 80 Mbit/s.

```
LRQ_Net_Interface_Bandwidth := 100
RRQ_Net_Interface_Bandwidth := 80
```

Step 3 — Feasibility Check: After having received all necessary information form the application in terms of data stream requirements and available system and network resources from SYSREM a thorough analysis of the configuration request is possible. This check of feasibility tests a set of expressions that is used to specify certain pre-conditions to be fulfilled. These pre-conditions are stored separately from the Protocol Resource Pool, but follow a similar syntax and evaluation as for

[10] Beyond the scope of this paper is the description of the use of a signaling protocol between PROCOM and one or multiple remote SYSREMs. Additional tools in F-CSS, especially the Session Manager [20], allows for sending one or multiple remote resource requests and transferring the answer(s) back to the initiating PROCOM.

REQUIRED- and **FORBIDDEN**-conditions. Therefore, each symbolic variable for data stream quality criteria and resource information values may be combined arithmetically or logically — depending on the type of the variable, such as the following examples present.

```
DSQ_Throughput_T * Overhead_Factor > LRQ_Net_Interface_Bandwidth
DSQ_Delay_T > NRQ_Delay_Max
```

The use of this feasibility check is to avoid unnecessary configuration efforts, in particular situations where it is obvious that a protocol suited for the needs of the application cannot be configured with currently available resources. An obvious situation for a configuration failure is the available network bandwidth `LRQ_Net_Interface_Bandwidth`, which might be too low to provide the required application throughput requested by the user `DSQ_Throughput_T` multiplied by an estimated overhead factor for the future protocol machine. Additionally, the maximum acceptable delay for an application `DSQ_Delay_T` should always be greater than the maximum delay of the network `NRQ_Delay_Max`. The configuration will be continued only if the evaluation of all specified pre-conditions indicate a positive result.

Step 4 — Selection of Protocol Functions and Mechanisms: The selection of required protocol functions and mechanisms is defined as a two-stage process. For each function in the Protocol Resource Pool the need for using it in the actual protocol is determined by evaluating **REQUIRED**- and **FORBIDDEN**-conditions for each protocol function and mechanism as well. If PROCOM detects that a certain protocol function is required for the current protocol — which is logically similar to the evaluation of at least one **REQUIRED**-condition to `TRUE` and all **FORBIDDEN**-conditions to `FALSE` —, an afterward selection must be made among available mechanisms of this protocol function, using additional conditions in the mechanisms body of F-PCL. As an example the following expression depicts the **REQUIRED**-condition of a segmentation function:

REQUIRED `DSQ_Max_Data_Unit_Length + PMQ_Max_Header > NSQ_Max_Data_Unit_Length;`

Only the sum of the maximum length of any application data unit `DSQ_Max_Data_Unit_Length` and the maximum header length of the protocol data unit `PMQ_Max_Header` is greater than the maximum supported data unit length of the network `NSQ_Max_Data_Unit_Length`, a segmentation function is necessary. A second example depicts the configuration conditions for a checksum validation function.

REQUIRED `!DSB_Data_Corruption | (DSQ_Bit_Data_Corrupt_T = 0);`
FORBIDDEN `NSQ_Bit_Data_Corrupt > DSQ_Bit_Data_Corrupt_T;`

Step 5 — Protocol Machine Graph Construction: Once the selection of protocol functions and mechanisms is completed, PROCOM constructs protocol machine graphs. These graphs describe internal communication relations between selected protocol mechanisms. In this step of the configuration process **SUCC** (successor) and **PRED** (predecessor) expressions of the protocol mechanism description in a F-PCL description are used. These expressions determine the relevant processing order of all selected mechanisms.

Step 6 — Protocol Machine Graph Consistency Check: A consistency check is required to ensure that neighbouring protocol mechanisms of a protocol machine graph have compatible **INPUT**- and **OUTPUT**-parameters as specified in the F-PCL description. The graph is considered to be correct if each input parameter for every mechanism may be found in the output parameter list of one of the direct predecessors. If these requirements are not met completely, the constructed protocol machine might not function properly, since left open input parameters will never receive a real incoming value. Therefore, in this case the configuration approach must be considered as a failure.

Step 7 — Configuration Parameter Evaluation: The internal behavior of protocol mechanisms may be influenced by configuration parameters. This allows for the specification of, *e.g.*, buffer sizes or initial window sizes, while using arithmetical expressions including all different kinds of symbolic variables and identifiers for constants as well. The actual value may to be calculated by PROCOM or defined by default.

Step 8 — Submission of Protocol Machine Descriptions: Finally, PROCOM hands the protocol machine description for the configured communication protocol — including the protocol machine graph and local administration information — to the instantiation tool in an F-CSS station, which is the Code Configuration Manager (CODCOM) [135] that maps the protocol machine description on the underlying hardware architecture and prepares the protocol machine for execution[11].

C.2.4 Reconfiguration Issues

F-CSS supports the dynamic reconfiguration of communication protocols as well to meet changing application requirements, changing network states, and changing system resources. An analysis of possible reconfiguration scenarios has shown that even minor changes to the configuration information, *e.g.*, data stream specifications or system resource data, may cause major changes in the protocol machine graph, *e.g.*, adding or deleting protocol mechanisms. An example is the occurrence of bit errors in the network and the addition of a checksumming mechanism into the running protocol machine. Another example is an exchange of the window-based flow control mechanism into a rate-based mechanism as a result of the detection of too bursty traffic arriving at the destination and violating the threshold rate of arriving data.

Therefore, different choices of reaction in a reconfiguration process may be taken into account and used. They depend next to PROCOM tasks in terms of to be gained performance on the hardware-dependent tool CODCOM as well:

- Adaptation of one or multiple configuration parameters (PROCOM),
- Reordering of protocol mechanisms in the protocol machine graph, while adding or deleting certain new/old mechanisms (PROCOM) and recycling instantiated executable code on the hardware-platform (CODCOM), or
- Complete new configuration based on newly available information (PROCOM) and instantiation on the hardware-platform (CODCOM).

C.3 Performance Results on the Configuration Algorithm

The configuration algorithm used within the Protocol Configuration Manager has been implemented in an INMOS Transputer-based environment, like the entire Function-based Communication Subsystem, using the C++ dialect "Glockenspiel" [136]. Performance evaluations of the algorithm are depicted on a step by step basis, following Subsection's C.2.3 description.

In order to reduce the calculation overhead within the algorithm for arithmetical and logical expressions, which are included in configuration conditions, a tokenized representation for all defined symbolic variables is used. These tokens are stored in a token table and are derived once at boot and initialization time of F-CSS while reading the actual valid Protocol Resource Pool including all protocol function and mechanism specifications.

Table 15 depicts performance measurements of the configuration algorithm for each step separately in a randomly chosen situation. In this case, the Protocol Resource Pool consisted out of 8 different protocol functions and 14 different mechanisms. In step 1 the request for a protocol machine configuration have been received and all current QoS parameter values are stored in the token table (0.7 ms). The request of available local network and system resources has been performed in the current scenario in 12.6 ms. The time for the feasibility check in step 3 is heavily dependent on the number of conditions to be checked as well as the length and complexity of their internal structure (*e.g.*, "and", "or", "multiplication" in the logical expressions). Again, the time for step 4 relies on the number of to be evaluated **REQUIRED**- and **FORBIDDEN**-conditions, here done in 18.2 ms. As it can be seen, the highest amount of time is needed for the construction of the protocol machine graph out of the previously selected protocol mechanisms for the sender part (81.9 ms) and the receiver part (66.5 ms) — the sender part consists in this scenario out of 8 mechanisms and the receiver part out of

[11] In contrast to PROCOM tasks CODCOM processes hardware platform-dependent tasks, such as mapping protocol mechanisms into processes, placing them on one or multiple processors, and downloading the executable code.

Step	Common Parts	Sender Part	Receiver Part
Step 1: Preparation	0.7	-	-
Step 2: Resource Requests	12.6	-	-
Step 3: Feasibility Check	4.6	-	-
Step 4: Selection	18.2	-	-
Step 5: Construction	-	81.9	66.5
Step 6: Consistency Check	-	1.1	0.6
Step 7: Parameter Evaluation	-	13.2	6.8
Step 8: Submission	-	3.3	2.3
Partial Sums	*36.1*	*99.5*	*76.2*

Table 15: Performance Results of the Configuration Algorithm (Times in ms)

6 mechanisms. The remaining steps as the consistency checks, the configuration parameter evaluation, and the submission of the protocol machine descriptions to CODCOM are performed in the sum of 17.6 ms for the sender part and in 7.7 ms for the receiving part.

Therefore, the time needed for the complete sender and receiver protocol machine configuration in this scenario takes 211.8 ms as the overall sum of the partial sums in Table 15. Important to recognize is the potential possibility to process steps 5 to 8 for the sender and receiver part in parallel, since they are completely independent from each other. These numbers indicate that the configuration tasks based on this configuration algorithm can be done in limited amount of time. In reconfiguration scenarios this time will differ dependent on the above mentioned choices (cf. Subsection C.2.4), which are currently under detailed study. A further gain of configuration time can be reached if pre-defined Application Service Classes are used, where the corresponding protocol machine graphs do not have to be configured any more, only the instantiation process by CODCOM has to be performed.

C.4 Summary and Conclusions

In summary, throughout this paper a configuration algorithm, embedded into a suitable communication subsystem framework has been outlined. The Protocol Configuration Manager configures communication protocols reflecting dedicated application requirements. Prerequisites encompass a fine-grained Application Service Interface expressed in terms of qualitative and quantitative criteria. The separation of the configuration task (algorithm) and configuration condition specifications (in F-PCL) leads to a flexible system approach, which may be adapted and extended in a convenient manner.

Concluding this work, the presented algorithm, the prototypical implementation, and the performance results as well show a reasonable behavior of the system components. This includes acceptable processing times dependent on defined configuration conditions and a defined Protocol Resource Pool.

Acknowledgements

The author would like to thank A. Tantawy and M. Zitterbart for their close cooperation in developing the original framework of F-CSS and O. Steinmeier for implementing most important parts of PROCOM.

D Protocol Configuration and Interoperability — A Case Study —

The following document has been published at:

IEEE Singapore International Conference on Networks and International Conference on Information Engineering 1995 (SICON/ICIE'95)
Singapore, July 3 – 7, 1995; pages 299 – 303.

Protocol Configuration and Interoperability — A Case Study —

Burkhard Stiller

University of Cambridge
Computer Laboratory
New Museums Site, Pembroke Street
Cambridge, CB2 3QR, England, U.K
Phone: +44 +1223 334476
FAX: +44 +1223 334678
Burkhard.Stiller@cl.cam.ac.uk

Thomas Plagemann

UNIK Center for Technology at Kjeller
University of Oslo
Granaveien 33, P.O. Box 70
N-2007 Kjeller, Norway
Phone: +47 63 81 45 71 214
FAX: +47 63 81 81 46
plageman@unik.no

Abstract

Providing an application-tailored communication protocol increases the exploitation of service specific characteristics by any application. Especially, the configuration approach of communication protocols according to specified and requested application Quality-of-Service (QoS) requirements offers a highly flexible scenario to allow for a sufficient and appropriate protocol processing. The approaches of the Function-based Communication Subsystem (F-CSS) *and the* Dynamic Configuration of Protocols (Da CaPo) *provide a basis for configuring specifically requested services. Important for a future integration into the existing networking environment is the interoperability of configuration-based approaches. The following case study evaluates the possibility of interoperating F-CSS and Da CaPo.*

D.1 Motivation and Introduction

Today, communication subsystems have to provide a wide range of services to an emerging diversity of distributed multimedia applications. These services require a huge amount of performance like high throughput, low delay, and low delay jitter that has to be delivered unaltered from the network up to the applications. Traditional communication subsystems are built on top of networks, such as Token Ring and Ethernet in the local area or X.25 in the wide area. The corresponding protocols are tailored according to low bit rates and relatively high error rates of these networks. However, modern networks, such as FDDI, DQDB, or the ATM-based B-ISDN, offer much higher bit rates and lower error rates. Traditional protocols running on top of these networks include redundancy in terms of unnecessary protocol functions and provide lower performance than expected. Furthermore, most traditional communication subsystems support reliability related Quality-of-Services (QoS) aspects only. Missing aspects include, *e.g.,* synchronization, security, or the negotiation of QoS.

Communication subsystems like *ADAPTIVE* [48], *F-CSS* [19], [20], and *Da CaPo* [47], [137] try to reduce protocol complexity, increase protocol performance, and serve as well various QoS guarantees by introducing a protocol configuration approach. Further approaches for high-speed communication systems are presented and classified in [138].

The main principles of *Protocol Configuration* are decomposition and configuration. Complex protocols are decomposed into fine granular building blocks, each defining a single protocol task. Configuration comprises the task of selecting building blocks according to application requirements and combining them into an executable communication protocol. *ADAPTIVE*, *F-CSS*, and *Da CaPo* define different architectures to fulfil these tasks.

Interoperability between these systems would obviously broaden the acceptance and usage of protocol configuration. Generally, the term interoperability refers to the ability of different protocol implementations to interwork. Systems like *ADAPTIVE*, *F-CSS*, and *Da CaPo* do not implement a particular protocol, instead, they embody systems that implement as many different protocols as could be configured out of the available building blocks. By this, the question of interoperability between protocol configuration approaches is more complex and includes the following:

- Are the systems able to implement, *i.e.,* configure the same protocol ?
- Could the systems agree upon a common protocol configuration ?
- Are these different protocol implementations interoperable ?

It is the purpose of this work to outline the possible interoperation between different configuration approaches. This study is exemplified by using technical details of *F-CSS* and *Da CaPo*. However, a formal proof of interoperability is beyond the scope of this work. The remainder of this paper is structured as follows. Section D.2 introduces the basic concepts of protocol configuration with respect to interoperability issues. Section D.3 discusses whether *F-CSS* and *Da CaPo* are able to implement the same protocol. Section D.4 presents how different systems can negotiate a coincident protocol configuration. Finally, a summary and outlook is given in the concluding Section D.5.

D.2 Configuration and Interoperability

Protocol configuration is regarded as the task of mapping QoS requirements of applications, available network services, and local system services onto certain functionality of protocols, in detail the basic building blocks. These building blocks implement single protocol functions or microprotocols [139] that communicate with their peers by handling messages and adding control information to them. The communication subsystem automatically selects the most appropriate building blocks and defines their dependencies according to QoS requirements and available resources. The resulting protocol description is used to construct an executable communication protocol, which in turn is the final result of the entire protocol configuration.

In order to communicate, communication partners have to use the same protocol and the same protocol configuration, respectively. This includes the building blocks and their interdependencies as well as the Protocol Data Units (PDU) and their structure. Therefore, configuration requires the negotiation of a common and possibly best suited protocol between the communication partners. At least such a configuration negotiation protocol has to inform communication partners on the selected protocol by sending them a corresponding description. The connection establishment terminates unsuccessfully, if the responding peer is not able to construct a corresponding executable protocol. In other words, such an unilateral negotiation represents a “take it or leave it” approach.

In more complex negotiation schemes, communication partners collaborate to find a protocol configuration that is suited to the needs and aims of every involved partner. For simplicity, in this paper the simple unilateral negotiation schema is discussed only, in which the initiating partner demands its peer to use a particular configuration.

In order to achieve interoperability between different configuration approaches, three main aspects are of central importance:

- **Building Blocks and Protocols:**
  Microprotocols are like conventional protocols, they are only more simple, since *one single* specific task is associated with them only. Therefore, the problem of interoperability

between different implementations of a microprotocol (or building blocks in different communication subsystems) is similar to the problem of interoperability between conventional protocols. For configuring interoperable protocols out of interoperable building blocks, it is necessary that the communication peers are able to use the same PDU structures.

- **Configuration Description:**
  Every building block used within the configuration has to be specified uniquely in terms of relevant information (such as an identifier) to allow for the correct identification of similar features on either side of the communicating peers. Additionally, the resulting protocol description has to be formalized to specify the correct interdependencies of selected building blocks. Furthermore, the structure of PDUs has to be specified.
- **Configuration Negotiation Protocol:**
  To negotiate a common protocol configuration correctly, every participating side has to rely on the same negotiation rules. This includes the transfer of well-defined configuration descriptions between communicating peers.

D.3 Building Blocks and Protocols

Different protocol configuration approaches have to use the same notion of building blocks to be able to construct protocols that are interoperable.

In *F-CSS* a two level hierarchy is used to define separate functionality. Protocol functions define at the upper level a specific task, such as acknowledgement, flow control, or checksumming. The lower level defines protocol function mechanisms — shortly, mechanism — that represents the algorithm used to fulfil the task. Examples are a positive-cumulative or selective-negative acknowledgement mechanism for the protocol function acknowledgement. Every of these mechanisms is characterized by certain conditions that allow for the definition of the dependency of the regarded mechanism on different QoS parameters, such as delay, bit and packet error rate, or throughput. These interdependencies and further information are specified in the *Function-based Protocol Configuration Language (F-PCL)*. An executable protocol function is called an instantiation, which is in turn part of the executable communication protocol. These protocols may be executed on a workstation or a Transputer network. Therefore, the building blocks for protocol configuration in *F-CSS* are mapped onto pieces of software running on the underlying hardware platform.

Da CaPo uses also the notion of protocol functions. A protocol function represents typical communication tasks like flow control, presentation coding, or compression. Protocol functions can be implemented by different protocol mechanisms. For example, compression can be performed by applying H.261, JPEG, or MPEG. Protocol mechanisms in turn can be implemented by software or hardware modules. Each module is characterized by certain properties, *e.g.,* different degrees of error correction or detection, which are defined in the language *L*. Each module is a single part of the executable communication protocol in the communication subsystem. The *Da CaPo* architecture is tailored to workstations with one, or a few Central Processing Units (CPU). The building blocks for protocol configuration in *Da CaPo* are software modules executed by a CPU or hardware modules that are plugged into the main workstation bus, like a JPEG compression card.

Obviously, both systems apply the same hierarchy of abstraction comprising protocol functions, mechanisms, and the implementation of these mechanisms. Therefore, mechanisms specify certain abstract protocol rules, supporting data transfer from one service user to its peer and they specify user data handling. In other words, the terms mechanism and microprotocol have the same semantic. Moreover, communicating entities have to use building blocks that implement the same mechanism. Whether these building blocks are implemented in hardware or software, or run on the main CPU or on an external Transputer board is not of importance for interoperability.

A configured protocol relies on a unique definition of a PDU structure. Therefore, *F-CSS* does not limit this structure, since each building block is separately responsible for relevant control data. A subset of building blocks may use a common part of a header or every building block may add a predefined length of information as well. The scheme used is dependent on the current implemented building blocks.

Building blocks that are combined to a protocol in *Da CaPo* do not exchange (or share) any information except the PDUs that are exchanged between the communication peers. During the initialization of a protocol each building block is informed about the location of its control information field(s) in the PDU. By this, *Da CaPo* protocols can adapt to given PDU structures, if the number of control fields and their size satisfies the requirements of all building blocks in the particular protocol configuration. The only restriction is that the header parts of PDU structures have to be static during the lifetime of a connection or communication relation, *i.e.,* variable PDU structures are not supported.

D.4 Configuration Negotiation

Naturally, communicating entities have to use a coincident communication protocol to exchange user data correctly. In the context of protocol configuration such a coincident protocol has to be negotiated between partners. The negotiation in turn has to be performed according to a standardized *Configuration Negotiation Protocol (CNP).* In the simplest negotiation scheme, the initiating partner sends a description of the protocol to be used to its peer(s). In more complex scenarios (such as peer-to-multipeer) an initial description will be distributed as well, but the rules of acceptance include more possibilities than only “accept” or “reject”. Syntax and semantic of this protocol description are defined in either case in a *Protocol Graph Description Exchange Format (PG-DEF).* This format forms a central part of the CNP and requires at least:

- Standards for mechanisms (*i.e.,* specifications of building blocks), including a specification of the necessary control fields for the PDU header structure, and
- a unique and standardized identifier for each mechanism in use.

Furthermore, the standardization of applicable protocol functions (such as acknowledgements, synchronization, or checksumming) and their refined mechanisms (such as negative-cumulative and positive-selective acknowledgements, or inter- and intra-stream synchronization) is necessary to support collaborative configuration and negotiation.

D.4.1 Configuration Description

The basic protocol description comprises the specification of dependencies between every building block (in form of a graph) and the specification of the protocol data unit format to be used. The basic description may be extended with additional information for the instantiation of mechanisms.

The graph of the configured protocol is described in PG-DEF. The Extended Backus-Naur Form specification of PG-DEF in Figure 14 is based on [140]. The direction has to

```
protocol  = "PROTOCOL" "(" direction ")"
             path [path] "END".
direction = "READ" | "WRITE".
path      = "PATH" ("MAIN" | "ALTERNATE")
             instance {";" instance}.
instance  = "NAME" instname "FUNCTION" modname
            ":" instlist ";" instsucc.
instsucc  = instname {, instname}.
instname  = letter {letter}.
modname   = letter {letter}.
instlist  = "EMPTY" | (instance {"," instname}).
```

Figure 14: EBNF Specification for PG-DEFs

```
headerdesc   = header [header].
header       = "HEADER" ("MAIN" | "ALTERNATE")
                hentry {"," hentry} "END;".
hentry       = srcaddr | destaddr | hfield |
               "(" sharedfields ")".
srcaddr      = "SOURCE" ["POS" number]
               "LEN" number.
destaddr     = "DESTINATION" ["POS" number]
               "LEN" number.
hfield       = hident ["POS" number]
               "LEN" number.
hident       = instname ["." fieldident].
instname     = letter {letter | digit}.
fieldident   = letter {letter | digit}.
sharedfields = hfield {"," hfield}.
number       = digit {digit} | digit
               {hexdigit} "H".
```

Figure 15: EBNF Specification for PDU-DEFs

be specified because a configured protocol supports in *Da CaPo* and *F-CSS* unidirectional user data transfer only (and bidirectional control data transfer).[12] A pre-defined list of valid identifiers for selected building blocks (instance names) within the configured protocol is added. A relevant order of processing these building blocks correctly is defined by the successor attributes for every configured building block.

As mentioned earlier, for each protocol configuration a valid PDU structure has to be fixed. Therefore, the selected PDU structure is described in the *PDU Description Exchange Format (PDU-DEF)*, which is exchanged between communicating peers. Figure 15 depicts the PDU-DEF specification in EBNF and indicates valid PDU header structures [140]. An automatic generation of the header structure is possible.

- Each protocol mechanism specification includes the number of bytes it needs for control information and the direction of its transfer.
- The graph will be traversed at both end-systems in such a way that both peers generate the same linear list of building block names using unique identifiers.

[12] Bidirectional protocols have to be combined out of basic unidirectional protocols.

```
protdesc = "PG" pgdef ";" "PDU" pdudef1 ","
            pdudef2 {";" "ID" para}.
para     = paraid value {"," paraid value}.
paraid   = text.
value    = number | text.
text     = letter {letter | digit}.
```

Figure 16: Specification for Configuration Descriptions

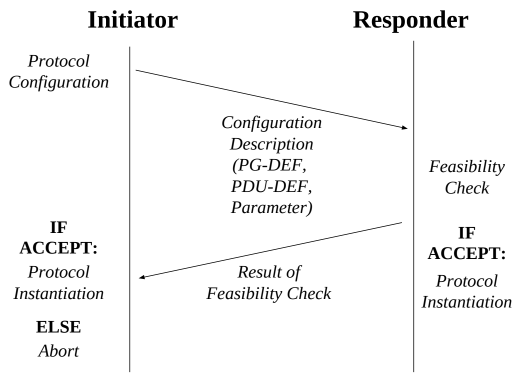

Figure 17: A Simple Unilateral Negotiation

- By substituting the building block name with control info fields and their length of this building block, the header structure for both directions can be generated.
- As there are only fixed header structures supported for protocol performance reasons, this approach is sufficient.

Additionally, a building block may encounter certain parameters that are of vital interest. Examples encompass a *window size* for a flow control, a *message transfer unit size* for segmentation, or a *sending rate* for the data transmission function. Initial values for these parameters have to be set-up accordingly for every peer. Therefore, a simple distribution of these initial values or a more complex negotiation is supported. Handling these parameters correctly, a similar approach as for the instance names is required, since all relevant parameters have to be specified by a unique identifier.

As depicted in Figure 16 for unidirectional protocols, a configuration description comprises a PG-DEF, two PDU-DEFs (one for payload transfer including control and the other one for backward control), and optional parameter identifiers including initial values.

D.4.2 Configuration Negotiation Protocol

The *Configuration Negotiation Protocol* deals with the exchange and negotiation of the protocol description and parameter values as well. Figure 17 outlines a simple unilateral negotiation of a communication protocol and its parameters for two peers. Additionally, the protocol instantiation events are included.

The initiator selects a most appropriate protocol configuration and transmits the configuration description to the responder. This configuration description comprises the PG-DEF, two PDU-DEFs, and an optional number of values for standardized parameters. The responder checks, whether it is possible to establish the corresponding executable protocol (*i.e.,* all building blocks mentioned in the configuration description are available) and returns the result of this feasibility check to the initiator.

D.5 Conclusions and Outlook

Further aspects of a configuring approach for communication protocols encompass the increase of standardization efforts for generic application interfaces —including detailed QoS parameters—, a number of basic protocol functions and mechanisms, and the formal language for defining generally acceptable configuration results.

Generic application service interfaces will allow for a very detailed specification of application requirements. Furthermore, these information can be additionally used by resource reservation protocols or by the underlying operating system for providing service guarantees.

Additionally, any building block for basic communications has to be specified separately, including its functionality, its unique instance name identifier, and relevant parameters. This approach looks quite similar as already pursued in the application layer of the ISO/OSI (International Organization for Standardization/Open Systems Interconnection) protocol stack, while standardizing special application service elements. The proposed definition of a *Protocol Graph Description Exchange Format* solves the problem for specifying configuration results that have to be agreed upon and transmitted to any communication partner.

Acknowledgements: We would like to thank Martina Zitterbart, Andreas Gotti, and Martin Vogt for discussion of these topics.

E Abbreviations

A-QoS	Application-oriented QoS
AAL	ATM Adaptation Layer
ADAPTIVE	A Dynamically Assembled Protocol Transformation, Integration and Validation Environment
ANSA	Advanced Network System Architecture
ASV	Application Service
ATM	Asynchronous Transfer Mode
B-ISDN	Broadband-Integrated Services Digital Network
B-ISUP	Broadband-ISDN User Part
BVSC	Broadband Virtual Signaling Channel
C-QoS	Communication-specific QoS
CEC	Commission of the European Communities
CINEMA	Configurable Integrated Multimedia Architecture
CLNP	Connectionless Network Protocol
CLP	Cell Loss Priority
CM	Connection Management
CMAP	Connection Access Management Protocol
CNP	Configuration Negotiation Protocol
CODCOM	Code Configuration Manager
CogPiT	Configuration of Protocols in TIP
CPU	Central Processing Unit
CRV	Call Reference Value
CSV	Communication Service
Da CaPo	Dynamic Configuration of Protocols
DAN	Desk Area Network
DCE	Data Circuit-Terminating Equipment
DEF	Description Exchange Format
DQDB	Distributed Queue Dual Bus
DSS 1	Digital Subscriber Signaling System No. 1
DSS 2	Digital Subscriber Signaling System No. 2
DTE	Data Terminal Equipment
EBNF	Extended Backus-Naur Form
ETSI	European Telecommunications Standards Institute
F	Flag
F-CSS	Function-based Communication Subsystem
F-PCL	Function-based Protocol Configuration Language
FDDI	Fiber Distributed Data Interface
GFC	Global Flow Control
GMF	Generic Mapping Function
HEC	Header Error Check
HeiTS	Heidelberger Transport System
ILMI	Interim Local Management Interface
IP	Internet Protocol
ISO	International Organization for Standardization
ISUP	ISDN User Part
ITU-T	International Telecommunications Union — Telecommunications Standardization Sector
JPEG	Joint Pictures Expert Group
LAN	Local Area Network
LCRV	Length of Call Reference Value

M	Mandatory
MAN	Metropolitan Area Network
ML	Message Length
MPEG	Motion Pictures Expert Group
ms	milli seconds
MSNA	MultiService Network Architecture
MSNL	MultiService Network Layer
MT	Message Type
MTP	Message Transfer Part
MVSC	Meta Virtual Signaling Channel
N-ISDN	Narrowband-Integrated Services Digital Network
NP	Network Performance
N-QoS	Network-dependent QoS
NNI	Network-Node Interface
O	Optional
OSI	Open Systems Interconnection
PD	Protocol Discriminator
PDU	Protocol Data Unit
PG	Protocol Graph
PROCOM	Protocol Configuration Manager
PT	Payload Type
PVSC	Point-to-Point Virtual Signaling Channel
QoS	Quality-of-Service
QoS-A	Quality-of-Service Architecture
s	seconds
SAAL	Signalling ATM Adaptation Layer
SPANS	Simple Protocol for ATM Network Signaling
SSN 7	Signaling System No. 7
SYSREM	System Resource Manager
TCP	Transmission Control Protocol
TINA	Telecommunications Information Networking Architecture
TP4	Transport Protocol Class 4
TQoS	Transport-oriented QoS
TCP	Transmission Control Protocol
TIP	Transport and Internetworking Package
ToS	Type-of-Service
TSV	Transport Service
U.K.	United Kingdom
UNI	User-Network Interface
VC	Virtual Channel
VCC	Virtual Channel Connection
VCI	Virtual Channel Identifier
VCL	Virtual Channel Link
VMTP	Versatile Message Transaction Protocol
VP	Virtual Path
VPC	Virtual Path Connection
VPI	Virtual Path Identifier
VPL	Virtual Path Link
VSC	Virtual Signaling Channel
WAN	Wide Area Network
XTP	Xpress Transfer Protocol

References

[1] B. Stiller, "A Survey of UNI Signaling Systems and Protocols for ATM Networks," *ACM Computer Communications Review*, vol. 25, pp. 21–33, April 1995.

[2] B. Stiller, *Quality-of-Service — Dienstgüte in Hochleistungsnetzen*, vol. TAT Nr. 21. Bonn, Germany: International Thomson Publishing GmbH, August 1995.

[3] ISO Standard, IS 7498, *Information processing systems — Open Systems Interconnection — Basic Reference Model*, 1985.

[4] International Organization for Standardization, "Quality-of-Service — Basic Framework — CD Text," Tech. Rep. ISO/IEC JTC1/SC21 N9309, ISO, 9. – 13. January 1995.

[5] ITU-T Recommendation E.800, *Quality of service and dependability vocabulary.* Geneva, Switzerland, 22. November 1994.

[6] ITU-T Recommendation I.356, *B-ISDN ATM Layer Cell Transfer Performance.* Geneva, Switzerland, 23. December 1994.

[7] F. Ross, "FDDI — A Tutorial," *IEEE Communications Magazine*, vol. 24, pp. 10–17, May 1986.

[8] ISO Standard IS 8073, *Information processing systems — Open Systems Interconnection — Transport Protocol Definition*, 1988.

[9] ISO Standard, IS 8473, *Information processing systems — Open Systems Interconnection — Connectionless Network Protocol Definition*, 1988.

[10] W. T. Strayer, B. J. Dempsey, and A. C. Weaver, *XTP: The Xpress Transfer Protocol.* Reading, Massachusetts, U.S.A.: Addison Wesley, 1992.

[11] R. Händel and M. Huber and S. Schröder, *ATM Networks: Concepts, Protocols, Applications.* Reading, Massachusetts, U.S.A.: Addison-Wesley, 1994.

[12] ITU-T Recommendation I.211, *B-ISDN service aspects.* Geneva, Switzerland, 30. March 1994.

[13] ITU-T Recommendation I.350, *General aspects of quality of service and network performance in digital networks, including ISDN.* Geneva, Switzerland, 6. April 1994.

[14] ITU-T Recommendation I.371, *Traffic Control and Congestion Control in B-ISDN.* Geneva, Switzerland, 30. March 1994.

[15] A. Danthine, *The OSI'95 Transport Service with Multimedia Support — Research Reports ESPRIT, Project 5341, Volume 1.* Berlin, Germany: Springer, 1994.

[16] A. Campbell, G. Coulson, and D. Hutchison, "A Quality-of-Service Architecture," *ACM Computer Communications Review*, vol. 24, pp. 6–27, April 1994.

[17] A. Campbell, G. Coulson, F. García, and D. Hutchison, "Orchestration Services for Distributed Multimedia Synchronization," in *High Performance Networking, IV*, (Amsterdam, Netherlands), pp. 153–168, IFIP Transactions C-14, North Holland, 1993.

[18] A. Campbell, G. Coulson, F. García, and D. Hutchison, "Integrated Quality of Services for Multimedia Communications," in *IEEE INFOCOM*, (San Francisco, California, U.S.A.), pp. 732–739, IEEE, 28. March – 1. April 1993.

[19] M. Zitterbart, B. Stiller, and A. Tantawy, "A Model for High-Performance Communication Subsystems," *IEEE Journal on Selected Areas in Communication*, vol. 11, pp. 507–518, May 1993.

[20] B. Stiller, *Flexible Protokollkonfiguration zur Unterstützung eines diensteintegrierenden Kommunikationssubsystems*, vol. 10, no. 306, (Fortschrittberichte). Düsseldorf, Germany: VDI, 16. February 1994.

[21] K. Nahrstedt and J. Smith, "Design, Implementation and Experiences of the OMEGA End-Point Architecture," Tech. Rep. TR-95-22, Distributed Systems Laboratory, University of Pennsylvania, Philadelphia, Pennsylvania, U.S.A., 1995.

[22] K. Nahrstedt and R. Steinmetz, "The QoS Broker," *IEEE Multimedia*, vol. 2, pp. 53–67, Spring 1995.

[23] K. Rothermel, I. Barth, and T. Helbig, "CINEMA — An Architecture for Distributed Multimedia Applications," in *Architecture and Protocols for High Speed Networks*, (Amsterdam, Netherlands), pp. 253–271, Kluwer Academic Publishers, 1994.

[24] C. Schmidt and M. Zitterbart, "QoS Management for Service Integrated Communication Systems," in *7th IEEE Workshop on Local and Metropolitan Area Networks*, (Marathon, Florida, U.S.A.), pp. Session 6, Paper 4, 26.–29. March 1995.

[25] C. Vogt, R. Herrtwich, and R. Nagarajan, "HeiRAT: The Heidelberg Resource and Administration Technique, Design Philosophy and Goals," in *Kommunikation in Verteilten Systemen, KiVS'93*, (München, Germany), pp. 501–515, 3. – 5. March 1993.

[26] A. Lazar, L. Ngoh, and A. Sahai, "Multimedia Networking Abstractions with Quality of Service Guarantees," in *SPIE Conference on Multimedia Computing and Networking*, (San Jose, California, U.S.A.), pp. 140–154, 6. – 8. February 1995.

[27] G. Nilsson, F. Dupny, and M. Chapman, "An Overview of the Telecommunications Information Networking Architecture," in *Conference on TINA*, (Melbourne, Australia), pp. 1–12, 1995.

[28] G. Li and D. Otway, *An Open Architecture for Real-time Processing.* APM Ltd., Cambridge, U.K., APM.1270.00.01, 31. July 1994.

[29] J. Jung and D. Seret, "Translation of QoS-Parameters into ATM Performance Parameters in B-ISDN," in *IEEE INFOCOM 1993*, (San Franciso, California, U.S.A.), pp. 748–755, 28. March – 1. April 1993.

[30] M. Hofmann and C. Schmidt and H. Wiltfang, *Mapping the MMT FlowSpec on ATM Quality-of-Service Parameters.* Institut für Telematik, Universität Karlsruhe, Germany, November 1994.

[31] R. Gopalakrishna and G. Parulkar, "Efficient Quality of Service Support in Multimedia Computer Operating Systems," Tech. Rep. WUCS-94-26, Department of Computer Science, Washington University, St. Louis, Missouri, U.S.A., 3. November 1994.

[32] J. Hyman, A. Lazar, and G. Pacifici, "Real-Time Scheduling with Quality-of-Service Constraints," *IEEE Journal on Selected Areas in Communication*, vol. 9, pp. 1052–1063, September 1991.

[33] B. Stiller, "Hierarchical Mapping of Enhanced QoS Parameters Based on OSI Protocols," in *3rd IEEE Workshop on the Architecture and Implementation of High Performance Communication Subsystems*, pp. Session 2, Paper 1, 23. – 25. August 1995.

[34] P. Robin, G. Coulson, A. Campbell, G. Blair, M. Papathomas, and D. Hutchison, "Implementing a QoS Controlled ATM Based Communications Systems in Chorus," in *4th International IFIP Workshop on Protocols for High Speed Networks*, (Vancouver, Canada), pp. 19–35, 10. - 12. August 1994.

[35] C. Topolcic, *Experimental Internet Stream Protocol, Version 2 (ST-II), RFC 1190*, October 1990.

[36] L. Zhang, S. Deering, D. Estrin, S. Shenker, and D. Zappala, "RSVP: A New Resource Reservation Protocol," *IEEE Network Magazine*, vol. 7, pp. 8–18, September 1993.

[37] C. Lowery, "Protocols for Providing Performance Guarantees in a Packet-Switching Intenet," Tech. Rep. TR-91-002, International Computer Science Institute, Berkeley, California, U.S.A., January 1991.

[38] D. Anderson, R. Herrtwich, and C. Schaefer, "SRP: A Resource Reservation Protocol for Guaranteed-Performance Communication in the Internet," Tech. Rep. TR-90-006, International Computer Science Institute, Berkeley, California, U.S.A., February 1990.

[39] L. Zhang, "Virtual Clock: A New Traffic Control Algorithm for Packet Switched Networks," *ACM Computer Communications Review (SIGCOMM)*, vol. 20, pp. 19–29, 24. – 27. September 1990.

[40] T. Burkow, "Operating System Support for Distributed Multimedia Applications; A Survey of Current Research," Tech. Rep. Memoranda Informatica 94–57, University of Twente, Faculty of Computer Science, Twente, Netherlands, June 1994.

[41] E. A. Hyden, "Operating System Support for Quality-of-Service," Tech. Rep. 94-340, University of Cambridge, Computer Laboratory, Cambridge, England, U.K., June 1994.

[42] I. Leslie, D. McAuley, and S. Mullender, "Pegasus – Operating Support for Distributed Multimedia Systems," Tech. Rep. TR 282/Pegagus-92-2, University of Cambridge, Computer Laboratory, Cambridge, England, U.K., December 1992.

[43] T. Roscoe, *The Structure of a Multi-Service Operating System.* Cambridge, England, U.K., April 1995.

[44] B. Stiller, "Aushandlungsverfahren für QoS-Parameter und (Re-) Konfigurationsbeschreibungen," in *Anwendungsunterstützung für heterogene Rechnernetze*, (Freiberg/Sachsen, Germany), pp. 79–88, 30. – 31. March 1995.

[45] H. Eriksson, "MBone: The Multicast Backbone," *Communications of the ACM*, vol. 37, pp. 54–60, Aug. 1994.

[46] T. LaPorta and M. Schwartz, "Architectures, Features, and Implementation of High-Speed Transport Protocols," *IEEE Network Magazine*, vol. 5, pp. 14–22, May 1991.

[47] T. Plagemann, B. Plattner, M. Vogt, and T. Walter, "A Model for Dynamic Configuration of Light-Weight Protocols," in *IEEE 3rd Workshop on Future Trends of Distributed Systems*, (Taipeh, Taiwan), pp. 100–106, 14. – 16. April 1992.

[48] D. F. Box, D. C. Schmidt, and T. Suda, "ADAPTIVE: An Object-Oriented Framework for Flexible and Adaptive Communication Protocols," in *High Performance Networking, IV*, (Amsterdam, Netherlands), pp. 367–382, IFIP Transactions C-14, North Holland, 1993.

[49] M. Zitterbart, "High-Speed Transport Components," *IEEE Network Magazine*, vol. 5, pp. 54–63, January 1991.

[50] B. Stiller, "A Framework for QoS Updates in a Networking Environment," Tech. Rep. TR 374, University of Cambridge, Computer Laboratory, Cambridge, England, U.K., July 1995.

[51] B. Stiller, "PROCOM: A Protocol Configuration Manager in the Function-based Communication Subsystem," in *First International Workshop on High Performance Protocol Architectures*, (Sophia-Antipolis, France), pp. Session 3, Paper 9, 15.-16. December 1994.

[52] B. Stiller, "CogPiT — Configuration of Protocols in TIP," Tech. Rep. TR 368, University of Cambridge, Computer Laboratory, Cambridge, England, U.K., June 1995.

[53] B. Stiller and T. Plagemann, "Protocol Configuration and Interoperability — A Case Study —," in *IEEE Singapore INnernational Conference on Networks (SICON)*, p. ???, 3. – 7. July 1995.

[54] ITU-T Recommendation I.121, *Broadband aspects of ISDN (B-ISDN Protocol Reference Model).* Geneva, Switzerland, 16. June 1993.

[55] ITU-T Recommendation I.150, *B-ISDN asynchronous transfer mode functional characteristics.* Geneva, Switzerland, 28. March 1994.

[56] ITU-T Recommendation Q.931, *Digital subscriber signalling system no. 1 (DSS 1) — ISDN user-network interface layer 3 specification for basic call control.* Geneva, Switzerland, 22. November 1994.

[57] ITU-T Draft Recommendation Q.2931, *Edinburgh TD 155, Broadband Integrated Services Digital Network (B-ISDN), Digital Subscriber Signalling System No.2, User Network Interface Layer 3 Specification for Basic Call/Connection Control.* Geneva, Switzerland, 13. – 21. June 1994.

[58] ITU-T Draft Recommendation Q.298x, *Draft Text for Q.298x, Edinburgh TD 152 Rev 1 & 147, Broadband Integrated Services Digital Network (B-ISDN), Digital Subscriber Signalling System No. 2, User Network Interface Layer 3 Specification for Point-Point Multiconnection Call Control.* Geneva, Switzerland, 13.-21. June 1994.

[59] ITU-T Draft Recommendation Q.2961, *Draft Text for Q.2961, Negotiation/Renegotiation: Traffic and QoS Parameters.* Geneva, Switzerland, 1994.

[60] ITU-T Draft Recommendation Q.2962, *Draft Text for Q.2962, Negotiation/Renegotiation: Traffic and QoS Negotiation during Call Establishment.* Geneva, Switzerland, 1994.

[61] ITU-T Draft Recommendation Q.2963, *Draft Text for Q.2963, Negotiation/Renegotiation: Traffic and QoS Negotiation during Active Phase.* Geneva, Switzerland, 1994.

[62] ATM-Forum, *ATM User Network Interface Specification, Version 3.0.* Englewood Cliffs, New Jersey, U.S.A.: Prentice Hall, 1993.

[63] ATM-Forum, *ATM User Network Interface Specification, Version 3.1*, 21. July 1994.

[64] Fore Systems, Inc, "SPANS: Simple Protocol for ATM Network Signalling, Version 2.1.1.," tech. rep., Fore Systems, Inc, Pittsburgh, Pennsylvania, U.S.A., 1992.

[65] Fore Systems, Inc, "ATM Networks," tech. rep., Fore Systems, Inc, Pittsburgh, Pennsylvania, U.S.A., September 1993.

[66] R. Bubenick, J. DeHart, and M. Gaddis, "Multipoint Connection Management in High Speed Networks," in *IEEE INFOCOM 1991*, (Bal Harbor, Florida, U.S.A.), pp. 59–68, 9.-11. April 1991.

[67] K. Cox and J. DeHart, "Connection Management Access Protocol (CMAP) Specification, Version 3.0," Tech. Rep. WUCS-94-21, Department of Computer Science, Washington University, St. Louis, Missouri, U.S.A., 10. November 1994.

[68] S. Crosby, "MSNL Connection Management," Tech. Rep. ATM Document Collection 3 (Blue Book), University of Cambridge, Computer Laboratory, Cambridge, England, U.K., 16. February 1993.

[69] S. Crosby, "MSNL UNI Signalling Protocol, DRAFT Specification," Tech. Rep. Revision 1.5, University of Cambridge, Computer Laboratory, Cambridge, England, U.K., 1. December 1994.

[70] ITU-T Recommendation I.361, *B-ISDN ATM layer specification.* Geneva, Switzerland, 22. April 1994.

[71] DARPA, *Internet Protocol — DARPA Internet Protocol Specification, RFC 793*, September 1981.

[72] ITU-T Recommendation E.164, *Numbering Plan for ISDN Area.* Geneva, Switzerland, 25. March 1994.

[73] ISO Standard, IS 8473 Addendum 4, *Information processing systems — Open Systems Interconnection — Connectionless Network Protocol Definition Addendum 4, Network Addressing*, 1988.

[74] P. Newman, "ATM Local Area Networks," *IEEE Communications Magazine*, vol. 32, pp. 86–98, March 1994.

[75] International Organization for Standardization, "Telecommunications and Information Exchange between Systems — Second Draft on Multipeer Taxonomy," Tech. Rep. ISO/IEC JTC1/SC6 N8693, ISO, 1. July 1994.

[76] ITU-T Recommendation I.363, *B-ISDN ATM Adaptation Layer (AAL) specification.* Geneva, Switzerland, 4. August 1994.

[77] ITU-T Draft Recommendation Q.2100, *B-ISDN signalling ATM adaptation layer (SAAL) overview description.* Geneva, Switzerland, 29. July 1994.

[78] ITU-T Draft Recommendation Q.2110, *B-ISDN ATM adaptation layer — Service specific connection oriented protocol (SSCOP).* Geneva, Switzerland, 29. July 1994.

[79] ITU-T Draft Recommendation Q.2120, *B-ISDN meta-signaling protocol connection oriented protocol (SSCOP).* Geneva, Switzerland, 17. December 1993.

[80] ITU-T Draft Recommendation Q.2130, *B-ISDN signalling ATM Adaptation Layer — Service specific coordination function for support of signalling at the user-network interface (SSCF at UNI).* Geneva, Switzerland, 29. July 1994.

[81] DARPA, *Transmission Control Protocol — DARPA Internet Protocol Specification, RFC 791,* September 1981.

[82] Netcomm, Ltd., "DV 2 — Switch Installation Managers Guide, Version 2.0," tech. rep., Netcomm, Ltd., Basildon, Essex, U.K., 1993.

[83] ITU-T Draft Recommendation Q.2650, *B-ISDN interworking between SS No. 7 broadband ISDN user part (B-ISUP) and digital subscriber SS No. 2 (DSS 2).* Geneva, Switzerland, 17. December 1993.

[84] ITU-T Draft Recommendation Q.2660, *Broadband integrated services digital network (B-ISDN) – Interworking between SS No. 7 broadband ISDN part (B-ISUP).* Geneva, Switzerland, 17. December 1993.

[85] ITU-T Draft Recommendation Q.2661, *Broadband integrated services digital network (B-ISDN) functional description of B-ISDN user part (B-ISUP) SS No. 7.* Geneva, Switzerland, 17. December 1993.

[86] ITU-T Draft Recommendation Q.2662, *Broadband integrated services digital network (B-ISDN) — General functions messages, signals of B-ISDN user part (B-ISUP).* Geneva, Switzerland, 17. December 1993.

[87] ITU-T Draft Recommendation Q.2663, *Broadband integrated services digital network (B-ISDN) — SS No. 7 B-ISUP user part B-ISUP — Formats and codes.* Geneva, Switzerland, 17. December 1993.

[88] ITU-T Draft Recommendation Q.2664, *Broadband integrated service digital network (B-ISDN) SS No. 7 B-ISDN user part (B-ISUP) Basic call procedures.* Geneva, Switzerland, 17. December 1993.

[89] ITU-T Recommendation Q.761, *Functional description of the ISDN user part of signalling system no. 7.* Geneva, Switzerland, 27. April 1994.

[90] ITU-T Recommendation Q.762, *General function of messages and signals of the ISDN user part of signalling system no. 7.* Geneva, Switzerland, 27. April 1994.

[91] ITU-T Recommendation Q.763, *Formats and codes of the ISDN user part signalling procedures.* Geneva, Switzerland, 27. April 1994.

[92] ITU-T Recommendation Q.764, *Signalling System No. 7 — ISDN user part signalling procedures.* Geneva, Switzerland, 1. Juni 1994.

[93] ITU-T Recommendation Q.700, *Introduction to CCITT Signalling System No. 7.* Geneva, Switzerland, 1. June 1994.

[94] ITU-T Recommendation Q.701, *Functional description of the message transfer part (MTP) of signalling system no. 7.* Geneva, Switzerland, 27. April 1994.

[95] ITU-T Draft Recommendation Q.1400, *Architecture framework for the development of signalling and OAM protocols using OSI concepts.* Geneva, Switzerland, July 1994.

[96] G. Siegmund, *ATM — Die Technik des Breitband-ISDN.* Heidelberg, Germany: R. v. Decker's Verlag, G. Schenk, 2nd ed., 1994.

[97] CCITT Recommendation X.25, *Interface between data terminal equipment (DTE) and data circuit-terminating equipment (DCE) for terminal operating in the packet mode and to public data networks by dedicated circuit.* Geneva, Switzerland, 1989.

[98] ITU-T Recommendation Q.930, *Digital subscriber signalling system no. 1 (DSS 1) — ISDN user-network interface layer 3 — General aspects.* Geneva, Switzerland, 12. August 1994.

[99] ITU-T Draft Recommendation Q.93B, *B-ISDN User-Network Interface Layer 3 Specification for Basic Call/Bearer Control.* Geneva, Switzerland, October 1992.

[100] ETSI Standard, DE/SPS 5024, *Signalling: Basic Call*, 1992.

[101] ETSI Standard, DE/SPS 5034, *Signalling: Supervisory Call*, 1992.

[102] ATM-Forum 94-0998, *Signalling Subgroup Meeting Notes, UNI 4.0 Features*, 26. – 29. September 1994.

[103] L. Roberts, T. MacDonald, K. Duffie, and G. Bernstein, "Fast Select Virtual Circuit Routing for B-ISDN Network ," tech. rep., NetExpress, Foster City, California, U.S.A., 1992.

[104] S. E. Minzer, "A Signalling Protocol for Complex Multimedia Services," *IEEE Journal on Selected Areas in Communications*, vol. 9, pp. 1383–1394, December 1991.

[105] P. A. Miller and P. N. Turcu, "Generic Signalling Protocol: Architecture, Model, and Services," *IEEE Transactions on Communications*, vol. 40, pp. 957–966, May 1992.

[106] P. A. Miller and P. N. Turcu, "Generic Signalling Protocol: Switching Networking, and Internetworking," *IEEE Transactions on Communications*, vol. 40, pp. 967–979, May 1992.

[107] J. DeHart, M. Gaddis, and R. Bubenick, "Connection Management Access Protocol (CMAP) Specification, Version 2.1.1," Tech. Rep. WUCS-92-01, Department of Computer Science, Washington University, St. Louis, Missouri, U.S.A., 7. May 1992.

[108] G. Armitage, "gNET: An ATM LAN Signalling Protocol," Tech. Rep. ISBN 0-85867-077-1, University of Melbourne, Department of Electrical Engineering, Grattan Street, Parkville 3052, Melbourne, Australia, February 1994.

[109] S. E. Minzer and D. R. Spears, "New Directions in Signalling for Broadband-ISDN," *IEEE Communications Magazine*, vol. 16, pp. 6–14, February 1989.

[110] G. E. Daddis and H. C. Tong, "A Taxonomy of Broadband Integrated Switching Architectures," *IEEE Communications Magazine*, vol. 16, pp. 32–42, February 1989.

[111] W. M. Harman and C. F. Newman, "ISDN Protocols for Connection Control," *IEEE Journal on Selected Areas in Communications*, vol. 7, pp. 1034–1042, September 1989.

[112] W. Stallings, *ISDN and Broadband ISDN.* New York, New York, U.S.A.: MacMillan, 2nd ed., 1992.

[113] G. I. Stassinopoulus and I. S. Venieris, "ATM Adaptation Layer Protocols for Signalling," *Computer Networks and ISDN Systems*, vol. 23, pp. 287–304, 1992.

[114] A. Tschirner, "Entwurf eines Multicast-Protokolls für ATM-Netze," Master's thesis, Diplomarbeit at Universität Karlsruhe, Institut für Telematik, Germany, 30. November 1993.

[115] T. LaPorta, M. Veeraraghavan, E. Ayanoglu, M. Karol, and R. D. Gitlin, "B-ISDN: A Technology Discontinuity," *IEEE Communications Magazine*, vol. 32, pp. 84–97, October 1994.

[116] R. O. Onvural, *Asynchronous Transfer Mode Networks: Performance Issues.* Boston, Massachusetts, U.S.A.: Artech House, 1994.

[117] ISO Standard, IS 8072, *Information processing systems — Open Systems Interconnection — Transport Service Definition*, 1986.

[118] ISO Standard IS 8326, *Information processing systems — Open Systems Interconnection — Basic Connection Oriented Session Service Definition*, 1987.

[119] ISO Standard IS 8822, *Information processing systems — Open Systems Interconnection — Connection Oriented Presentation Service Definition*, 15. August 1987.

[120] ITU-T Recommendation I.351, *Relationship among ISDN performance Recommendations.* Geneva, Switzerland, 30. March 1994.

[121] G. Solvie, *Adaptive Kommunikationsunterstützung in offenen Systemen*, vol. 10, no. 301. Düsseldorf, Germany: VDI, 22. December 1993.

[122] L. Svobodova, "Measured Performance of Transport Service in LANs," in *Computer Networks and ISDN Systems*, vol. 18, pp. 31–45, 1989.

[123] N. C. Hutchinson and L. L. Peterson, "The *x*-kernel: An Architecture for Implementing Network Protocols," *IEEE Transactions on Software Engineering*, vol. 17, pp. 64–76, January 1991.

[124] Defense Advanced Research Project Agency, *Transmission Control Protocol*, September 1981.

[125] D. D. Clark and D. L. Tennenhouse, "Architectural Considerations for a New Generation of Protocols," *Computer Communication Review (SIGCOMM)*, pp. 200–208, 24.-27. September 1990.

[126] A. Danthine, "ESPRIT Project OSI 95 — New Transport Service for High-Speed Networking," in *Computer Networks and ISDN Systems*, vol. 25, pp. 384–399, 1992.

[127] R. W. Watson and S. A. Mamrak, "Gaining Efficiency in Transport Services by Appropriate Design and Implementation Choices," *ACM Transactions on Computer Systems*, vol. 5, pp. 97–120, May 1987.

[128] A. A. R. Cockburn, "Efficient Implementation of the OSI-Transport Protocol Checksum Algorithm Using 8/16 bit Arithmetic," tech. rep., IBM Research Division, Zürich Research Laboratories, Rüschlikon, Switzerland, 13. April 1987.

[129] T. Braun, *Ein paralleles Transportsubsystem für zellenbasierte Hochgeschwindigkeitsnetze*, vol. 10, no. 254. Düsseldorf, Germany: VDI, 30. June 1993.

[130] T. Braun, "PATROCLOS: A Flexible and High-Performance Transport Subsystem," in *4th International IFIP Workshop on Protocols for High Speed Networks*, (Vancouver, Canada), 10.-12. August 1994.

[131] T. Braun, B. Stiller, and M. Zitterbart, "XTP and VMTP on Multiprocessor Architectures," in *2nd Joint Workshop on High Speed Networks*, (Stuttgart, Germany), pp. 41–43, 19.-20. September 1992.

[132] M. A. Sidenius, "Hardware Support for Implementation of Transport Protocols," in *Protocols for High Speed Networks, II*, (Amsterdam, Netherlands), pp. 251–267, North Holland, 1991.

[133] D. R. Cheriton, "VMTP: A Transport Protocol for the Next Generation of Communication Systems," *Computer Communication Review (SIGCOMM)*, pp. 406–415, 5.-7. August 1986.

[134] C. Schmidt, B. Stiller, and M. Zitterbart, "Towards Flexible Service-integrated Communication Subsystems," in *IASTED/ISMM International Conference on Distributed Multimedia Systems and Applications*, (Honolulu, Hawaii, U.S.A.), pp. 361–367, 15.-17. August 1994.

[135] B. Stiller, "CODCOM: A Code Configuration Manger in the Function-based Communication Subsystem," in *4th Open Workshop on High Speed Networks*, (Brest, France), 7.-9. September 1994.

[136] Glockenspiel, Documentation Unit, Dublin, Ireland, *Glockenspiel C++ Version 2*, 1990.

[137] T. Plagemann, J. Waclawczyk, and B. Plattner, "Management of Configurable Protocols for Multimedia Application," in *ISMM International Conference on Distributed Multimedia Systems and Applications*, (Honolulu, Hawaii, U.S.A.), pp. 78–81, 15. – 17. August 1994.

[138] D. Feldmeier, "A Framework of Architectural Concepts for High-Speed Communication Systems," *IEEE Journal on Selected Areas in Communication*, vol. 11, pp. 480–488, May 1993.

[139] S. W. O'Malley and L. L. Peterson, "A Dynamic Network Architecture," *ACM Transactions on Computer Systems*, vol. 10, pp. 110–143, May 1992.

[140] A. Gotti, *The Da CaPo System.* Tech. Rep. Laboratory of Computer Engineering and Networks, Swiss Federal Institute of Technology, Zürich, Switzerland, 1994.