Handling Realtime Traffic in Mobile Networks

Subir Kumar Biswas

Darwin College
University of Cambridge

A dissertation submitted for the degree of
Doctor of Philosophy

August 1994

Abstract

The rapidly advancing technology of cellular communication and wireless LAN makes ubiquitous computing feasible where the mobile users can have access to the location independent information and the computing resources. Multimedia networking is another emerging technological trend of the 1990s and there is an increasing demand for supporting continuous media traffic in wireless personal communication environment. In order to guarantee the strict performance requirements of realtime traffic, the connection-oriented approaches are proving to be more efficient compared to the conventional datagram based networking. This dissertation deals with a network architecture and its design issues for implementing the connection-oriented services in a mobile radio environment.

The wired backbone of the proposed wireless LAN comprises of high speed ATM switching elements, connected in a modular fashion, where the new switches and the user devices can be dynamically added and reconnected for maintaining a desired topology. A dynamic reconfiguration protocol, which can cope with these changing network topologies, is proposed for the present network architecture. The details about a prototype implementation of the protocol and a simulation model for its performance evaluation are presented.

CSMA/AED, a single frequency and carrier sensing based protocol is proposed for the radio medium access operations. A simulation model is developed in order to investigate the feasibility of this statistical and reliable access scheme for the proposed radio network architecture. The effectiveness of a per-connection window based flow control mechanism, for the proposed radio LAN, is also investigated. A hybrid technique is used, where the medium access and the radio data-link layers are modelled using the mentioned simulator; an upper layer end-to-end queueing model, involving flow dependent servers, is solved using an approximate Mean Value Analysis technique which is augmented for faster iterative convergence.

A distributed location server, for managing mobile users' location information and for aiding the mobile connection management tasks, is proposed. In order to hide the effects of mobility from the non-mobile network entities, the concept of a per-mobile software entity, known as a *representative*, is introduced. A mobile connection management scheme is also proposed for handling the end-to-end network layer connections in the present mobile environment. The scheme uses the representatives and a novel connection caching technique for providing the necessary realtime traffic support functionalities.

A prototype system, comprising of the proposed location and the connection managers, has been built for demonstrating the feasibility of the presented architecture for transporting continuous media traffic. A set of experiments have been carried out in order to investigate the impacts of various design decisions and to identify the performance-critical parts of the design.

To my parents

Acknowledgements

I would like to thank my supervisor, Dr. Andy Hopper, for his unceasing encouragement, support and valuable advice during my research. I am also indebted to him for allowing me to work for six months in the Olivetti Research Ltd. Cambridge. My thanks and appreciations also go to Prof. Roger Needham who, at different stages of this project, has provided moral and technical supports.

At different stages of this research, discussions with John Porter, David Greaves, Simon Crosby, Narmhee Lee, Roy Want, Glenford Mapp and Shaw Chuang have helped me enormously to develop my understanding of the subject. For reading and commenting on the earlier drafts of this dissertation, I thank David Greaves, Cosmos Nicolaou, John Porter, Noha Adly and Damian Gilmurray. I am also grateful to Chris Hadley, Graham Titmus, Martin Johnson and Piete Brooks for their excellent team work, yielding a stupendous computing infrastructure in the Computer Laboratory.

I will always remember Narmhee, Noha, Sanjay, Simon and Suresh for their help and friendship during my four years of inhabitation in the Computer Laboratory. It would be a futile attempt to name all my friends in Cambridge who, in their own ways, have moulded my perspective towards life. People who deserve special mention are Abhijit, Amaresh, Anjan, Dwaraka, Ganesh, Saibal and Tony.

With a risk of under-expressing their contributions, I would like to thank Preetha, Atul and Sunil for their endless support, care and love during my stay in Cambridge. Without them, this dissertation would have been non-existent.

I acknowledge my debt to the Indian people who, through their toil, have financed my education for nearly 24 years. Their sacrifice cannot be paid – I will honour it. Words cannot code my gratitude to my parents, brother and sister. They never ran out of encouragements during the most difficult and the vulnerable part of my stay in Cambridge. It is my father who taught me how to look at the apparently distinct line between success and failure as a blurred one. Distance could never deter my friends Indrajit and Rupam from being continuous sources of love and encouragement from India.

This work was supported by a Nehru Cambridge Scholarship, jointly funded by the Cambridge Commonwealth Trust and the Nehru Trust for Cambridge University. I am grateful to the Olivetti Research Ltd. and the Board of Graduate Studies for funding during my fourth year research. Thanks are also due to the following bodies for generously assisting me to attend conferences: Darwin College, Cambridge Philosophical Society, Cambridge Commonwealth Trust, Olivetti Research Ltd. and Computer Laboratory.

Declaration

Except where otherwise stated in the text, this dissertation is the result of my own work and is not the outcome of work done in collaboration. This dissertation is not substantially the same as any I have submitted for a degree or diploma or any other qualification at any other University. No part of this dissertation has already been, or is being currently submitted for any such degree, diploma or other qualification.

Publications

Materials presented in several parts of this dissertation also appear in the following publications.

S. K. Biswas, J. D. Porter and A. Hopper, "Performance of a Multiple Access Protocol for an ATM Based Pico-Cellular Radio LAN," *Proceedings of the Third IEEE International Symposium on Personal, Indoor and Mobile Radio Communications,* Boston, Massachusetts, October 1992, pp. 5.2.1-5.2.6.

S. K. Biswas, "Performance Modelling of Window Based Flow Control in a Multihop Packet Radio LAN," *Proceedings of IEEE INFOCOM,* vol. 3, San Francisco, March-April 1993, pp. 1222-1231.

S. K. Biswas and A. Hopper, "A Connection Management Scheme for a Mobile Radio LAN," *Proceedings of the IEEE International Conference on Personal Wireless Communications,* Bangalore, India, August 18-19, 1994.

S. K. Biswas, "An Agent Based Location Manager Design for a Mobile Radio LAN," Under Preparation.

Trademarks

Contents

List of Figures

List of Tables

Glossary of Terms

AAL	ATM Adaptation Layer
ATM	Asynchronous Transfer Mode
BISDN	Broadband Integrated Service Digital Network
BNT	Base Neighbour Table
BS	Base Station
CBR	Constant Bit Rate
CLP	Cell Loss Priority
CLS	Cluster Location Server
CM	Continuous Media
CRC	Cyclical Redundancy Check
CSMA	Carrier Sense Multiple Access
DID	Destination Identifier
DMR	Domestic Mobile Register
DMTM	Dynamic Minimum Tree Maintenance
ENS	Embedded Network Scheme
FDL	Fairisle Data Link
GFC	Generic Flow Control
GPS	Global Positioning System
GSM	Groupe Special Mobile
HLR	Home Location Register
HRLI	High Resolution Location Information

IP	Internet Protocol
IPC	Inter Process Communication
IPIP	IP-within-IP
IRQ	Interrupt Request
ISDN	Integrated Service Digital Network
ISR	Interrupt Service Routine
LAN	Local Area Network
LRLI	Low Resolution Location Information
MAC	Medium Access Control
MAN	Metropolitan Area Network
MICP	Mobile Internetworking Control Protocol
MNT	Mobile Neighbour Table
MR	Mobile Representative
MS	Mobile Station
MSC	Mobile Switching Centre
MSDL	Multi-Service Data-Link Layer
MSNA	Multi-Service Network Architecture
MSNL	Multi-Service Network Layer
MSSA	Multi-Service Storage Architecture
MSSAR	Multi-Service Segmentation and Reassembly
MU	Mobile Unit
MVA	Mean Value Analysis
NCLSP	Neighbour Cluster Location Server Pointer
PAS	Permanent IP-Address Scheme
PCN	Personal Communication Network
PDU	Protocol Data Unit
PLMN	Public Land Mobile Network
PLT	Payload Type

PRMA	Packet Reservation Multiple Access
PSTN	Public Service Telephone Network
PTM	Packet Transfer Mode
QoS	Quality of Service
QP	Query Partition
QPSK	Quadrature Phase-Shift Keying
RARP	Reverse Address Resolution Protocol
RF	Radio Frequency
RP	Roaming Partition
RPC	Remote Procedure Call
SAP	Service Access Point
SDU	Service Data Unit
SHM	Segment and Handover Manager
SID	Source Identifier
STM	Synchronous Transfer Mode
SU	Stationary Unit
TDMA	Time Division Multiple Access
UMRLC	Unseen Mobile Representative Location Cache
UMRLCP	Unseen Mobile Representative Location Cache Pointer
UNI	User to Network Interface
VBR	Variable Bit Rate
VC	Virtual Circuit
VCI	Virtual Circuit Identifier
VLR	Visitor Location Register
VLSI	Very Large Scale Integration
VMR	Visiting Mobile Register
VPI	Virtual Path Identifier
WAN	Wide Area Network

Chapter 1

Introduction

1.1 Background

Wireless personal communication networks (PCN) based on new digital technologies have emerged as an important field of activity in telecommunications. The third generation PCN systems aim for providing a ubiquitous computing and communication environment to a range of mobile users. At the same time, the design thrust for office automation systems gave rise to the development of the wireless in-campus and, more recently, in-building local area networks. The concurrent advancements of the VLSI and surface mount technologies are also contributing towards the proliferation of mobile computing, by scaling down the size and power consumption of the computing devices.

Although the initial PCN proposals have generally focussed on voice applications, it has been envisaged that these systems will be required to evolve towards supporting a wide range of telecommunication applications involving computer data, still and motion video and the other multimedia traffic. Location independence of the newly emerging mobile applications is also emphasised as one of the essential requirements of the PCN design exercise. The support of multimedia applications on portable computing devices, in a location independent manner, requires an integration of the present day wireless networks with the multimedia communication technology. Third generation PCN systems will also be required to co-exist with the broadband fixed communication networks which are capable of transporting constant and variable bit rate continuous media traffic. In order to maintain the application compatibility, the wireless personal communication networks will be expected to support Quality of Service guarantee for the multimedia applications.

1.2 Motivation

Although extensive research has been done for supporting mobility in the conventional IP environment, those work do not address the issues regarding continuous media traffic support over wireless. Clearly, the limited availability of communication bandwidth in a wireless network poses the primary hurdle for the implementation of the realtime services. Considering this, it has been decided that the wireless networks should be designed with an aim for providing at least qualitatively equivalent[1] services even if the quantitative equivalence is not feasible. This gives rise to the natural choice of ATM which is the de facto transfer mode for the broadband fixed communication networks.

However, the implementation of connection-oriented ATM services over wireless is much more challenging, partially due to the scarcity of bandwidth, and more importantly, because of the mobility of the end users of a connection. Furthermore, the implementation of Quality of Service requires a set of specialised wireless and wire resource allocation strategies which are far more complicated compared to the situations in a fixed network scenario. It has also been recognised that the QoS guarantee in wireless needs active supports from the lower protocol layers, like Medium Access Control (MAC), in terms of resource reservation and management. The handling of mobility, in an ATM environment, is closely related to the connection management functions and it needs a separate set of location management services to be developed.

1.3 Scope

Based on the above considerations, this dissertation proposes an ATM mobile LAN architecture for addressing a number of issues, involving multimedia traffic support in a mobile environment. The addressed issues range from the proposal for a statistical medium access scheme through the detailed designs of a location and a mobile connection manager, tailored for ATM traffic. By putting most of the mobility management functionalities at the network layer, this architecture attempts to insulate the higher layer protocols from the effects of mobility. As an extension of this mobility isolation strategy, the concept of a per-mobile software agent, known as a *representative*, is proposed. It is shown that with the assistance of these representatives, along with an appropriate location management strategy, it is possible for two network entities to communicate without knowing the location of their respective peer entities.

The presented connection management scheme is different from the contemporary approaches because it is tailored for the performance-critical multimedia traffic, especially in terms of *handover* performance. Low handover latency and scalability are achieved by using a novel connection caching scheme where the relevant parts of a mobile connection are setup in a predictive manner, even before the handover operation is initiated. Unlike most of the other

[1] Equivalent to the services offered by the fixed ISDN/BISDN networks.

existing systems, this design supports simultaneous handovers at both the ends of an ATM connection, when required.

Finally, the dissertation deals with a prototype implementation of the proposed location and connection management functionalities on an emulated radio ATM testbed. The implementation and its related experiments provide the performance and cost efficiency of the proposed management strategies. The prototype also serves as a means for identifying the performance-critical parts of the presented design.

1.4 Dissertation Outline

Depending on the specific interest the reader may choose one of the following *chapter routes*, without any loss of contextual continuity.

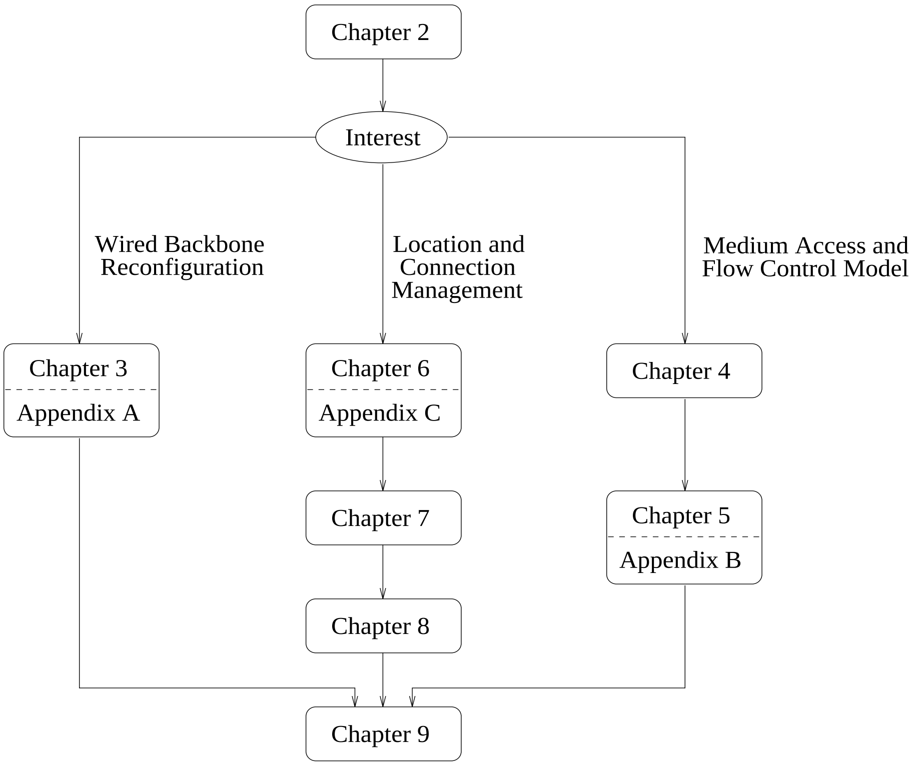


Figure 1.1: Structure of the dissertation

Chapter 2 reviews the recent development of personal communication networks and identifies the issues which will need to be addressed for supporting ATM in a mobile environment. In order to investigate few of these identified research issues, a mobile radio LAN architecture, capable of transporting ATM traffic, is proposed in the later part of the chapter.

Chapter 3 proposes a dynamic topology reconfiguration algorithm for the point-to-point wired backbone of the proposed radio LAN.

Chapter 4 explores the issue of medium access and presents a radio MAC scheme and its simulated performance figures.

Chapter 5 presents a hybrid modelling technique for analysing a window based flow control protocol, operating in a connection-oriented radio network.

Chapter 6 describes the design details of the proposed location and representative management strategies.

Chapter 7 outlines a representative assisted connection manager design for the proposed wireless ATM network.

Chapter 8 furnishes the implementation details of a prototype location and connection management system. This chapter also outlines the performance results, obtained experimentally from the implemented prototype.

Finally, Chapter 9 concludes the dissertation with a summary of the presented work and some suggestions for future work.

Chapter 2

Background and Addressed Issues

This chapter describes the background of the work reported in the later parts of this dissertation. It starts with a brief review of high speed multimedia networking and the current status of the mobile and ubiquitous computing. Few related research works are briefly discussed for identifying the recent trends in the integration of multimedia communication technology and the personalised mobile communication systems. In order to investigate few of the identified research issues, a mobile radio LAN architecture, capable of transporting ATM traffic, is proposed. The rest of the chapter deals with the network design issues and examines various implementation possibilities in the light of the existing approaches.

2.1 Realtime Communication

A considerable amount of research is currently being carried out in order to investigate and develop the multiservice networks which are suitable for carrying high bandwidth realtime traffic. This section discusses few of the issues related to *Continuous Media* (CM) and explains the role of ATM in transporting continuous media traffic in a packet switching environment.

2.1.1 Continuous Media

In their raw forms, continuous media typically exhibit a Constant Bit Rate (CBR). Audio, for example, can be represented as a periodic stream of eight bit or more samples and video as a periodic stream of frames of fixed size. Compression is one of the techniques used for avoiding unnecessary bandwidth consumption caused by the inherent redundancy of the continuous media streams. Silence detection for audio is a simple example of stream compression in a multimedia environment. For video however relatively more complex techniques like H.261 [CCITT 90b] and MPEG [ISO/IEC 91] are used for compressing the streams based on the chrominance sub-sampling, quantisation and the motion compensation mechanisms. After compression, a CBR stream often exhibits Variable Bit Rate (VBR) nature. In an extreme situation, an audio stream does not have any data sample during a silence and similarly, a video stream would not have any data when the successive video frames are exactly identical.

2.1.2 Asynchronous Transfer Mode

While the traditional circuit switched (also known as *Synchronous Transfer Mode* (STM)) networks offer better realtime performance, they do not share the network bandwidth efficiently and often provide a rather limited aggregate rate of data transfer to a group of end users. This problem is more severe for VBR traffic for which the allocation of bandwidth requires to be more dynamic than the CBR ones. While providing a better bandwidth sharing efficiency, the packet switched networks (referred to as *Packet Transfer Mode* (PTM)) can introduce excessive delay and jitter (variance of the delay) into a traffic stream. Large packet size also limits the granularity of multiplexing of the transmission medium. In order to overcome these problems associated with the STM and PTM, an intermediate strategy *Asynchronous Transfer Mode* (ATM) is chosen by the CCITT as the standard for transporting continuous media traffic over the broadband networks. ATM is a compromise between STM and PTM where the channel time is divided into small, fixed-length slots or *cells*[1].

In ATM, it is possible to dynamically allocate the physical medium to a new stream after each cell boundary. This feature of dynamic bandwidth allocation makes ATM particularly suitable for carrying the VBR continuous media traffic. If a stream comprises of bursty traffic, the unused channel bandwidth while the source is not busy, can be statistically allocated to some other streams for which the sources are busy. This however increases the stream jitter which is considered harmful for continuous media traffic. By the means of this fine grain channel multiplexing and statistical cell scheduling it is possible for the ATM networks to provide Quality of Service (QoS) guarantee to the end users, while accommodating a range of performance-critical traffic. More about the advantages and the disadvantages of ATM, along with a description of the multiservice protocols used on the experimental networks in Cambridge, can be found in [McAuley 90] and [Dixon 91].

Apart from the short cell size and the channel allocation at each cell boundary, the ATM networks maintain a lightweigth end-to-end *Virtual Circuit* (VC) [Leslie 83] for each data stream. This abstraction provides the end points a standard circuit while allowing the mentioned multiplexing to work on the physical channel. In this way each of the streams gets its share of the medium which is dynamically allocated after each cell transmission. Each ATM cell requires to carry a tag in order to identify the virtual circuit and that tag is known as a *Virtual Circuit Identifier* (VCI). A VCI of a cell is meaningful only for a single hop in a multi-hop connection and it is replaced at each of the subsequent hops. This operation, referred to as *remapping*, is performed by the switching elements between two successive hops of a connection. Like in the STM networks, an end user in ATM needs to

[1] A 5 octet header and a 48 octet payload have been chosen for Broadband ISDN [CCITT 90a], giving a 53 octet cell size.

setup a virtual circuit before the actual cell transfer takes place.

The CCITT and "ATM forum" [NCLATM 92, ATMUNI 93] standardised format of an ATM cell is shown in Figure 2.1. Apart from the VCI for identification of the virtual circuit, each fixed size cell header needs to carry some additional information in order to specify the end-to-end traffic characteristics and requirements. Each header is also protected by its own Header Checking Code (HEC).

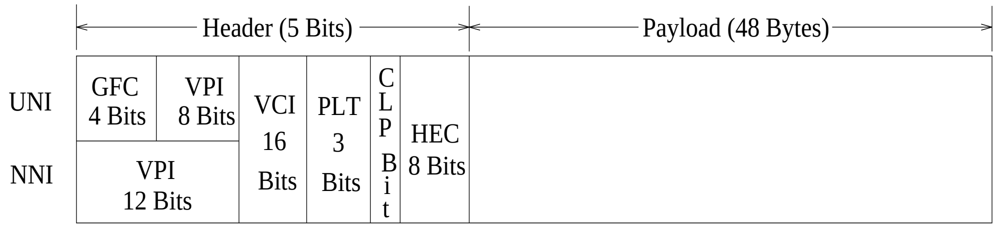


Figure 2.1: The CCITT ATM cell format (from [Hayter 93])

The channel identifier is partitioned into two parts: the Virtual Path Identifier (VPI) and the Virtual Circuit Identifier. It is possible to multiplex multiple virtual circuits onto a single virtual path between two points of the network. The entities between a virtual path end-points need to use and remap only the VPI field in order to route all the cells which belong to the corresponding virtual path. A 4-bit Generic Flow Control (GFC) field is used at the User to Network Interface (UNI). The Payload Type (PLT) distinguishes between the user cells and the management cells in order to treat them with the desired priorities. A single bit Cell Loss Priority (CLP) field is included for deciding which cell is to be dropped in a congested situation. Note that the mentioned cell format is still in the process of standardisation.

While sending data through an ATM network, it is necessary to segment and reassemble the large data blocks or frames at the source and the sink ends respectively. The standard defines a series of ATM Adaptation Layers (AALs) in order to provide these functions. AAL 1 and 2 are defined for supporting the CBR and the VBR continuous media traffic and AAL 3 and 4 are designed for the transport of data traffic. AAL 3 and 4 are recently replaced by a new one called AAL 5 which is both simpler and more efficient compared to AAL 3 and 4. A similar adaptation layer protocol AAL X is currently used by the XUNET project [Berenbaum 93] in AT&T Bell Labs.

As mentioned earlier, in order to fully exploit the ATM advantages it is necessary to carefully allocate the network resources to the streams of varying traffic types and requirements. This also requires the appropriate admission control and policing schemes [Bae 91] for providing Quality of Services. When the network receives a call setup request with a specified QoS requirement, the call is accepted only if the network decides that it has sufficient resources to support the call. This process of deciding about accepting or rejecting a call is referred to as *call admission control*. Once a call is admitted, based on the requested performance level, it is necessary for the network management services to ensure that the call is not consuming more resources than what it was initially allocated with. The process of ensuring this is known as *policing.*

2.1.3 Multimedia Computing

The integration of continuous media in a traditional data computing environment has opened up the possibility of creating the multimedia applications for handling a variety of information forms including text, graphics, voice, still video images and full motion video. With the recent advent of the hardware and software technologies, the distributed multimedia applications are becoming feasible and popular, at the same time. In addition to the continuous media networking, a distributed multimedia architecture requires an efficient network storage service to cope with the data, generated and used by the multimedia applications. A comprehensive review of the issues, related to the design of a continuous media file server, is presented in [Jardetzky 92]. This research demonstrates how a specially designed continuous media file server may be integrated into a generic distributed multimedia environment, while providing the QoS guarantee to the end users. In another work, [Lo 94] describes a multi-layer design approach to support the multiple service data types by a generic storage system which is termed as Multi-Service Storage Architecture (MSSA). While handling the continuous media, stream synchronisation is another important issue which is to be addressed in order to achieve consistent presentation of the streams at the end devices. An example of it is the synchronisation of the voice output with the lip movement of the speaker during a multimedia conferencing session; this is commonly known as *lip-sync.* In [Sreenan 93] a range of techniques are discussed and investigated for providing the necessary synchronisation operations in a distributed multimedia environment.

The operating system support issues for multimedia is currently being studied in the Pegasus [Leslie 92] project. A customised kernel [Fairbairns 93], known as Nemesis, has been designed to support the reception of video streams with a standard compression technique like JPEG. [Hyden 94] presents a flexible resource management paradigm, based on the notion of quality of service, with which it is possible to provide the scheduling support required by continuous media applications. The mechanisms which are required within an operating system to support the multimedia applications are described, and the design and implementation of a prototypical kernel which implements them are presented in this work.

The Olivetti Research Pandora project [Hopper 90] has investigated the integration of audio and video in a distributed environment. The high speed Cambridge Fast Ring [Hopper 88] and the Cambridge Backbone Network [Greaves 90] are used for CM traffic transportation between the multimedia workstations and the relevant network storage servers. Typical applications of the Pandora system includes video conference, video mail and accessing the television and radio tuners through the standard X-window based application interfaces. A detailed investigation report about the handling of audio and video streams in the Pandora environment can be found in [Jones 93]. Medusa [Wray 94], a peer-to-peer distributed architecture for controlling the networked multimedia devices, is currently being investigated at the Olivetti Research Ltd. This project explores a novel approach of handling multimedia data in a modular way where most of the realtime processing operations are performed by a set of active software modules in a secured networking environment.

In [Hayter 93], a Desk Area Network (DAN) architecture is presented where the workstation components like CPUs, memories, caches, network interfaces and the other peripheral devices are connected to each other by a high performance ATM interconnect. In this extremely decentralised environment, the interconnection network can be suitably programmed for routing multimedia streams to and from the peripheral devices. [Pratt 94] presents the "ATM Camera" which converts an analogue video signal into a stream of ATM cells and connects directly to the switch in the DAN. In case when the video stream has to be delivered to an application, the interconnection switch is programmed to deliver the stream to the processor. On the other hand, if the video data is only to be displayed in the local VDU, it is directly sent to the local frame store, bypassing the CPU. The aim of the DAN project is to integrate the data transfer mechanisms, used in the end system and in the local and wide area networks, to build a multimedia workstation which can effectively utilise the high speed networks.

Multimedia-on-demand is one of the newly emerging group of applications, suitable for the high-speed ATM/BISDN networking environments. Such a research is reported in [Ott 94] where two types of applications are considered. In the first scenario, "multimedia documents" are processed by the workstations connected to an ATM network, with image and video information being retrieved on demand from the remote multimedia servers. In the second, subscribers with a terminal/TV connected to the broadband network access a video information database for on-demand viewing of the movies and the other video materials. This work identifies the networking and the application related issues for a multimedia-on-demand architecture and attempts to address few of the related issues by building a simplified prototypical system.

2.2 Mobile Communication

The first cellular[2] radio system [Steele 92] in Europe was installed in Scandinavia in 1981 and it served initially only a few thousands subscribers. Over the last decade the cellular telephone networks became more pervasive and millions of mobile subscribers are included into it. The original purpose of the cellular radio was to provide telephone services to a large population of users in their automobiles [Young 79]. As the popularity of cellular radio escalated in the 1980s, this approach promised increasing capacity through the splitting and reduction of the transmission cell sizes. With cells becoming smaller and smaller, it becomes

[2]The communication paradigm where a network is covered by multiple radio *transmission cells* with a control point, known as a base station, at the centre of each such cell. While the base stations are connected by a wired backbone, the mobile subscribers communicate to the rest of the network through the base stations. Note that the term *transmission cell* should not be confused with the *ATM cells*, specified previously.

increasingly difficult to place the base stations at the locations that offer the necessary radio coverage. The situation is worse in the highly congested cities where the capacity requirements are particularly pressing.

These limitations of the *First Generation* cellular systems motivated the design of a *Second Generation* network whose principal goals are to provide higher capacity in a wider system with full international roaming facilities. In Europe, North America and Japan different approaches are taken to fulfill these requirements. Groupe Special Mobile (GSM) [Mallinder 88] is the European approach where the system capacity is improved by new spectral allocation and using a digital technology for information transportation. A hierarchical network signalling strategy is adopted for supporting continental roaming of the mobile subscribers. In North America, an Electronic Industry Association Interim Standard (IS-54) [IS54 89] has been proposed for mitigating the problems of the first generation cellular systems by adopting a dual-mode technology which makes use of the existing analogue systems but with radically different advanced technologies. In North America, where no new cellular frequency band is available, higher capacity is derived from intensive application of the advanced transmission techniques including efficient speech coding, error correcting channel codes and bandwidth efficient modulations. It is noticeable that in North America a collection of different systems are expected to co-exist, as opposed to the single digital system like GSM in Western Europe.

Although the second generation cellular networks provide high capacity and improved support for mobility, the difference in standard prevents the mobile users from roaming between the countries with heterogeneous cellular approaches. The vision of the *Third Generation* [Goodman 91b] is a unified wireless access to the advanced information services. The third generation network subscribers are also expected to be able to handle the multimedia traffic in a location independent manner. This new *Personal Communication Networks* (PCNs) will merge the separate first and second generation cellular and cordless [Swain 90] services, and will also encompass the other means of wireless access such as paging, dispatch, public safety and wireless local area networks. It is envisaged that all the first and the second generation cellular networks will be completely integrated with the BISDN and the other existing broadband computer data networks in order to provide transparent services to the end users. To meet the challenges of the third generation, researchers have begun in the past few years to study the *microcells* [Steele 85, Steele 89] with mobile terminal and base station power levels at least two orders of magnitude lower than in today's cellular systems. While the first and the second generation cellular systems are based on cell dimensions of the order of one to several kilometres, the minimum distance between the base stations of the third generation systems will be of the order of 10 metres to 100 metres. To support the enhanced functionalities in these micro and pico cellular environments, the call control and call management functions will have to be more decentralised compared to the earlier generation systems.

2.3 Related Research

This section outlines few of the projects which are carrying out relevant research, aiming for the mobile services both in the local and wide area environments. Wherever applicable, each system is discussed with respect to the following issues.

- Physical layer specifications.
- Medium Access Control (MAC) mechanism.
- Network architecture and management policies.

2.3.1 Groupe System Mobile (GSM)

A general objective of the GSM Public Land Mobile Network (PLMN) system is to provide a wide range of voice and data services which are compatible with those offered by the existing Public Service Telephone Network (PSTN) and the Integrated Service Digital Network (ISDN). Another goal of it is to provide access to any mobile subscriber of any country which operates GSM and provides facilities for automatic roaming, locating and updating the mobile subscribers' status. The GSM system operates in two pair bands: 890-915 MHz for the uplink transmission, where the mobile terminals transmit and the base stations receive, and 935-960 MHz for the downlink transmission, where the base stations transmit and the mobile terminals receive. The medium access scheme, adopted by GSM, is Time Division Multiple Access (TDMA) with 8 timeslots per radio carrier frequency [Steele 92]. The duration of each timeslot is approximately 0.58 ms, which leads to a TDMA frame duration of approximately 4.6 ms.

A simplified model of the GSM architecture is illustrated in Figure 2.2. The Mobile Stations (MS) usually consist of more than one piece of equipment such as a telephone set and a Data Terminal Equipment (DTE). There are various types of mobile stations [Imielinski 92b] such as the vehicle mounted stations, the portable stations and the handheld terminals. The Base Stations (BSs) are associated with the radio channel management including channel allocation, link quality supervision, transmission of associated signalling information and broadcast messages, as well as controlling power levels and frequency hopping. The BSs are also responsible for intra cell handover initiation to a better RF channel, as well as the data and the signalling encryption.

The Mobile Switching Centre (MSC) is linked to the BSs via the 'A' interface and performs all the switching functions needed for the operation of the mobile stations in the group of cells it services. The functions of an MSC include call routing, call control and the procedures needed for internetworking to connect to the PSTN, ISDN and the future BISDN networks. The inter cell handovers[3] are also supervised by the MSCs.

[3] GSM also supports a intra cell handover which is a call transfer from one channel to another within

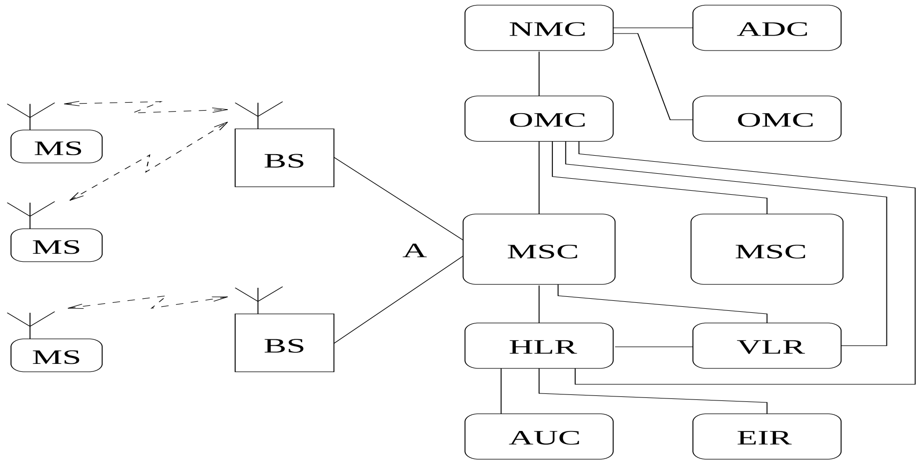


Figure 2.2: Simplified architecture of GSM PLMN system (from [Steele 92])

The Home Location Register (HLR) is a database unit for storage and management of the mobile subscribers' location information. The location management requires an MS to periodically inform the PLMN about its geographical location so that the network can update the contents of the HLR. The location information stored in the HLR is used for routing the calls and to perform various kinds of handovers, when necessary. The HLR contains the International Mobile Subscriber Identity (IMSI) number which is used for the authentication of the subscriber by his or her AUthentication Centre (AUC). In order to facilitate the roaming and the location registration procedures, the PLMNs are divided into disjoint geographic areas characterised by unique identifiers broadcast regularly to all the MSs via the so-called Broadcast Control Channels (BCCHs) conveyed over reserved RF carriers. Every subscriber is registered to a home network (which is a geographic network partition) and the subscriber's communication profiles [Badrinath 92] are stored into his or her HLR. The Equipment Identity Register (EIR) allows for stolen, fraudulent or faulty mobile stations to be identified by the network operators.

The Visitor Location Register (VLR) is the functional unit that attends to an MS, operating outside the area of its home network. The visiting MS is automatically registered at the nearest MSC and the VLR is informed of the MS's arrival. A roaming number is then assigned to the MS and this enables the calls to be routed to it. The Operations and Maintenance Centre (OMC), Network Management Centre (NMC) and ADministration Centre (ADC) are functional entities through which the system is monitored, controlled, maintained and managed. More about the mobile registration, call setup and handover procedures are discussed in [Mallinder 88] and [Targett 88].

---

the same cell. This happens when a channel cannot be used any longer due to interference or maintenance problems.

2.3.2 WINLAB Cellular Packet Switch (CPS)

Anticipating the demand for the integration of data and continuous media traffic in the mobile networks, researchers at Rutgers University WINLAB are investigating [Hellstern 93] the design of a third generation cellular packet switch to implement the functions provided by GSM. Introduction of packet switching is expected to serve to:

1. Reduce substantially the burden on the central controller of radio resource management.
2. Efficiently multiplex voice, network control information and a wide variety of user data transmissions.
3. Harmonise with the anticipated BISDN which will be based on packetised ATM communications.

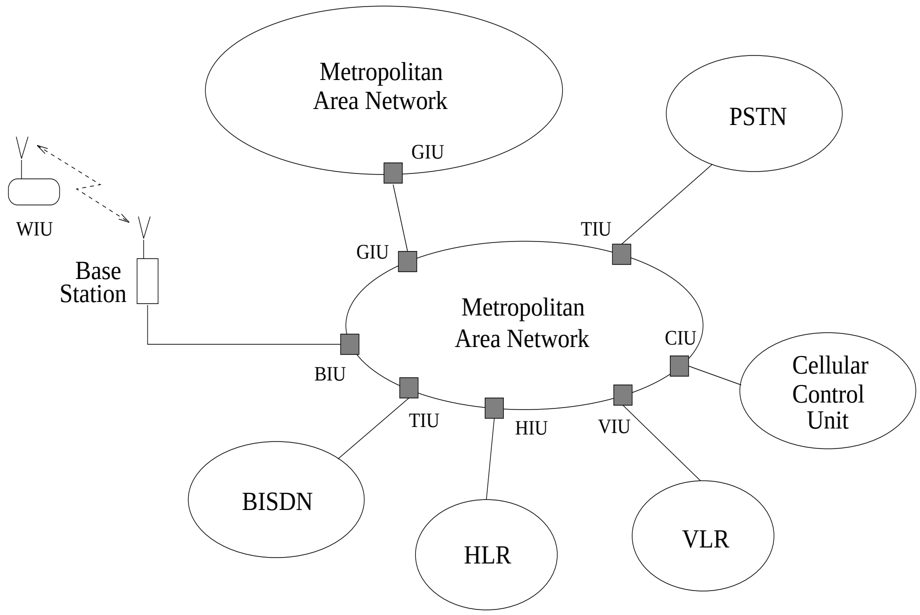


Figure 2.3: Architecture of the WINLAB Cellular Packet Switch (from [Goodman 91b])

In the cellular packet switch of Figure 2.3, all the base stations in a service area are connected to a Metropolitan Area Network (MAN) which is also connected to the trunks of the public switched telephone network and the facilities of a BISDN. Network interface units associated with each Wireless terminal Interface Unit (WIU), central office Trunk Interface Unit (TIU) and Base station Interface Unit (BIU) contain simple routing tables. Based on the information in these tables, which change in a very dynamic manner, the interface units directs each information packet to the correct destination even as users move frequently between base station service areas. The principal call processing task of the cellular controller is to assign a virtual circuit identifier to a wireless terminal and a PSTN or BISDN trunk at the beginning of each call. Other interfaces to the MAN include a VLR Interface Unit (VIU), a HLR Interface Unit (HUI) and a Gateway Interface Unit (GUI) for connection to the other cellular packet switches. Together, they implement in a non-hierarchical manner most of the functionalities described for GSM in Section 2.3.1.

The channel access scheme chosen for CPS is the Packet Reservation Multiple Access (PRMA) [Goodman 91a], which can be viewed as a combination of TDMA and slotted ALOHA [Tasaka 83, Lam 80]. Like TDMA, the channel bit stream is organised into slots and frames. Within a frame, terminals recognise each slot as either available or reserved on the basis of a feedback packet broadcast in the previous frame from the base station to all of the terminals. As in slotted ALOHA, the terminals with new information to transmit contend for access to the available slots. At the end of each slot, the base station broadcasts the feedback packet that reports the result of the transmission. A terminal that succeeds in sending packet to the base station obtains a reservation for exclusive use of the corresponding time slot in the subsequent frames.

The primary goal of the WINLAB CPS architecture is to implement the third generation cellular network functionalities in a packet switch environment. More about the location and connection management functions within the CPS can be found in [Hellstern 93] and [Pollini 92].

2.3.3 Mobile IP

While the last two reported projects investigate the design issues of the second and the third generation cellular networks, the current section discusses few research works which aim for providing wireless data services, using the Internet Protocol (IP) [Comer 91], in local and wide area environments. Although various works [Aziz 94, Perkins 92a, Ioannidis 92, Awerbuch 91, Bhagwat 93, Perkins 93, Perkins 92b] have been reported in this area, only three of them are chosen for discussion in this section.

Mobile IP Routing in LANs

In [Cohen 92], a campus wide local area network is considered where the entire campus is covered by a wired backbone, typically containing workstations, file servers and gateways. The Base Stations (BS), connected to the backbone, are used for providing network access to the roaming Mobile Units (MUs). The intent of this research is to provide IP-based efficient and smooth communications between the MUs, the static units of the backbone (such as various servers) and the other computers in the Internet. Three different mobile addressing and routing schemes for providing the standard IP services [Postel 81] in a wireless environment are proposed in this research.

Permanent IP-Address Scheme (PAS): Under this strategy, each MU is assigned a permanent IP-Address from the campus address space. On power-up, an MU notifies the reachable base station about its IP-address and the BS enters this address as a *host entry* in its routing table. The information is then propagated among the hosts and the gateways inside the network as part of the regular network routing information update procedure. When the MU moves, the new BS updates its routing table by adding the new entry and the old one deletes the corresponding entry from its routing table. Since these updated information propagate as in the power-up case, the old BS soon receives the whereabouts of the corresponding MU. Hence, the packets arriving to the old BS could promptly be rerouted to the new one. Although this scheme offers simplicity, it suffers from severe message and storage complexity with increasing MU population and their roaming frequency.

Temporary IP-Address Scheme (TAS): In this scheme, soon after a power-up, an MU is assigned with a temporary IP address which remains valid as long as the MU remains up. Every BS maintains a *bank* of IP-addresses which are permanently associated with it. On power-up when an MU identifies[4] itself to a BS, the latter assigns the mobile one of the available addresses in its bank. From this point onwards the mobile unit starts communicating smoothly with this assigned IP address. When the MU moves from its *source* BS (the one which has assigned its IP-address) to another BS, instead of updating the routing tables network-wide, the source base station reroutes all the packets (which are destined to the MU) to the appropriate new base station. This needs the source base station to be aware of the BS that currently serves the IP-address in question.

The main advantage of this scheme is that no routing table updates are incurred due to a power-on. The source BS forwarding scheme is further advantageous because routing table updates do not occur even after subsequent roaming of the mobile units. The disadvantages however are quite severe considering the nonoptimal routing of the source forwarded packets. Routing dependency on the source BS also suggests that a crash of that entity will stop all the communication with the relevant MU. The recycling of the IP-addresses makes the scheme more complicated in terms of network management and control operations. The consistent reuse of the IP-addresses needs an expensive *address consultancy* service on a per-MU address assignment basis. To summarise, at the cost of address recycling management and less optimal routes, a scheme that does not require routing table updates can be achieved.

Embedded Network Scheme (ENS): This is a refinement of the previous Temporary IP-Address Scheme. The strategy is based on a concept of an embedded network, i.e., a network that contains its own hosts, BSs, MUs and gateways and has its own network protocol. Each of the network entities is assigned a permanent IP-address. In addition, each entity also has an embedded network address which is a two level construct *(b, u)* where b is the BS identifier and u is the MU identifier (this is a local id. of an MU). The u component need not be network-wide unique but the *(b, u)* address must be unambiguously resolvable at any time within the embedded network. The base station and the gateways are addressed

[4] This identification is done via the MUs hard-coded Electronic Identifier (EID).

as *(b, 0)* and *(g, 0)* respectively where b and g are taken from the same address space. Each gateway of the embedded network maintains a mapping between the IP-address and the *(b, u)* address of the MUs within the embedded network. To route a packet, it is forwarded towards the base station b and once reached to the BS, it is forwarded to the MU, identified by u. Similarly, when an MU intends to forward a packet outside the network it sends it to the appropriate gateway and then the gateway forwards it to the relevant destination.

When an MU moves from the base station b_1 to the base station b_2, the MU's address changes from (b_1, u_1) to (b_2, u_2). This change must be reflected in the mapping tables of all the gateways of the embedded network. It is a joint responsibility of the base stations to update the gateway address mapping entries whenever a mobile moves from one BS to another. It is possible to maintain the continuity of a connection even while the gateways update their tables. Every BS maintains a mapping between the u's and IP-addresses for the MUs within its jurisdiction. Before sending an IP packet to an MU, every time a BS checks and verifies the correspondence between the IP-address and the corresponding u values. Any discrepancy indicates that the MU, bearing the destination IP-address, has moved to some other BS. Note that this will happen only when a gateway contains a transient out of date mapping table entry. In this situation the BS talks to an address consultant in order to know about the new *(b, u)* associated with the given IP-address. Once it receives the new *(b, u)*, the packet is forwarded accordingly. For efficiency, the BS can cache for a while the mapping of a recently departed MU.

The advantage of this scheme is its efficient handling of packets once they enter within the embedded network. While assigning permanent MU IP-addresses, the packet rerouting is similar to the previous scheme, although the address recycling is much simplified due to the two-level addressing structure. The ENS scheme adds some additional overhead to each IP datagram for putting the *(b, u)* entry onto it, but this is done only over the backbone and not on the wireless. Although it suggests a novel addressing strategy for efficient packet routing, the scheme does not scale very well for bigger networks with large number of mobile and base entities.

Mobile Internetworking

The IP-within-IP (IPIP) protocol, described in [Ioannidis 91], is used to *tunnel* the IP datagrams from one base station to another or from one network to another. This work also proposes a Mobile Internetworking Control Protocol (MICP) which is used by the base stations to exchange control information, and by the MUs to signal to their BSs that they have changed cells. A set of MICP messages and their usage have been specified in order to track the mobile users in a distributed manner. The proposal also argues that the network layer is the most appropriate position to put the additional codes for handling mobility in an IP environment. Handling mobility at the network layer ensures the re-usability of all the existing IP codes above the network layer. This also enables most of the existing applications to run without even realising the existence of mobility in the network. This design also tries to put most of the added functionalities on the mobile units rather than changing the existing IP gateway and the routing protocols considerably. Considering the mentioned overheads of the dynamically changing mobile IP addresses, the design adopts a fixed IP addressing strategy for each individual mobile unit.

According to IPIP, when an IP datagram reaches the output routine of the IP network layer code, the routing and the MU location entries are consulted. If the destination MU is known to be within the jurisdiction of a known BS, the datagram is encapsulated in an IP datagram of a specific type and sent to the destination BS. Upon receipt by the remote BS, the IP code hands the datagram over to the IPIP protocol module. The latter strips the IPIP header and after decrementing the time-to-live field, it feeds the packet (which has now the real MU destination and the source IP addresses) back to the IP queue so that it can be delivered. When a BS has to relay a packet to an MU which does not exist in the service area of the base station, it must search for the location of the MU by sending queries to the other BSs. This particular feature prevents the scheme from being very scalable in a wide area environment. Details about the MICP messages and the location management procedure are out of the scope of this discussion and can be found in [Ioannidis 91].

Host Migration Transparency

While maintaining the philosophy of handling mobility at the network layer, the mentioned scalability problem for IPIP is removed by the Virtual Internet Protocol (VIP) approach, presented in [Teraoka 93]. In this scheme, the network layer is divided into two sublayers: the *Virtual Network* (VN) sublayer and the *Physical Network* (PN) sublayer. While the VN layer address of a mobile entity is permanently assigned, the PN layer address of it depends on the instantaneous location of the entity. Each mobile unit is registered with a home network and its VN and PN layer addresses are same when the mobile is residing in its home network. All the upper layer protocols use the VN address to send an IP datagram to the relevant destination mobile. The mapping between the VN address to the PN address is performed by using a *propagating cache method*, where the cache size is made independent of the network size and the total number of migrating mobile units in the network. The protocol is also designed in such a way that the cache consistency does not depend on any global time-stamping strategy.

Once a datagram with the VN address of the destination is handed to the network layer code of the source entity, it tries to map the VN to PN address if there is an appropriate entry in its Address Mapping Table (AMT). Otherwise, the source host assumes that the destination mobile unit is currently residing in its home network and then forwards the packet to that home network. In this case, either an intermediate gateway or the home network of the destination mobile unit performs the address conversion. The AMT entry for a mobile unit propagates across the wide area network as the MU communicates with the other hosts. The AMT entry also propagates due to the control packets of the connection/disconnection procedures.

The work demonstrates that this propagating cache mechanism, in conjunction with the virtual networking concept, can provide complete host migration transparency to the upper layer protocols while maintaining modest network layer overhead for handling mobility. Two different prototypes are implemented in order to investigate the feasibility of the scheme in an IP networking environment. It is also shown that the scheme scales up well, while providing adequate cache consistency and fault tolerance.

2.3.4 NEC Wireless ATM

Although the Mobile IP approach provides compatibility with the existing worldwide Internetwork, it fails to meet [Minzer 89] the performance requirements of continuous media traffic, as specified at the beginning of this chapter. This gives rise to the requirements for implementing connection-oriented wireless networks in order to provide Quality of Services guarantee to the end applications of the transported streams. While providing continuous media support, the connection-oriented approaches also show the promise of a better integration of the third generation Personal Communication Networks (PCNs) with the future BISDN systems. Based on the above considerations, the research reported in [Raychaudhuri 94] proposes an ATM cell relay paradigm to be adopted as the basis for the next-generation wireless transport architectures. It is argued that in addition to its regular advantages, the use of ATM switching for inter transmission-cell traffic also avoids the crucial problem of developing a new backbone network with sufficient throughput to support intercommunication among the base station cells.

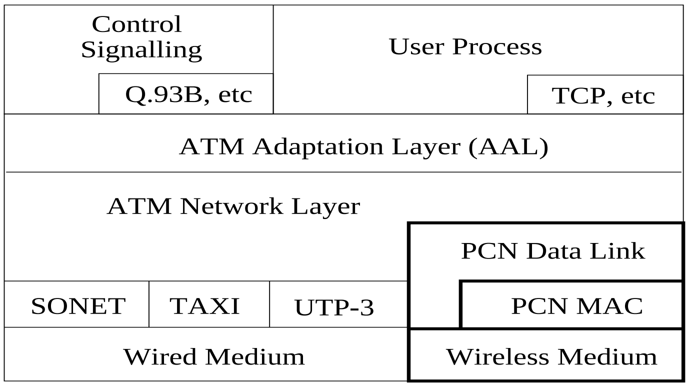


Figure 2.4: Wireless addition to the ATM protocol stack (from [Raychaudhuri 94])

A protocol layering strategy, harmonised with the ATM protocol stack, is proposed. As illustrated in Figure 2.4, the wireless specific physical, medium access control and data-link layers below the ATM network layer are added to the standard ATM protocol stack. This implies that the regular ATM network layer and control services such as call setup, VCI/VPI addressing, cell prioritisation and flow control indication will continue to be used for the mobile services. However, the baseline ATM network and signalling protocol will have to be augmented to support specific mobility related functions such as address registration, roaming, handover and QoS renegotiation in response to the channel impairment. Alternative AAL protocols may also be considered for customised support of the wireless applications.

This work also identifies the Medium Access Control (MAC) as a major technical issue related to the multiservice PCN design and proposes a Dynamic TDMA scheme for carrying both the CBR and VBR traffic in a PCN network. The scheme follows a combination of *circuit mode* reservation of slots over multiple TDMA frames for CBR voice calls, along with dynamic assignment of the remaining capacity for the VBR or the packet data traffic. For VBR, the allocation is based on a suitable statistical multiplexing algorithm in which the available capacity is prorated among demands, based on some usage parameter control values declared during the corresponding call establishment. The scheme also advocates for different access layer queueing strategies [Raychaudhuri 93, Panwar 88] for CBR and VBR traffic in order to expedite the realtime packet transportation. A simulation study of the proposed access scheme indicates that the demand assigned protocols of this type are the viable candidates for multiservice PCN.

2.3.5 Olivetti Radio ATM LAN

As discussed in Section 2.2, the third generation personal communication networks are expected to merge with the existing local area networks which are deployed for carrying multiservice traffic in the in-building environments. The work, reported in [Porter 94], presents the design issues of a wireless ATM LAN, currently being developed in the Olivetti Research Ltd. Cambridge. The Olivetti four, eight and sixteen port ATM switches [Milway 94] are used for constructing the LAN backbone and each base station is connected to the backbone by a point-to-point co-axial link. All the base stations operate on the same radio frequency[5] and so there is no hard boundary between the used small size cells. It is argued that a single frequency for all the transmission cells improves the handover performance by avoiding any physical layer operations, necessary for the handovers. This single frequency approach, along with the point-to-point backbone design, provides a modular and extensible wireless LAN where the network can be extended just by adding new base stations to the backbone switch fabric.

A network partitioning scheme is implemented in order to facilitate the mobile location and connection management operations. Mobile location information within such a partition, known as a domain, is managed by a *Domain Location Server*. In order to aid the location and the connection management tasks, a software agent is deployed for each of the mobile units in the network. Such an agent is used for handling the connections and the location information regarding its client mobile unit. The work also investigates the necessity of redefining the QoS parameters in a wireless ATM environment. It is identified that the QoS renegotiation strategies, based on the frequently varying wireless network conditions

[5] A carrier frequency of 2.45 GHz with 10 MHz radio channel bandwidth is chosen. This is currently an unlicenced band in UK, available for short-range data links within buildings.

and the user mobility, are much more necessary in a wireless environment compared to the conventional wired ATM networks.

A wireless ATM cell format, complying with the CCITT specifications, is proposed. With an aim for keeping mobility transparent to the end-points as much as possible, it is argued that the VCIs used by the end-points should not change during a connection handover. This requires that the allocation of the VCI space should remain valid as the mobile moves through different transmission cells within a domain. It is also desirable that the translation of the VCI space due to movement between domains should be as simple as possible. The proposed wireless ATM cell format tries to ensure these features for keeping mobility transparent to the stream end-entities.

Olivetti radio ATM system uses a slotted ALOHA with exponential back-off as the MAC layer protocol. An ATM cell is used as the slot size where the slot structure is maintained by re-synchronising the mobile units to the base stations on each reception. It is argued that ALOHA is quite robust in an overlapping transmission cell environment. Although theoretically pure slotted ALOHA has a maximum throughput of $1/e$, in a practical implementation there is the possibility of *capture* [Cidon 85] in which a collision between two sources is not necessarily destructive [Namislo 84]. Simulation of a realistic propagation model [Porter 94] suggests that throughput of approximately half the bit-rate are achievable. The combination of capture and large scale diversity also has a large impact on the overall throughput.

2.3.6 Active Badge

While discussing the previous systems, mobile location management is identified as one of the most crucial operations which determines the performance of the services offered to the mobile users. The Active Badge system [Want 92], designed at the Olivetti Research Ltd. Cambridge, provides a useful means of locating individuals within a building by determining the location of their Active Badges. These small devices, worn by the individuals, transmits a unique infra-red signal once in every 10 seconds. Each office within a building is equipped with one or more networked sensors which detect these transmissions. The location of a badge can be determined on the basis of the information provided by these sensors. The badge also has two small push-buttons which when pressed cause it to transmit immediately. The buttons can be used to send simple input information to the system. The descriptions of the used infra-red protocols can be found in [Harter 93].

Teleporting, one of the numerous applications [Harter 94] of the Active Badge system, is used as a means for providing ubiquitous computing facilities where the system enables the applications to be mobile without having the *real* mobile computing devices. When the teleporting is enabled, it is possible for the X window environment of a badge wearer to follow the owner and to pop up at the badge's nearest possible X terminal. This requires each X terminal to be tagged with a Transmit Only Badge (TOB) so that the system can identify the relative locations of a user (badge wearer) and the usable X terminals. The badge buttons can be used to override an automatic terminal selection when the user wants his or her applications to be teleported on a specific terminal. Such a system is in daily use at Olivetti Research Ltd. For the details about the teleporting design issues, the reader is referred to [Richardson 94].

2.4 Research Scope

While reviewing the projects in the last section, the necessity of integration of multimedia with the third generation wireless PCN systems has been identified. Since most of the second generation designs are already standardised, there is very little room for ATM innovation within the second generation mobile radio systems. ATM however is expected to have strong impact [Armbrüster 92] on the concepts and the interfaces within the framework of the evolution into the third generation mobile systems and personal-mobility services. Although few prototypical systems have been built for investigating the feasibility of ATM in a wireless environment, the literature shows that the following research areas are still considered to be open for further investigations.

2.4.1 Medium Access Control (MAC)

The MAC technique used for mobile ATM will have a significant impact on user performance, system capacity and mobile terminal complexity, and is therefore an important design parameter. It is being envisaged that in order to transport the connection-oriented CBR and VBR traffic, the adopted MAC approach must provide mechanisms to deal with each of these BISDN type services. It is also recognised that to provide per-connection QoS guarantee over wireless, it is necessary to have a MAC protocol with sufficient reservation control on the raw wireless medium resources. While realising the poor performance of the wireless media compared to the wired ones, it is identified as a design challenge to come up with a family of *multimedia capable* MAC layer protocols with sufficient degree of transparency for many of the existing ATM applications.

2.4.2 Location Technology

Although GSM proposes a fairly general architecture for handling location information in a wide area cellular environment, it is very much customised for the circuit switched voice networks. The discussed mobile IP protocols, on the other hand, deal with the location and routing strategies assuming a completely datagram based communication paradigm. Both in the host migration [Teraoka 93] and the IPIP [Ioannidis 91] cases, each IP datagram needs location service, in order to be routed to the destination. The in-band signalling approach of IP is exploited in order to develop and maintain the location information database over a mobile Internetwork.

For mobile ATM however the call-by-call routing needs a separate and stand-alone location service to be present in the network. This leads to the realisation that in order to support ATM routing and the necessary connection maintenance operations, the user mobility information needs to be dealt with by a different approach than the ones used for the mobile IP implementations. This also gives rise to the necessity of choosing proper addressing, naming and network partitioning schemes to aid the mobile location management operations. It is found that several addressing concepts [Cohen 92], used for the mobile IP paradigm, map very well for the connection-oriented mobile networks. The mobile location information distribution schemes (e.g. HLR, VLR etc), adopted by GSM, also provide a basic framework for a customised location manager design in a mobile ATM environment. Based on these broad architectural frameworks, it is necessary to investigate the alternative approaches for location management technology which are tailored for the wireless ATM communication paradigm.

2.4.3 Connection Management

The implementation of ATM in the third generation PCN systems requires careful redesign of the network layer signalling and the connection management services. It is recognised that a customised packet header for the wireless data-link layer encapsulation is necessary for the standard ATM cells to be usable in wireless. Although it should be used as a last resort, if necessary for better channel reservation and access efficiency, the 48 byte wired ATM payload may have to be further divided (e.g. 24 or 16 bytes [Raychaudhuri 94]) in the wireless medium. The format and management of the VCI space also need to be reconsidered for providing better handover control, as proposed in [Porter 94]. Routing in a connection-oriented wireless environment needs a radically different approach (compared to the IP datagram-oriented designs) where the call establishment and maintenance procedures require sophisticated location and connection management services from the network.

Literature also indicates [Pollini 92] that handover is one of the most crucial management operations which dictates the mobile connection management designs. This is more so when it is necessary to provide QoS guarantee during a handover. The resource reservation mechanisms in a wireless network are noticeably different from its wired counterpart, mainly because of the multiple access nature of the former. Mobile movements also require the resource reservation to be predictive and stochastic in nature. As mentioned in Section 2.4.1, guaranteeing QoS to the mobile users requires active support from the adopted medium access layer protocols in a PCN system.

2.4.4 Adaptation Layer

Because of the frequent topology reconfigurations, caused by the mobile movements, ATM cell out-of-ordering is quite common in a wireless environment. This is more so during a handover process when the cells of a single connection travels through different paths before and after the handover. AAL 5, which is proposed for the wired ATM networks, does not specify any protection mechanism for cell out-of-ordering. If cells are out-of-ordered, like in the cell loss situations, the corresponding block CRC will fail and the higher layer protocols have to recover from it. In order to reduce the performance degradation caused by too many block retransmissions, alternative AAL protocols need to be considered for the customised support of the wireless applications.

2.4.5 Quality of Service

One of the main attractions of ATM is that the out-of-band control and the connection-oriented approach allow a Quality of Service (QoS) to be associated with each connection. Although ideally the wireless networks should provide the same QoS interface as the wired networks, in the context of mobility it is necessary to add and modify the existing quality of service parameters which were originally defined for the wired networks. The user specified degree of mobility, for instance, is considered to be one of the most important wireless QoS parameters where the network needs to reserve the resources according to the expected number of connection handovers based on the specified mobility figure. The handover failure and latency are also to be investigated as the QoS specifications where the parameters can be chosen based on the specific traffic types and their performance requirements. More about the QoS extension ideas are presented in Chapter 7.

2.5 Proposed Architecture and the Addressed Issues

In order to explore few of the open design issues, identified in the last section, a mobile radio network architecture for transporting ATM traffic is presented in this dissertation. This section highlights the main features and their relevant design issues for the proposed radio LAN.

2.5.1 Physical Layer and Medium Access

The proposed network uses a single radio frequency for all the base stations and that provides a homogeneous medium where the transmission cells do not share any hard boundary between them. In order to alleviate the capacity limitation problem in this single frequency scenario, a large number of small transmission cells (known as pico-cells) are deployed. The pico-cell profiles (both size and shape) within a building are highly irregular and unpredictable because of the complicated terrain and attenuation characteristics of the building materials. For a fixed placement of a BS, the corresponding transmission cell profile may even vary with time, depending on the relative location and movement of the objects within the cell. To mitigate the unreliability, caused by this unpredictable pico-cell characteristics, an overlapping cell strategy is introduced where an MU is likely to be seen simultaneously by several base stations. The cost paid for this is a reduction of the overall frequency reuse efficiency. In order to facilitate the implementation of an efficient location management service, the network is partitioned into several administrative and/or geographical domains, which are termed as *clusters*.

In the present single frequency framework, it is necessary to use an efficient medium access protocol in order to utilise the scarce radio resources as efficiently as possible. When narrow band modulation is used at the physical level, some form of TDMA-based medium access control is typical for the PCN systems, supporting isochronous CBR traffic [IS54 89]. When multiservice VBR and packet data requirements are also taken into account, static TDMA approaches must be extended to offer the provisions of dynamic resource allocation. The Packet Reservation Multiple Access (PRMA) [Nanda 91] is such a dynamic TDMA protocol which is currently being used by the WINLAB cellular packet switch [Hellstern 93]. Few other variations of dynamic TDMA have also been investigated in [Wilson 93]. The NEC wireless ATM project also intends to use such a dynamic TDMA protocol where a maximum number of slots in a TDMA frame can be assigned for CBR voice traffic. VBR and packet data messages are dynamically assigned one or more 48 byte slots in the TDMA interval following the last allocated voice slot in a frame.

For the present architecture, a random medium access approach is taken for ensuring statistical multiplexing of the channel. A Carrier Sense Multiple Access (CSMA) family protocol is investigated in Chapter 4 for evaluating its performance under the proposed network model. The random access nature of the protocol is expected to cater for the VBR traffic whose bandwidth requirements usually vary in a rapid and unpredictable manner. In order to detect the transmission errors, caused by different types of collisions and other failures, an acknowledgement packet is used for each ATM cell transmission. While consuming a part of the RF bandwidth, these acknowledgements can provide a virtually error free service to the higher layer protocols. This is particularly suitable for the lightweight AAL schemes, which rely on low error rates at the lower layer protocols. Although the scheme does not provide any reservation facility, it is shown that by adjusting the individual transmitter's back-off delay it is possible to implement a dynamic priority scheme for the participating radio entities. Such a mechanism is used for controlling the MAC layer congestion, caused due to asymmetric loading from the *base* and the *mobile* entities.

2.5.2 Backbone Reconfiguration

The backbone of the proposed LAN is made of Olivetti four port switches[6], connected by high speed point-to-point links. Since the base stations are also connected to the backbone by the point-to-point links, an extension of the wireless LAN requires nothing more than plugging more base stations to the backbone ATM switches. The modular feature of the backbone network allows it to be modified *on the fly* by adding or removing switching elements, radio base stations and the end host devices like ATM cameras [Pratt 94] and

[6] A bus-based 4x4 ATM switch, designed and developed at Olivetti Research Ltd., Cambridge.

workstations. In order to cope with these online topology changes, a dynamic reconfiguration mechanism, for connectivity information maintenance, needs to be implemented for the backbone network entities. The primary requirement for a reconfiguration scheme is identified as the optimum distribution of the network connectivity information in order to facilitate call-by-call routing in the ATM environment. To achieve this, a *Dynamic Minimum Tree Maintenance* (DMTM) protocol is proposed where the *minimum hop trees* for all the network interfaces are constructed and dynamically maintained in the network nodes.

The DMTM protocol, designed for a high speed point-to-point ATM environment, uses an incremental approach for the construction and maintenance of the minimum hop tree connectivity information. In addition to the low reconfiguration latency, the spatial locality of the proposed protocol can be exploited for implementing simultaneous connection routing and reconfiguration operations at different parts of the backbone network. The advantages of the scheme become more apparent when the network is large and a local topology change remains completely transparent to the remote parts of the network. It is also demonstrated in Chapter 3 that DMTM can cope with a standard IP type network partitioning strategy and hence it is usable for the internetworking purposes.

2.5.3 Mobile Representative

Inspired by the approaches, described in [Teraoka 93] and [Ioannidis 91], most of the mobility handling functionalities are put into the network layer so that the existing applications can run transparently without even realising the existence of mobility in the network. For each mobile unit, a software entity, known as *mobile representative* (MR), is employed for hiding the effects of user mobility from the rest of the network. Any entity, interested in setting up a connection with a mobile unit, sends a setup request to its representative and then it is the representative's responsibility to locate its client MU and send the setup request to the mobile unit. To avoid non-optimal connection routing, certain degree of mobility is assigned to a mobile representative so that it can migrate across the network clusters in order to follow its client MU. The details about the architecture and the migration policies of the representatives are discussed in Chapter 6.

Like the PAS and the ENS schemes, proposed in [Cohen 92], a permanent mobile addressing policy is used for the present architecture. The entire network address space is equally divided into a *mobile* address space and a *fixed* address space. It is shown that by using a simple two level partitioning hierarchy, along with the permanent mobile addressing policy, it is possible to design an efficient and distributed location server which is capable of managing the location information of the mobile units and their migratory software representatives. Since most of the representative management operations rely heavily on the location information of the MUs and their representatives, in order to reduce the cross management entity traffic, most of the MR management functions are integrated within the location server.

2.5.4 Mobile Connection

The adopted connection management relies on a novel *connection caching* scheme for handling the handovers. It is shown that by splitting an end-to-end mobile connection into a number of *connection segments* and dynamically creating and deleting those segments according to the end mobile movements, it is possible to keep the handover penalties within a reasonable limit. This strategy seems to be quite advantageous when compared to the schemes where an end-to-end network layer connection has to be setup during each handover (see the *full re-establishment* scheme in [Keeton 93]) procedure. While routing a mobile connection, a combination of *source* and *dynamic hop-by-hop* approaches is used for achieving an optimal performance. Each mobile connection is routed through the representatives of the end mobile units, in order to implement the representative mediated handovers. It is worth mentioning that the current connection management scheme does not distinguish between the *inter* and the *intra* cluster handovers; the only difference, seen by the applications, is the increased latency because of the larger geographical separations between the signalling entities in an inter cluster situation. However, the inter cluster location and representative migration management operations are much more complicated compared to the intra cluster ones.

2.5.5 Handover

Although the handovers are quite well understood in the conventional cellular networks, third generation PCN system design requires the conventional handover protocols to be re-investigated for handling continuous media traffic in the packet-switched networks. The conventional circuit-switched telephone networks mainly carry voice traffic which can tolerate some degree of service disruption and information loss during a call handover. In a multimedia PCN environment, on the other hand, the carried data traffic is very much sensitive to packet loss during a connection handover [DeSimone 92]. Moreover, while handling multiple connection handovers, the QoS guarantee on a continuous media stream requires the handover jitter to be within a specified limit. Note that, since a circuit-switched telephone network behaves as a *blocking* system rather than a *delay* one, most of the mentioned jitter related problems are inherently absent in the conventional cellular environment. The use of a single frequency makes the handover design more challenging in the present mobile ATM LAN architecture.

Handovers, in general, can be classified [Chia 91] as *backward* and *forward* where in a backward case, while moving, a mobile entity exchanges all the handover signalling messages with the old base station. In a forward handover, on the other hand, the mobile negotiates with the backbone through a newly chosen base station. Although a forward handover allows the establishment of a new signalling channel even after the one with the old base station is lost, this approach requires more network resources to be occupied by multiple signalling channels through the old and the new base stations. When a mobile moves with a very high velocity, it is possible that the contact with the old base station is lost before a channel is established with the new one. To continue a call, in this situation, it is unavoidable for the mobile entity to perform a forward handover. In the presented LAN architecture, the *forward* policy, along with a connection caching scheme, is adopted for low latency handover implementation. Considering the unreliability of the radio links in an in-building environment, the forward scheme is found to be more reliable, but of course, at the cost of additional wired and wireless resources.

A connection handover [Pollini 92] in the WINLAB Cellular Packet Switch is performed by the end mobile entity, old and the new base stations, and it takes place without any intervention from any other central management entity. This requires the handover signalling messages to travel from the relevant base stations to the remote end-point of the connection. While maintaining its distributed nature, the protocol suffers from the lack of salability because of this end-to-end exchange of the signalling messages. Moreover, the protocol for CPS does not cover the *both-end* handover situations where the mobile units at both the ends of a connection require simultaneous handovers to occur. This distributed approach also requires two different protocols for handling the *intra* and the *inter* MAN handovers. A family of scalable and low latency handover protocols are presented for the proposed mobile LAN architecture where a single protocol is applicable for both the *intra* and the *inter* cluster situations. It is achieved by arranging a handover to be supervised by the representative of the relevant end mobile unit. This prevents the handover signalling messages from traveling beyond the representative of the roaming mobile unit. Apart from ensuring scalability, it completely isolates the signalling messages which are generated by the handovers at both the ends of a connection.

2.6 Summary

A number of issues, related to mobile network design, have been examined in this chapter. Starting with a description of the recent evolution of the cellular mobile networks, a number of related research projects were reviewed in order to identify the design requirements for the third generation Personal Communication Network systems. The need for new medium access, location and connection technologies have been recognised in the context of newly emerging multimedia traffic in mobile environment. With an aim for integrating ATM within the PCN environment and addressing the identified design issues, a novel mobile LAN architecture was proposed in this chapter. The rest of this dissertation deals with the design of this ATM LAN, with special emphasis given to those issues.

Although recognised, this work does not address the security and the ATM adaptation layer issues for the mobile networks. While not making detailed recommendations about the mobile QoS implementation techniques, at different stages, the dissertation tries to identify the crucial design decisions which affect the QoS guarantee in a mobile network. Future work proposals for dealing with this issue in a more detailed manner is presented in Chapter 7.

Chapter 3

Dynamic Reconfiguration

This chapter deals with the design issues of a protocol which is proposed for dynamic topology reconfiguration in a high speed connection-oriented LAN, suitable as the backbone of the wireless network architecture, presented in this dissertation. A prototype version of the proposed *Dynamic Minimum Tree Maintenance* (DMTM) protocol is implemented as a user space reconfiguration server on a microkernel, known as WANDA. A distributed call-by-call routing protocol, for optimising the end-to-end delay and the network load distribution, is also proposed and implemented on top of the reconfiguration service. The chapter includes performance figures for the proposed reconfiguration and the routing protocols, obtained from a discrete event simulator.

3.1 Introduction

Most of the existing topology reconfiguration algorithms are classified into two categories: *event driven* and *time driven* [Abu-Amara 93]. In an event driven algorithm [Tajibnapis 77], a node broadcasts its topology information only when it detects a topology change. A time driven algorithm however requires all the network nodes to periodically broadcast their topology information, regardless of any topology change. The ARPANET protocol [McQuillan 80] is one of the classical examples of a time driven reconfiguration algorithm. The presented DMTM protocol, designed for a high speed point-to-point ATM environment, uses an incremental event driven approach for the construction and maintenance of the minimum tree connectivity information database within the network nodes.

The algorithm in [Schroeder 91] also deals with topology reconfiguration in a similar point-to-point network (*Autonet*) which carries datagram traffic. The original algorithm for distributed *spanning tree* computation was first presented in [Perlman 85]; Autonet uses an augmented version of it which is able to detect the termination of the spanning tree computation procedures. [Perlman 85] attempts to provide reconfiguration services for a Bridged Ethernet and generally, for a family of multiple access networks. When this protocol provides quite slow reconfiguration (of the order of seconds) facilities, both Autonet and DMTM have reconfiguration latencies of the order of milliseconds, but of course with high speed (100 Mbits/s) point-to-point links. It should be noted that unlike ARPANET, both the DMTM and the Autonet reconfiguration algorithms are designed for local area environments.

To provide the best high load performance on a simple hardware, Autonet switches support the *cut-through* mechanism where a part of a large packet can be forwarded while keeping the rest of it in a switch buffer. However as a consequence, transient forwarding loops might result in deadlock and thus cannot be tolerated. Autonet has taken an approach of rapidly calculating the entire topology whenever it changes and expunging all old forwarding tables before installing any new ones. While ensuring the immunity from deadlock and infinite looping, this scheme suffers from temporary service interruption during the reconfiguration process. This shortcoming is attempted to be removed in DMTM by adopting an incremental approach where a topology change initiates the minimum tree modification functions while not interrupting the ongoing ATM connections. As a result, a node or link failure affects only the virtual circuits which are passing through the broken entities; all the other active connections remain un-interrupted.

Another network that provides automatic topology maintenance is PARIS [Awerbuch 90]. Unlike Autonet, PARIS uses a time driven approach for maintaining topology information in a high speed connection-oriented network. Each connection is source routed, based on the complete topology information stored in each of the active nodes in the network. Although the present DMTM protocol does not maintain complete topology information, it ensures the minimum tree information storage for all the reachable network interfaces from a network node. It is shown that using these minimum tree information, with some degree of redundancy, DMTM can support call-by-call dynamic routing with a suitable combination of hop-count and load balancing performance.

3.2 Network Model

Any network entity, which is involved in the reconfiguration procedure, is known as a *node* and the nodes are connected to each other by point-to-point communication *links*. Note that, in order to support heavy local traffic, it is allowed for two *neighbour*[1] nodes to be connected by more than one link. The average number of links connected to a node is denoted as the network *connectivity*. Each link may have a cost factor associated with it for representing the link-delay, bandwidth or any other link status parameter. The present version of the reconfiguration protocol assumes equal cost factors for all the links in the network. An interface through which a node is connected to the network is referred to as a network interface (*nif*). Although the protocol can support both the per-node and the per-nif addressing schemes, a per-node addressing scenario with a unique identification for each network interface is considered throughout this chapter.

[1] Two entities, connected by a point-to-point link, are referred to as neighbours with respect to each other.

The events, causing any connectivity change, are termed as the topology change events; *Link Deletion*, *Link Addition*, *Node Deletion* and *Node Addition* are the four possibilities in the current network model. Functionally, a *Node Deletion* is equivalent to multiple *Link Deletion* events for all the links belonging to the deleted node. Similarly, a *Node Addition* is synonymous to multiple *Link Additions* for the links whose one end is the newly added node and the other end belongs to an *active node*[2].

3.3 Reconfiguration Protocol

The proposed DMTM protocol uses a distributed approach to construct, expand and prune the *minimum hop trees* for each network interface in the network. In steady state, for each nif there exists a minimum hop tree with the nif as its root. A minimum hop tree for nif_j represents the minimum hop paths from all the leaf nodes of the tree to nif_j. If nif_j is reachable from a node N_i, then the tree must span through N_i. In that case, the node N_i should have an entry in its connectivity database specifying the minimum hop outgoing link which is a branch of the existing minimum hop tree corresponding to nif_j. The thick lines in Figure 3.1 represent a minimum hop tree for nif_p of the node labelled 'a'. Any change in the topology triggers a sequence of reconfiguration operations to adjust the minimum hop trees affected by the event.

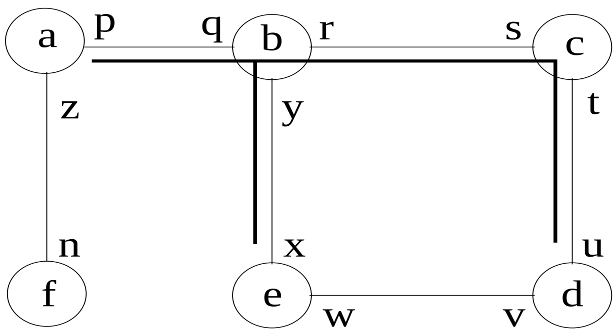


Figure 3.1: Minimum hop tree for the network interface 'p'

In order to limit the storage requirement, the protocol is extended for the networks with a suitable partitioning scheme, as presented in Section 3.4.

3.3.1 Design Objectives

The protocol is required to ensure that each network node should maintain the following connectivity information. If a node N_j is physically reachable from node N_i, then in steady

[2]A node which is willing to participate in the reconfiguration procedure.

state, the node N_i should have the minimum hop connectivity information for the node N_j. If k number of network interfaces of node N_j are reachable from node N_i then N_i should have the minimum hop connectivity information for all those k interfaces. This redundancy is used for the policy based distributed routing, as described in Section 3.6. The reconfiguration protocol also needs to meet the following performance requirements.

- High Responsiveness: Following a topology change, the network should be reconfigured as quickly as possible.
- Low Message Complexity: It is necessary to keep the amount of reconfiguration traffic as low as possible. Since the protocol is aimed mainly for a high speed network environment, low message complexity is not a critical requirement.
- Reliability: The protocol should be reliable so that it always maintains consistent connectivity information as stated above. It is also essential for the protocol to be free from loops during a reconfiguration process. Note that the current version of the DMTM protocol relies heavily on the integrity of the reconfiguration message delivery mechanism. Sequential delivery is ensured in the present connection-oriented approach where the messages are sent through the pre-established reconfiguration connections, as described in Section 3.3.2. Message delivery through the dedicated reconfiguration connections also guarantees very low message loss and error probabilities. Although a message loss can give rise to transient inconsistency of the connectivity information, DMTM is designed to prevent *looping* in such error situations. An inconsistent entry in the connectivity database disappears with subsequent reconfigurations.

3.3.2 Reconfiguration Connections and Messages

A connection-oriented approach is chosen where a node always forwards the reconfiguration messages to its neighbours through the pre-established reconfiguration connections. An active node periodically transmits a particular type of reconfiguration message, known as *setup* beacons, through all of its network interfaces. When a node N_j receives such a beacon from its neighbour node N_n, through the interface nif_k, the receiver node sets up a reconfiguration connection with N_n over the interface nif_k. Note that this happens only if the node N_j did not have a reconfiguration connection with N_n through the interface nif_k. Functionally, the *setup* beacons are used for the detection of the *Link Addition* and the *Node Addition* events where the periodicity of these beacons determines the granularity of the detection procedure. The following messages are used by the DMTM protocol during a reconfiguration procedure.

- Construct: These messages are sent periodically through all the available reconfiguration connections of an active node. If a node does not receive such a message through a reconfiguration connection for a preset amount of time, it decides that the corresponding link has been deleted. By controlling this time-out value, it is possible to adjust the grain of the deletion detection procedure. When a node receives the first *construct* message through a reconfiguration connection, the node uses it for the tree construction process, as outlined in Section 3.3.3. On the other hand, when a reconfiguration connection times out, the node creates and forwards a *prune* message in order to trim the minimum hop trees which are affected by this deletion event.

- Expand: When a new link is added, in addition to the construction of the necessary new minimum hop trees, a network requires to expand the existing trees (while maintaining their minimum hop nature) which are affected by the added link. The *expand* messages are used for these expansion procedures, as described in Section 3.3.4.

- Prune: Upon detection of a link deletion, a node creates the *prune* messages and forwards them to remove and trim the affected trees.

- Repair: After pruning, if a tree looses its minimum hop feature then it is necessary to repair it; the necessary tree fixing operations, described in Section 3.3.5, are performed with these *repair* messages.

The fields of a reconfiguration message, $Msg(src, type, node_{rem}, nif_{rem}, ver, hops, nif_{local})$, are as follows.

- src: The address of the generator node of the message; this remains unaltered during the entire lifetime of the message.

- $type$: Indicates one of the four message types.

- $node_{rem}, nif_{rem}$: The message is used to create or modify the minimum hop tree corresponding to an interface which is identified by the pair $(node_{rem}, nif_{rem})$.

- ver: Each active node maintains a version-counter which is incremented by unity after each set of *construct* message transmission takes place through all of its reconfiguration connections. The *ver* field of a *construct* message carries the existing value of the generator node version-counter. The details about the version number and its usage are described in Section 3.3.3.

- $hops$: Specifies the number of hops travelled by the message. This field is initiated to one by the message generator and is incremented by unity after each subsequent hop.

- nif_{local}: Upon reception of a reconfiguration message, the receiver node puts the incoming interface identifier into this field of the received message.

In each active node, the connectivity information is maintained in a format as depicted in Table 3.1. An $Entry(Node_{rem}, Nif_{rem}, Version, Hops, Nif_{local})$ in node N_j represents the position of N_j in the existing minimum hop tree corresponding to the network interface $(Entry.Node_{rem}, Entry.Nif_{rem})$. At any one time, the connectivity database in a node may contain at most one entry regarding a specific tree. The fields $Entry.Version$, $Entry.Hops$

and $Entry.Nif_{local}$ are used and updated by the reconfiguration protocol; note that the $Entry.Hops$ and the $Entry.Nif_{local}$ fields are also utilised by the multi-path routing protocol, described in Section 3.6.

$Node_{rem}$	Nif_{rem}	Version	Hops	Nif_{local}

Table 3.1: The format of a connectivity database entry

3.3.3 Tree Construction

After a new link addition, it is necessary to create two minimum hop trees for the network interfaces at both the ends of the newly created link. A *Node Addition* event gives rise to the similar requirements for all the links connected to the new node. A tree construction process is initiated when a node sends the first *construct* message through the present reconfiguration connection on the relevant nif. A *construct* message is uniquely identified by the $Msg.node_{rem}$, $Msg.nif_{rem}$ and the $Msg.ver$ fields. While the first two identify the root nif of the tree under construction, the third one carries a version number, stamped by the message generator.

After receiving a *construct* message, a node computes its position (in terms of hop-count from the root nif) in the corresponding tree and stores that information into the connectivity database; the version number is also stored for indicating the age of the entry. Nif_{local} field of the connectivity database stores the incoming interface identifier for the received *construct* message; this field indicates to the link belonging to the minimum hop tree which is being constructed. In case when the node connectivity database already had an entry regarding the same tree, the old information is replaced only in the following situations.

1. When the received *construct* message carries a version number which is equal to the one found in the existing entry, and the new hop-count is lesser than the existing one. The first condition indicates that the same *construct* message has re-arrived to the node through a different route.
2. When the received *construct* message carries a version number which is greater than the existing one. This implies that the previous connectivity information became outdated and it should be replaced by the newly received *construct* message information.

In such situations the $Entry.Version$, $Entry.Hops$ and the $Entry.Nif_{local}$ fields are updated for the corresponding tree entry. The mentioned checks are not necessary when the received *construct* message corresponds to a tree whose root nif belongs to a neighbour of the receiving node; at this node, the connectivity databases are updated on reception of each *construct* message from the neighbour nif.

In order to limit the message complexity and to avoid looping, the following message forwarding scheme is used. A neighbour node of the root nif forwards a *construct* message only after the first time it receives it. At any other node, a *construct* message is forwarded only if the message causes an inclusion or update of an entry in the connectivity database. While forwarding, after incrementing the hop-count field, a node transmits the message through all the available reconfiguration connections except the one through which it was received. Although this forwarding scheme reduces the chance of a message visiting a node more than once, a complete prevention of it cannot be ensured because of the stochastic nature of the link delays and the inherent loops in the network topology. When a *construct* message revisits a node, the hop-count field of the message must have a greater value than the one stored in the connectivity database of the node. In this situation the node avoids any chance of construct phase looping by ignoring the revisiting message. Note that a message is never forwarded to the node, containing the root nif of the tree of interest. Although the *construct* messages are periodically transmitted from the root network interface, a neighbour node forwards the message only for the first time when it receives it. The node however monitors the successive messages to detect a *Link Deletion* event. On detection of such an event, a network-wide tree pruning and repair procedure is triggered. Note that the current description refers to a single tree construction procedure. Since each tree construction is governed by a unique and independent *construct* message, it is possible for several tree construction processes to coexist in the network.

3.3.4 Expansion

The addition of a new link in the network may alter the minimum hop paths from various nodes to several destination network interfaces. When a node detects a new link on one of its interfaces, it starts an expansion procedure for all the minimum hop trees which, the node believes that it belongs to. For example, when the addition of $link_{pq}$, in Figure 3.1, is detected by $node_b$, it creates and sends an *expand* message through nif_q for each of the minimum hop tree entries in the connectivity database. Similar action are also to be taken by $node_a$. To avoid redundancy, no *expand* message is created by $node_b$ for the entries which correspond to the trees for the interfaces of $node_a$.

An arrived *expand* message is used much like a *construct* one, for computing the position of the receiver node in the relevant tree. If there was no corresponding entry in the connectivity database, then the node puts an entry with the appropriate version number and hop-count values. In the case when the node already had an entry for the relevant tree, the existing entry is updated only if the $Msg.hops < Entry.Hops$ condition is satisfied. These ensure that during a tree expansion process, if necessary, a node is allowed to change its position within the tree. The forwarding strategy for the *expand* messages is similar to that for the *Construct* ones. The only difference is that unlike the *construct* message case, a neighbour node of an *expand* message generator treats the message no differently than any other node.

3.3.5 Pruning and Repair

This part of the reconfiguration protocol deals with the link deletions where a set of minimum hop trees need to be pruned and repaired, as a result of a change of topology. When $Node_b$, in Figure 3.1, detects the deletion of $link_{pq}$, for each connectivity database entry with $Entry.Nif_{local} = nif_q$, the node creates a *prune* message from the corresponding database entry and forwards it through all the available reconfiguration connections. On reception of a *prune* message through nif_j, if a node finds a connectivity database entry with $Entry.Nif_{local} = nif_j$, the node deletes the corresponding entry from its database. A *prune* message is always generated at a node adjacent to a deleted link and it traverses the corresponding minimum hop tree downwards until it reaches the leaf nodes.

Although at the end of a pruning process the relevant tree becomes consistent with the changed topology, it may not remain as a minimum hop tree any longer. This requires a repair cycle after each pruning process, in order to reconvert the pruned tree as a minimum hop one. When a *prune* message from $node_l$ arrives at nif_j of $node_m$, and $node_m$ does not find a connectivity database entry with $Entry.Nif_{local} = nif_j$, the message is ignored by the receiver node. However, if $node_m$ happens to have an entry, regarding the relevant tree but with $Entry.Nif_{local}$ as a different interface nif_r then the node starts a repair cycle for the tree. Such a situation indicates that even after $node_l$ has been detached, $node_m$ still belongs to the relevant minimum hop tree through interface nif_r. As a start of the repair cycle, $node_m$ sends a *repair* message to $node_l$ through nif_j, indicating its own position in the tree. The receiver $node_l$ treats this exactly like a *construct* message in order to compute its own position in the tree. Once the necessary entries are updated, $node_l$ forwards the message through all the available reconfiguration connections except the one through which it was received.

While repairing a tree, there exists a possibility of infinite looping when a *repair* message is unceasingly chased by a *prune* one within a closed loop of a network. Such a situation for the network, shown in Figure 3.1, is described here. The pruning cycle, initiated by the deletion of $link_{pq}$, should ideally purge the complete nif_p minimum hop tree without any subsequent repair cycle. However, if a *prune* message generated by $node_b$ reaches to nif_v before nif_u then $node_d$ creates a *repair* message and transmits it through its interface nif_v. After a while, when $node_d$ receives another *prune* message through nif_u, it deletes the proper connectivity entry and forwards the message through the interface nif_v. This *prune* and the previously created *repair* messages are never discarded and they follow each other within the closed loop *bcdeb*.

One way of avoiding this looping problem is to slightly modify the forwarding policy for the *repair* messages. If the receiver node of a *repair* message is a neighbour of the message generator then the message is not forwarded any more. In Figure 3.1, once a *repair* message, generated by $node_d$, reaches to the neighbour $node_c$ through interface nif_s, the latter does not forward the message any more. Although this mitigates the looping problem, few unnecessary *repair* messages are generated and discarded before the reconfiguration cycle stabilises. These redundant message creation and deletion procedures, termed as *damped oscillations*, primarily depend on the network topology and the link delay profiles.

A formal specification of the complete DMTM topology reconfiguration protocol is presented in Appendix A.

3.4 Reconfiguration with Partitions

Without any partitioning scheme, the proposed protocol suffers from the following shortcomings.

1. The storage requirement for the connectivity database in each node grows linearly with the number of nodes in the network.
2. The computational complexity for both reconfiguration and routing operations escalate with increased search time for the connectivity database.
3. Since each minimum hop tree spans through all the physically reachable nodes, the total number of reconfiguration messages grows exponentially with the network size and connectivity. This, in turn, increases the network bandwidth requirement for the reconfiguration process.

In order to overcome these, a single level partitioning scheme is incorporated in the current network model. The DMTM protocol is also modified accordingly so that it can be used for the partitioned network reconfiguration.

3.4.1 Partition Model

As shown in Figure 3.2, a network may be partitioned into several disjoint subsets and each such subset is referred to as a *cluster*. The cluster boundaries need to be chosen in such a way that the inter cluster connectivity should remain much smaller than its intra cluster counterpart. A simple addressing scheme is used for the network where a node address $x.y$ represents that the node belongs to the cluster x, and y is the unique network layer identity of the node in that cluster.

Any node, connected to an *inter cluster* link, is called as a *border* node; the others are termed as the *internal* nodes.

3.4.2 Modified Tree Specifications

In a partitioned environment, a minimum hop tree for an internal node interface does not span out of the local cluster; this prevents the connectivity information about an internal

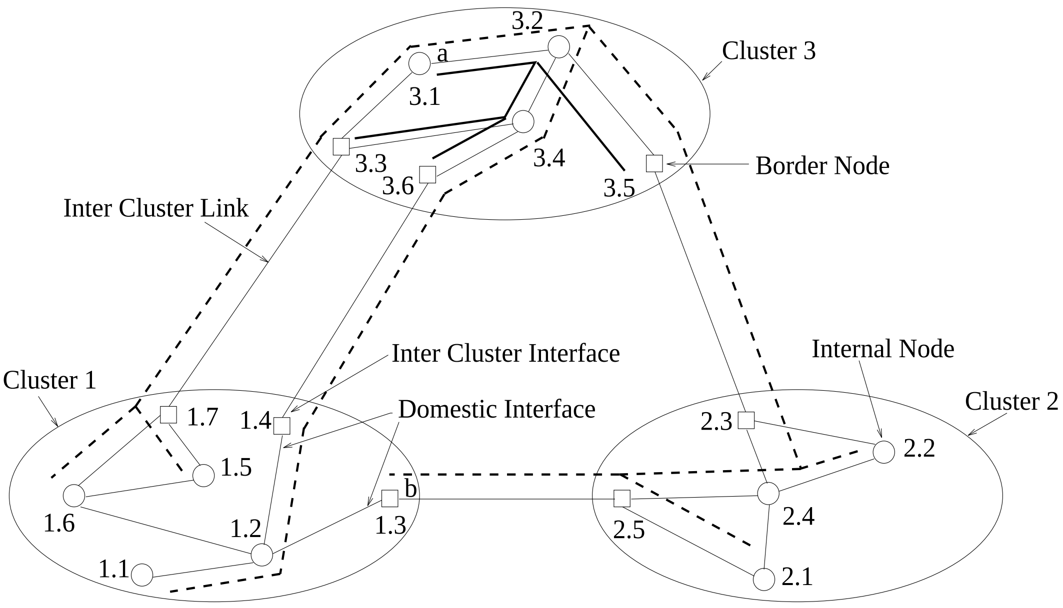


Figure 3.2: Minimum hop trees in a partitioned network

node from going out of the cluster. This restriction however is not applied to the minimum hop trees corresponding to the border node network interfaces. Both types of the trees are illustrated in Figure 3.2. While the thick lines show a cluster wide minimum hop tree for nif_a of the internal node 3.1, the dotted ones indicate a network-wide minimum hop tree for nif_b of the border node 1.3.

These modified tree specifications ensure that a node maintains the connectivity information about -

1. all the nodes in the local cluster and
2. only the border nodes of the remote clusters.

3.4.3 Modified Tree Maintenance Procedure

In order to maintain the trees with the modified specifications, the DMTM protocol needs to be changed only in terms of its forwarding policies. The basic requirement is to restrict the inter cluster transit of a reconfiguration message, depending on the root network interface of the corresponding minimum hop tree. After executing the basic DMTM protocol, if $node_f$ decides to forward a message (corresponding to the tree with root interface nif_r) through nif_t then it performs the following additional checks before actual forwarding of the message.

1. If $node_f$ is internal.

2. If $node_f$ is a border one and nif_r belongs to a border node.

3. If $node_f$ is a border one, nif_r belongs to an internal node and nif_t is a domestic network interface.

If at least one of these is true then the message is forwarded; otherwise not. The only exception about case 3 is that the message should be forwarded only through the available domestic network interfaces.

3.5 Reconfiguration with Hosts

Until this point, the DMTM reconfiguration protocol is described for the networks which consist of multiple router nodes, each with multiple network interfaces. To be able to function as a router, each node needs to maintain the required connectivity information, as described in Section 3.3.2. In addition to the router nodes, a network also contains the end *host* entities, each of which is connected to the network through a single network interface (see Figure 3.3). Although a host does not maintain any connectivity information, in order to route the connections and the packets, a router node is required to keep the connectivity information regarding the host entities.

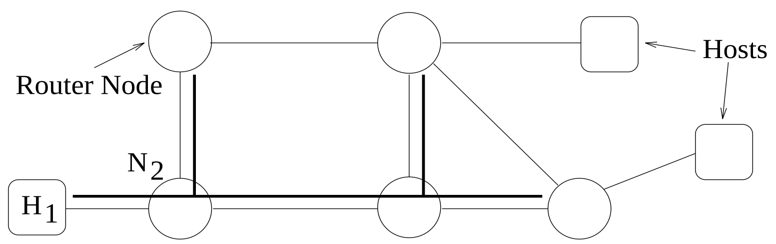


Figure 3.3: Network with single interface hosts

The following modifications of the DMTM protocol are necessary for incorporating the host entities into a network reconfiguration procedure. A host does nothing more than maintaining a reconfiguration connection with its only neighbour router node and sending *construct* messages periodically through that connection. These *construct* messages are used by the network for maintaining a minimum hop tree for the relevant host interface. Such a tree, for the host entity H_1, is indicated by the thick lines in Figure 3.3.

Since a host is connected to the network through a single link, at any one time a router node may contain at most one connectivity entry, regarding the host. However, in order to perform the policy based multipath routing, described in Section 3.6, it is desirable for the router nodes to have multiple routing information regarding an end host entity. To achieve this, a new field $Msg.node_{neighbour}$ is added to the reconfiguration message format. Before sending a *construct* message a host puts the address of its only neighbour router node into this new field. This is how all the router nodes, on the minimum hop tree of a host entity, come to know about the neighbour router node of the host. To keep this information, an additional field $Entry.Node_{neighbour}$ is added to the connectivity database entry format. While routing a connection or a packet to a host entity, an intermediate router node uses the redundant connectivity information regarding the neighbour node of the destination host. More about the routing is presented in the next section.

3.6 Routing

A distributed call-by-call routing policy, developed for a connection-oriented networking environment, is presented in this section. The protocol uses the connectivity information, created and dynamically maintained by the described DMTM reconfiguration protocol. The presented protocol aims to satisfy the following performance requirements:

- Reachability: If the source and the destination entities are physically connected to each other, a connection routing between them should not fail on the ground of unreachability.
- Deadlock Free: When unavoidable, a call routing process should fail explicitly without causing any looping or deadlock.
- Minimum Hop-count: If specified, the protocol must be able to ensure minimum hop routing.
- Load Balancing: When required, it is necessary to provide the specified degree of load balancing in the network.
- Optimal Performance: Combining the previous two features, it should be possible to achieve an optimal routing, when necessary.

3.6.1 Protocol

After a call-setup packet is generated and forwarded by the source, $node_{src}$, each intermediate node has to take a routing decision to choose an appropriate outgoing link in order to forward the packet towards the destination, $node_{dest}$. Since the scheme is distributed, the source and the intermediate nodes decide only about the next outgoing link, depending on the routing requirements and the current network status.

In the absence of any partitioning, if $node_r$ can physically reach to k number of network interfaces of $node_{dest}$, then the former should have k entries in its connectivity database, where each entry indicates to a minimum hop tree corresponding to one of the network interfaces of $node_{dest}$. The Nif_{local} field of a connectivity database entry points to the link which should be used to reach to the root nif of the corresponding tree. While routing a call, the following condition requires to be satisfied in order to avoid any looping situation.

Loop Avoidance Condition: When a call-setup packet arrives at $node_r$ along the tree T_{in}, the node can choose an outgoing tree T_{out}, only if the position of $node_r$ in the tree T_{out} is not worse than its position in the tree T_{in}. This implies that in each successive hop a call-setup packet travels closer to the destination node and therefore the possibility of any looping is avoided.

Since for a source node there exists no incoming tree, it can choose any one of the available outgoing trees for routing a call-setup packet. An intermediate node however chooses one of the outgoing trees, which satisfy the *Loop Avoidance Condition*; if more than one satisfies it then one is chosen based on some other independent routing heuristics. In a load balancing scenario, for instance, the load on a outgoing link can be used as the selection heuristics when multiple trees satisfy the *Loop Avoidance Condition*.

Basic Routing Protocol

When $node_r$ receives a call-setup packet through the incoming tree T_{in}, it executes the following algorithm for choosing an outgoing link for the setup packet.

for (each outgoing tree T_i, where T_i satisfies the *Loop Avoidance Condition*){

/* $1 \leq i \leq k$, where k is the number of available outgoing trees */

compute the route deciding factor –

$d_i = W * L_i + (1 - W) * H_i$;

/* H_i = The hop-count to the root nif for the tree T_i */

/* L_i = The current load on the outgoing link for the tree T_i */

}

The outgoing link, corresponding to the tree with the minimum d_i value, is chosen for the call forwarding. The control factor W determines the degree of load balancing during a call routing operation. When a unity W causes the routing to be solely based on load balancing, a null value of it can be used for achieving minimum hop routing.

Another appropriate heuristics for the routing protocol would be to consider the *total load*[3] on the next neighbour node instead of considering only the load on the outgoing link. In that case, the parameter L_i should indicate to the total load on the next neighbour node corresponding to the outgoing tree T_i. The effects of this modification on the routing performance are illustrated in Section 3.8.4.

[3] Sum of the load on all the links of a node.

3.6.2 Routing with Partitions

The following augmentations are necessary for adapting the routing protocol in a partitioned network environment. As indicated in Section 3.4.2, a minimum hop tree for a network interface of an internal node spans only through the local cluster whereas the trees corresponding to a border node network interfaces span through the entire network. This implies that if a connection destination entity is an internal node of a remote cluster, while forwarding the connection an intermediate node may not have a connectivity entry for the destination node, even though the destination is physically reachable from that node. Since the intermediate node is expected to contain the connectivity information regarding the border nodes of the destination cluster, it is possible to route the call by using the following two-step routing procedure.

- Step 1: Until the destination cluster is reached, execute the *Basic Routing Protocol* with the destination cluster as the target.
- Step 2: Once the call-setup packet is in the destination cluster, execute the *Basic Routing Protocol* with the destination node as the target.

A situation is considered where an intermediate entity $node_r$ has received a call-setup packet along the tree T_{in} and there exists k number of tree options for $node_r$ to reach the destination cluster $cluster_{dest}$. This k includes the minimum hop trees for the network interfaces of all the border nodes of $cluster_{dest}$ which are physically reachable from $node_r$. All these k trees are to be considered while performing the *Step 1* of the routing operation. Compliance with the *Loop Avoidance Condition* ensures that in each successive hop of *Step 1*, the call-setup packet travels closer to the destination cluster. The *Step 2* is exactly similar to the without partition case.

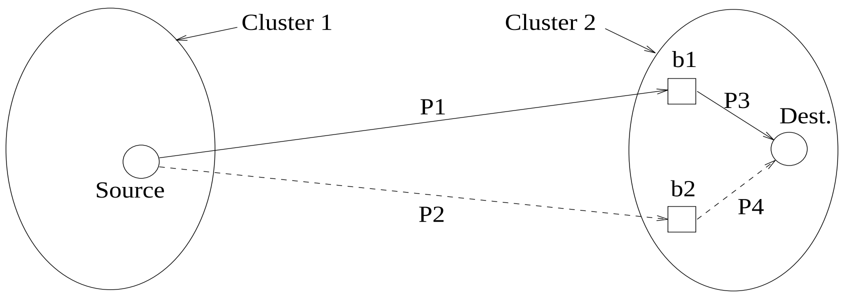


Figure 3.4: Inter cluster routing

It is important to note that during an inter cluster call setup the augmented protocol does not always ensure the minimum hop routing even when the control factor W is set to zero. Consider the case in Figure 3.4, where two end-to-end routes, P_1-P_3 and P_2-P_4, exist between the source and the destination entities. When P_1 represents the minimum hop route from the source to the border node b_1, P_3 indicates the minimum hop route from b_1 to the destination node. Similarly, P_2 and P_4 illustrate the minimum hop routes from the source to b_2 and from b_2 to the destination entity, respectively. When the control parameter $W = 0$, the *Step 1* of the augmented routing protocol will choose the shortest one from P_1 and P_2. Since at this stage of routing an intermediate node does not consider about the later part of the route (i.e. about P_3 and P_4), if the conditions $hops(P_1) + hops(P_3) > hops(P_2) + hops(P_4)$ and $hops(P_1) < hops(P_2)$ are true then a violation of the minimum hop routing is bound to occur. However, if the two clusters are connected only through a single inter cluster link then the minimum hop inter cluster routing is ensured for the case when $W = 0$.

3.6.3 Routing with Hosts

If the destination of a connection is a single interface host, then the corresponding call-setup packet is usually forwarded towards the only neighbour router node of the destination host entity. Once the packet arrives at the neighbour router node, it delivers the setup packet to the destination host entity. Unlike the only entry about the destination host entity, an intermediate node maintains multiple tree information regarding the neighbour router node. This multiplicity of the connectivity information is used for providing load balancing in a situation where the connection destination is a single interface host. The calls originated in a host are forwarded to its only neighbour router node without any explicit routing operation. The usual protocol is executed for routing the rest of the connection.

Unlike the routing policy mentioned in [Schroeder 91], the current protocol supports an un-interruptive service where the routing and the reconfiguration operations can be carried out simultaneously in the different parts of a network. The high responsiveness and the spatial locality of the DMTM reconfiguration protocol allow this simultaneous routing and reconfiguration operations to take place.

3.7 A Prototype Implementation

This section describes the implementation details of the proposed reconfiguration and the routing protocols in an ATM local area network environment. The Olivetti four port ATM switches[4], connected by 100 Mbits/s full-duplex point-to point-links, have provided an ideal platform for testing the proposed protocols. A locally developed micro-kernel, known as WANDA [Dixon 91], has been used as the generic software platform for the present implementation. A single address space (*sas*) [Sreenan 93] version of WANDA was ported by the author on the Olivetti Switches. An Inter Process Communication (IPC) mechanism, derived from the Berkeley UNIX socket abstraction, is provided by the WANDA kernel. It currently supports the Multi Service Network Layer (MSNL) [McAuley 90] protocol, adapted for the point-to-point ATM networks. The Fairisle Data Link (FDL) which

[4] A bus-based 4x4 ATM switch, designed and developed at Olivetti Research Limited, Cambridge.

is one of the available link layer protocols supported by the MSDL (Multi Service Data-link Layer), is chosen for the point-to-point links of the Olivetti switches. A detailed description of the Olivetti Switch, WANDA kernel and the MSNL protocol can be found in Chapter 8.

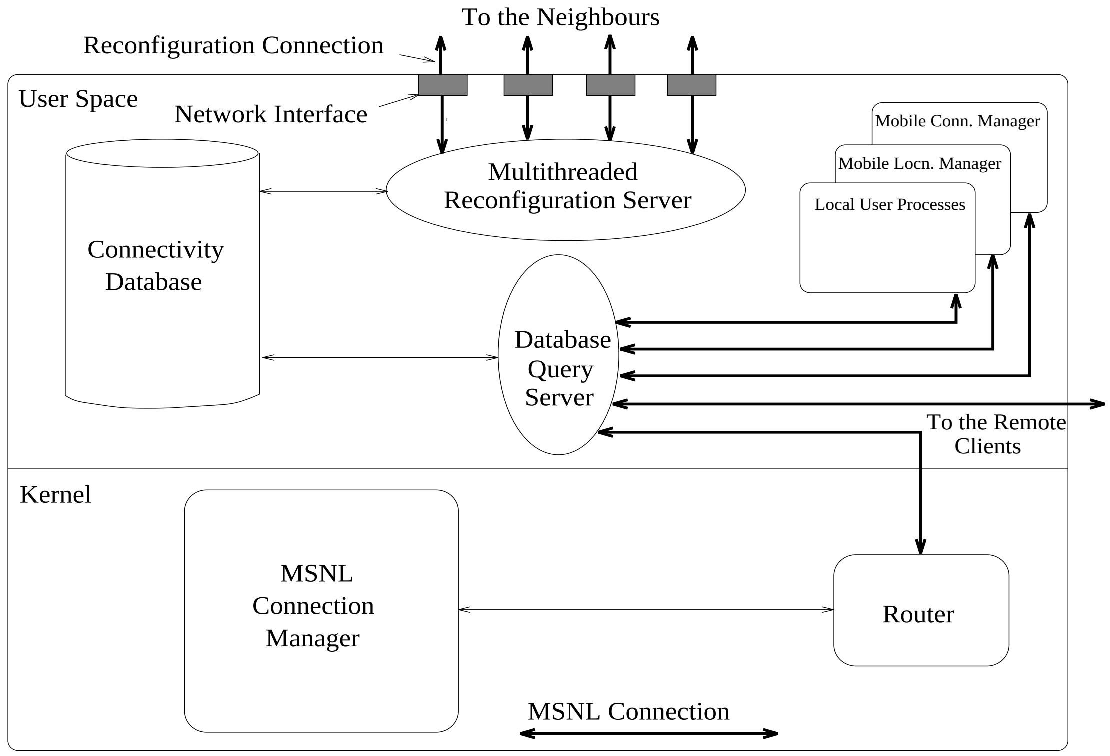


Figure 3.5: Architecture of the reconfiguration server

3.7.1 Reconfiguration Server

In order to maintain simplicity and portability, the reconfiguration functionalities are implemented as a user space daemon, known as a *Reconfiguration Server*. The components of the multi-threaded server are pictured in Figure 3.5. A dedicated thread is used as the *database query server*, which acts as an interface between the reconfiguration server and its client entities.

The database query server offers a connectivity information service on a predefined *service access point* (SAP) [McAuley 90], which is specified by the MSNL-address of the machine and a globally known MSNL-port. Any client application, interested in connectivity information, sets up an MSNL connection with the mentioned MSNL-SAP. Once connected, the query server forks a new thread for handling the queries on that connection. Note that the kernel resident MSNL connection manager is one of the active clients of the reconfiguration and its query servers; the manager needs to access the relevant connectivity information while routing each outgoing connection.

3.7.2 Reconfiguration Connection Setup

The current version of the MSNL protocol uses *metasignalling* [Black 93] datagrams for the virtual circuit control operations. The metasignalling packets are sent on the separate control circuits where the per-network-interface control circuits are permanently configured rather than established using the circuit control procedures. A control circuit operates in MSNL *promiscuous* mode [McAuley 90] where it can be used by multiple senders and receivers simultaneously. The detailed format of an FDL *metasignalling* packet can be found in [Black 93]. Only the relevant fields of a *setup metasignalling* beacon (see Section 3.3.2) are shown in Table 3.2.

Octet	Field	Description
3	MSCommand	MSCMD-SETUP
12-15	OrigAddr	Machine MSNL Address
16-19	OrigPort	Machine MSNL Cluster Mask
28-31 32	Argument	Reconfiguration Connection Port Network Interface Id.

Table 3.2: The reconfiguration related fields of a setup beacon

Soon after booting, each machine starts *listening* on a well-known MSNL port for the incoming reconfiguration connections. It also starts transmitting the periodic[5] *setup* beacons over all the available network interfaces. On reception of a *setup* beacon from M_s, the machine M_r tries to setup a reconfiguration connection with the former. At this stage, since the connectivity database of M_r does not contain any information regarding M_s, the connection manager would fail to route any connection to the neighbour node M_s. In order to avoid this problem, it is arranged to bypass the standard routing procedure while setting up a reconfiguration connection; such a connection is simply routed through the interface on which the corresponding *setup* beacon was received.

Although two neighbour machines are allowed to maintain multiple reconfiguration connections over different network interfaces, no more than one such connection may exist simultaneously through the same pair of interfaces. To ensure this, the following verification protocol is used for deleting the unwanted reconfiguration connections.

Verification Protocol:

When a machine M_r receives a *setup* beacon from M_s and there exists no reconfiguration connection with M_s through the same interface, the receiver entity M_r sends a connection request message to the machine M_s. On reception of the connection request from M_r, the neighbour machine M_s checks for any existing reconfiguration connection with M_r on the receiving interface. If no such connection is present, then M_r accepts the connection

[5] A period of 700 ms is chosen for the current implementation.

and marks it as *unverified* [Earnshaw 93]. Although these checks reduce the possibility of duplication, multiple unverified connections can still be established over the same pair of interfaces, because of the simultaneous transmission of *setup* beacons by both the neighbours. These possibilities are further reduced by randomising the *setup* beacon transmission periods. After an unverified connection establishment, the host machine waits for a while (5 ms for the current implementation) before it checks for any duplication of reconfiguration connection on the relevant interface. In case of any duplication, all the equivalent unverified connections are deleted; otherwise the only connection is marked as *verified* and the reconfiguration server starts using the connection.

The DMTM reconfiguration protocol requires each machine to maintain a version number counter and to transmit the *construct* messages with monotonically increasing version number stamps on them. Using an n-bit counter (set to zero at boot time), incremented every time a set of *construct* messages are sent, leads to the following problems.

- If a machine remains up for sufficiently long time, the reconfiguration procedure may get upset because of the wrapping of the version counter after 2^n-1 sets of *construct* messages are sent.
- When a machine boots very quickly after a crash, it may start sending *construct* messages even before the previous connectivity entries regarding the machine disappears from the network. Since the new messages start with a smaller version number, they will be ignored by the network and as a result inconsistent information will be disseminated throughout.

In the current implementation, the *system time* is used as the version number. Since the system time is downloaded from the boot-server at boot time, it always increases monotonically and that solves the mentioned problems. If a machine however is booted every time from different boot servers, there is a chance of non-monotonicity of the local time, seen by the booted machine. Assuming that the boot-servers are time synchronised with an accuracy figure (of the order of milliseconds) much lesser than the booting time (usually of the order of seconds), the chance of non-monotonicity can safely be ignored.

3.7.3 Frequent Link Failures

A hardware fault may cause frequent link failures followed by recoveries, generating a large amount of reconfiguration traffic. In order to limit this reconfiguration bandwidth consumption, it is needed to control the number of reconfiguration cycles, triggered by such faulty links. Since the link failures give rise to traffic disruptions, the reconfiguration protocol needs to respond quickly to such deletion events. However, a reconfiguration caused by a link recovery can be postponed until the link acquires a reasonably good performance history. To capture such history, a counter is maintained for each network interface, for keeping track of the number of failures over last T_w period of time. An incoming *setup* message is serviced only if the current history-count, for the relevant incoming interface, is smaller than a critical value N_c. This history mechanism can be used for controlling the number of reconfiguration cycles caused by rapid and faulty link failures. For a high T_w/N_c ratio, the scheme provides enhanced reliability, but of course at an expense of the degraded responsiveness to the link recovery events.

3.8 Simulation

A discrete event simulator is developed for evaluating the performance of the presented reconfiguration and routing protocols.

3.8.1 Network Model

All the simulation experiments are carried out for a point-to-point network model where the network is specified by the total number of nodes and the average number of links connected to each node; these two parameters are termed as the *network size* and the *node connectivity.* While creating the networks, an asymmetric approach is used where the specified average node connectivity is maintained instead of assigning equal connectivity to all the network nodes. For the prototype implementation, described in Section 3.7, the average DMTM protocol processing time[6] was found to be approximately 5 ms. Similarly, the networking code in WANDA, running on the Olivetti Switches, offers an average combined data-link and network layer protocol processing latency of approximately 500 μs. These two Poisson distributed delay figures are used for all the simulation experiments described in the rest of this chapter. A fixed bidirectional link capacity of 100 Mbits/s is used for all the point-to-point links in the network. Neighbourhood of any two nodes is defined only in terms of connectivity between them; no geographical information is used while modelling the networks. Note that no partitioning schemes are incorporated in the current simulation model.

3.8.2 Simulation Steps

Each simulated reconfiguration cycle consists of the following steps.

1. Construct a network with specified size and node connectivity.

2. Start the reconfiguration procedure and wait until it settles down over the entire network. Apply any standard graph searching algorithm in order to check the correctness of the contents of the connectivity database for each node in the network. This check validates the reliability of the tree construction procedure, outlined in Section 3.3.3.

[6] The time required by a node to execute the DMTM protocol on reception of a reconfiguration message.

3. Delete a specified number of randomly chosen links or nodes from the network and allow the reconfiguration process to continue. After it settles down, compare the modified connectivity information with the ones computed independently using the graph searching algorithm. The pruning and the repair parts of the reconfiguration algorithm are verified at this step.

4. Add the links or the nodes which were deleted in the previous step and allow the reconfiguration process to continue. When the protocol terminates, apply the graph searching algorithm once more for checking the correctness of the connectivity database entries. This step validates the expansion component of the DMTM protocol.

3.8.3 Reconfiguration Performance

For each set of the presented results, 5000 networks were generated using the model, outlined in Section 3.8.1 and for each generated network, the described reconfiguration cycle was carried out for 1000 times. The corresponding mean values of the measured statistics are presented in the following performance graphs. Various message complexities and latencies are measured after each step of a reconfiguration cycle. Note that a reconfiguration latency is measured from the time instant when a topology change event is detected by the relevant network entity.

Tree Construction Traffic

The first graph in Figure 3.6 shows the message complexity of DMTM protocol while setting up the minimum hop trees in a network, consisting of 10 nodes. For a network with N nodes and an average node connectivity of C, the worst case message complexity is $NC(NC - 1)$ where each *construct* message is expected to visit all the network interfaces at most once (the Basic Flooding Scheme [Arunkumar 92]). For the best possible case, where a *construct* message visits each link at most once, the message complexity is $(NC)^2/2$. The performance improvement of DMTM over the Basic Flooding scheme is illustrated in the second graph of Figure 3.6. The percentage reduction of the construction traffic is defined as $((M_{worst} - M_{DMTM})/M_{worst})100$, where M represents the message-count. In terms of message complexity, DMTM is found to perform better for the sparsely connected networks.

Pruning and Repair Traffic

The complexity of the reconfiguration messages, generated by the link and the node deletion events, are depicted in Figure 3.7. In the events of the multiple simultaneous link deletions (the first graph), initially the pruning and the repair traffic increases with the node connectivity. The message complexity however shows a tendency of connectivity independence when the network is too strongly connected. It is also to be noticed that for a sparsely connected network (e.g. node connectivity = 2, 3), the simultaneous deletions of a greater number of links do not generate significantly larger amount of pruning and repair traffic; the difference however is quite noticeable for the strongly connected ones.

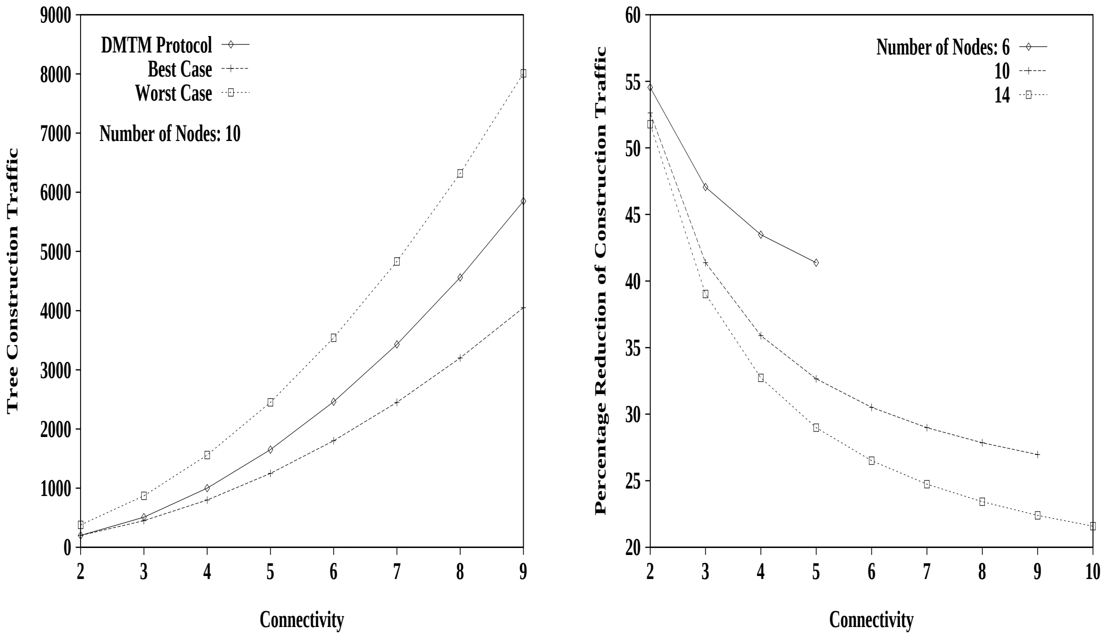


Figure 3.6: Construct message complexity of the DMTM protocol

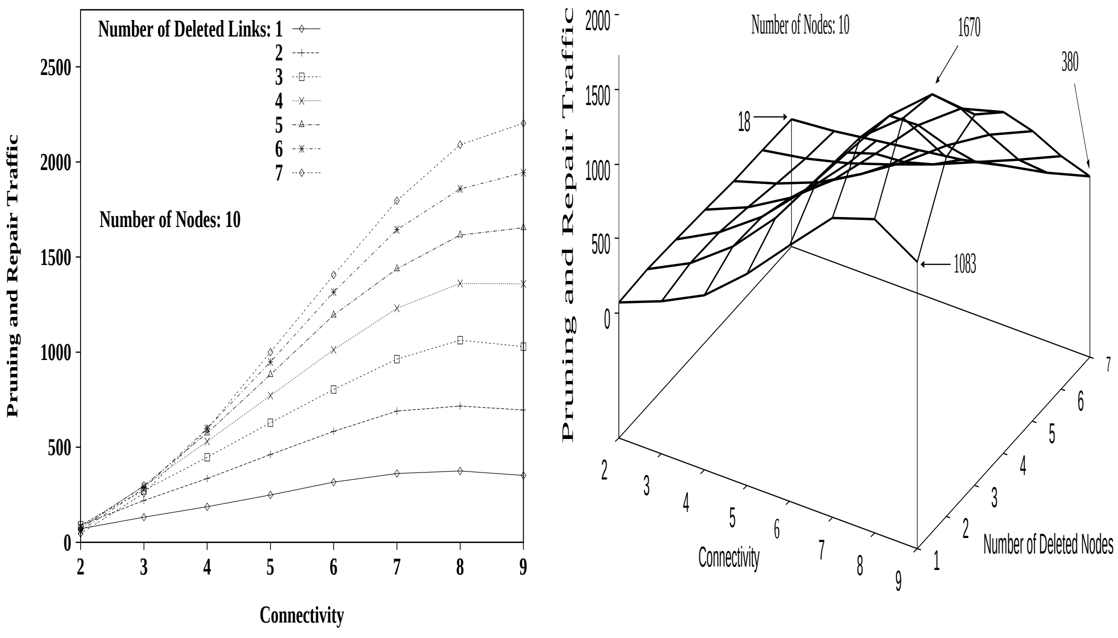


Figure 3.7: Pruning and repair traffic generated by the deletion events

In case of a node deletion (see the second graph of Figure 3.7), the combined pruning and repair traffic count shows an upward trend for the low node connectivities, and then again decreases when the network is almost fully connected. A large connectivity figure implies that more number of surviving nodes join the tree pruning and the repair procedure after a node is deleted and that accounts for the initial increment of the combined pruning and repair message complexity. In a nearly fully connected situation however very few nodes join the pruning and repair cycle, since most of them already had a dedicated link with a deleted node. This explains the reduction of message-count when the network is too strongly connected.

It is interesting to note that for a sparsely connected network, the number of simultaneously deleted nodes does not affect the generated pruning and repair traffic significantly. For the networks with larger connectivities (more than 2), the message complexity increases initially with the number of deleted nodes, and then it decreases again when more than 3 nodes are deleted simultaneously. When the number of simultaneously deleted nodes is lesser than 3, the deletion of a greater number of nodes causes the surviving nodes to take part in the pruning and repair process more intensely. On the other hand, if too many nodes are deleted then not many nodes survive to take part into the pruning and repair procedure and that is why the message complexity reduces.

Expansion Traffic

The complexities of the tree expansion traffic, generated during the last step of a reconfiguration cycle, are shown in Figure 3.8. Although the expansion traffic complexity increases linearly with the number of newly added links (first graph), the message-count shows an exponential growth with the increasing number of added nodes (second graph). Functionally, during the large number of simultaneous node additions, an expansion cycle behaves almost like a new network-wide tree construction procedure. The exponential nature of the message complexity confirms this, when compared to the construct message-count, shown in Figure 3.6.

Reconfiguration Responsiveness

The time requirements for the different parts of a DMTM reconfiguration cycle are presented in the first graph of Figure 3.9. With reducing network diameters[7], a *construct* message has to travel a lesser number of hops before it is discarded by the network. In the event of a link addition, a similar phenomenon happens during the corresponding expansion procedure. These explain the reduction of the construction and the expansion latency with increasing node connectivity. The combined pruning and repair delay, however, initially increases with node connectivity and then again decreases in the very high connectivity situations. A greater amount of repair activity in a low connectivity network, are found to be the reasons for the upward trend of the pruning and the repair delay. On the other hand, in a nearly fully connected network, the size of the minimum hop trees are usually very small; low

[7]The largest possible hop-count between any two nodes. For a network with N nodes and an average connectivity of C, the diameter can be expressed as $\lceil (N-1)/C \rceil$.

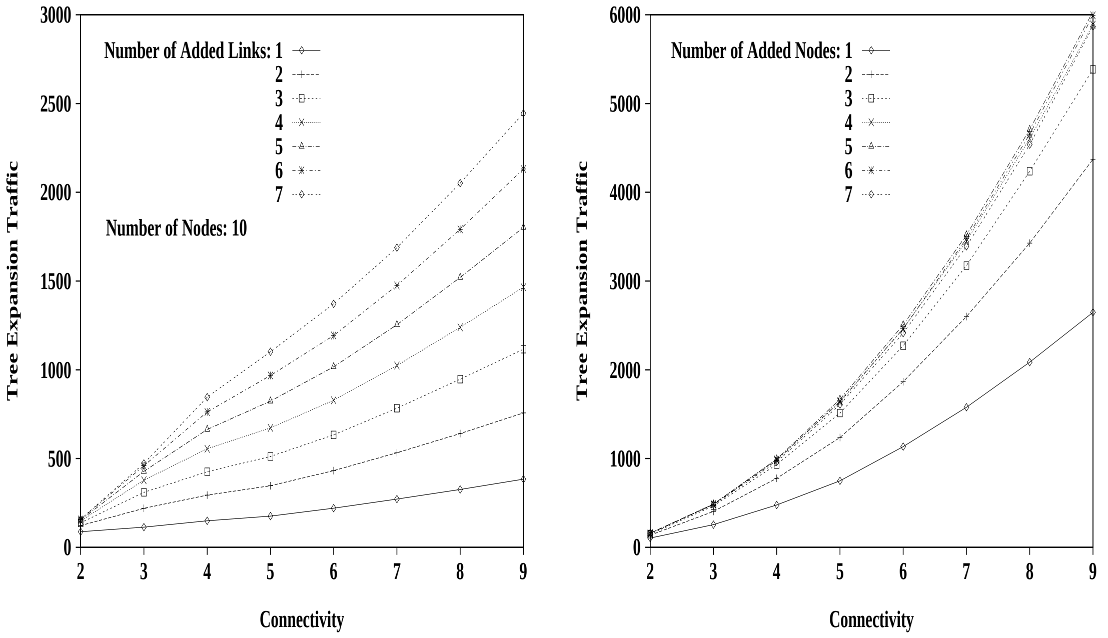


Figure 3.8: Message count for the tree expansion traffic

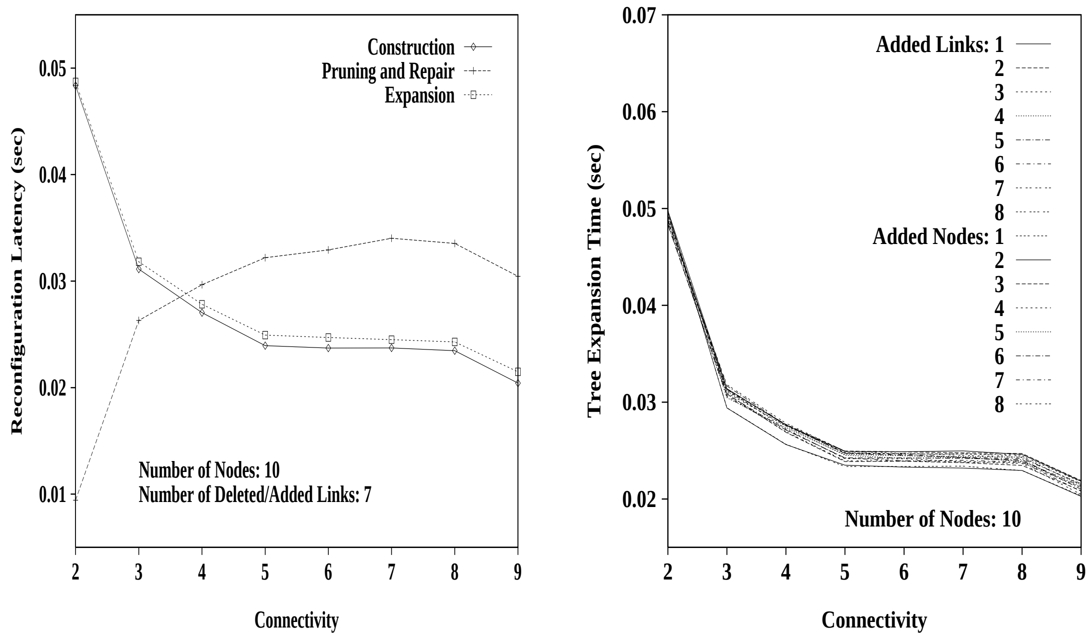


Figure 3.9: Responsiveness of reconfiguration

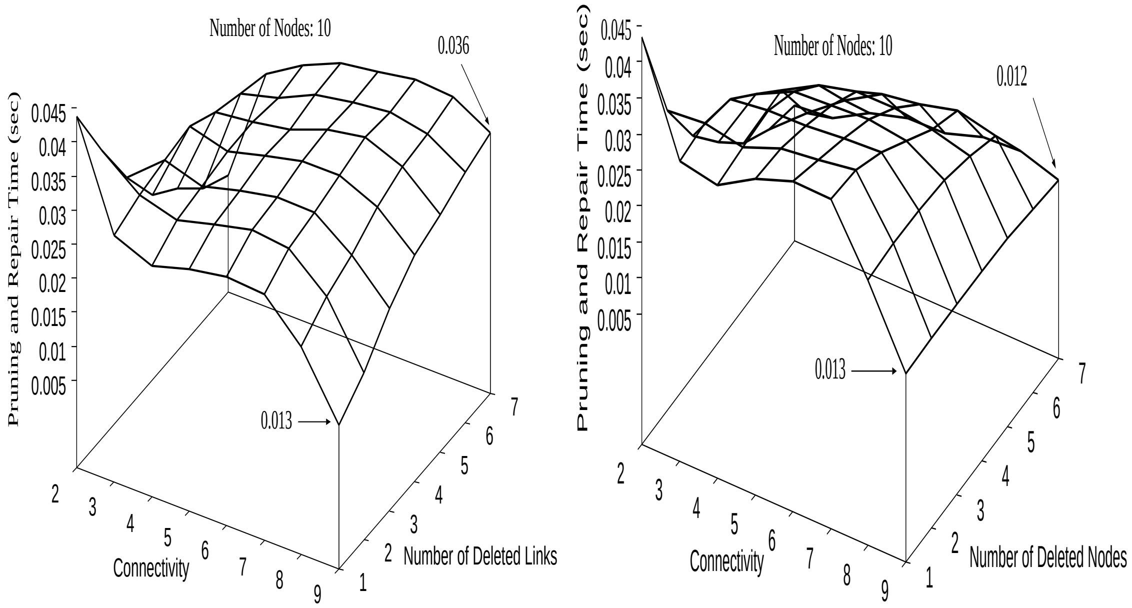


Figure 3.10: Pruning and repair responsiveness

pruning and repair activities for these small trees account for the downwards slope of the pruning and repair responsiveness. The second graph of Figure 3.9 indicates that the tree expansion delay is not very sensitive to the number of added links and nodes; the latency variation with node connectivity however is found to be quite significant.

The next two graphs in Figure 3.10 illustrate the combined pruning and repair latencies for different degrees of topology disruptions. In the first graph, for a network with a loop topology (connectivity = 2), the pruning and repair delay reduces with larger number of deleted links. On the other hand, for the highly connected networks, the pruning and the repair takes more time with increasing number of simultaneously deleted links. The graph also indicates that the latency has an upward trend for the low node connectivities, and it decreases later on for the connectivities greater than approximately 6. The latency variation patterns for the node deletions (shown in the second graph) are very much similar to the link deletion case.

3.8.4 Routing Performance

The rest of this chapter deals with the performance of the routing protocol, presented in Section 3.6. Once a network is generated (see the model in Section 3.8.1), a specified number of source and sink nodes are randomly chosen to load the network. To maintain symmetry, for each simulation run the number of sink nodes is arranged to be equal to the number of source nodes. Load from a source is controlled by adjusting the inter connection origination time, the connection datarate and the connection duration. While loading a network it is ensured that the links are never fully saturated. Each simulation is carried out for a sufficiently long time till the desired accuracy figures are achieved.

The presented routing protocol is designed for optimising the per-connection hop-count and the load balancing in a network. All the presented hop-count figures are normalised by the minimum possible hop-counts between the source and the sink nodes of the corresponding connections. If l_i is the load (aggregate datarate) on link i and there are L links in a network then the load variance can be expressed as -

$$s^2 = \frac{\sum_{i=1}^{L}(l_i - \hat{l})^2}{L}, \tag{3.1}$$

where $\hat{l}$ represents the average link load in the network. Load balancing performance of the routing protocol is expressed by a *load balancing index* which is defined as -

$$I = \frac{s^2}{(\hat{l})^2}, \tag{3.2}$$

where the smaller I values represent better load balancing performance of the routing protocol. Normalisation by the square of the mean is necessary to compare the load balancing performance in different loading conditions.

Effects of the Routing Heuristics

The routing performance is measured for the two different routing heuristics, described in Section 3.6.1. For both the graphs in Figure 3.11, the network is loaded from six randomly chosen source nodes and the same loading conditions are maintained for both the heuristics. As expected, the load balancing performance for the *Neighbour Node Load* heuristics routing is significantly better than the *Outgoing Link Load* heuristics. While routing a connection, by considering the total load on the neighbour entities, a node uses relatively more global network state information compared to the situation when it checks only its own link loads. This *look ahead* feature of the *Neighbour Node Load* heuristics accounts for its better load balancing performance. This improved load balancing performance however is achieved at an expense of the large end-to-end hop-counts of the connections, routed with the *Neighbour Node Load* heuristics. It is found that for the large values (greater than 0.3), the *load balancing factor* of the routing protocol (represented by W in Section 3.6.1) does not have significant effects on the routing performance.

Varying Loading Conditions

The first graph in Figure 3.12 shows the load balancing performance with varying network loading conditions. With a mean connection duration of 180 seconds, the inter connection origination time is varied from 50 seconds to 300 seconds and the corresponding load

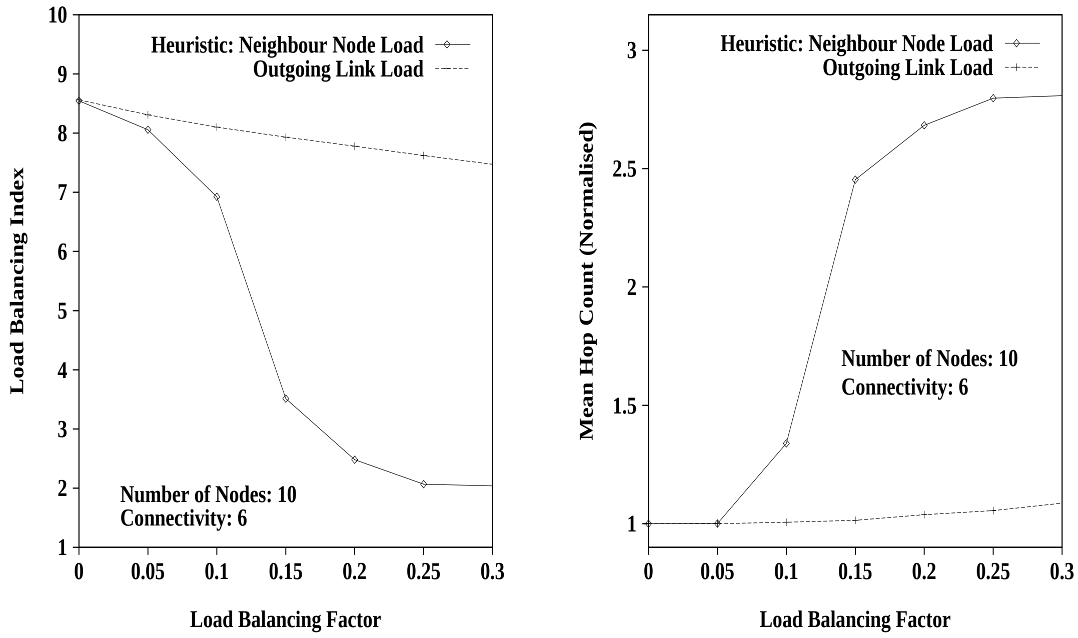


Figure 3.11: Effects of the heuristics on routing performance

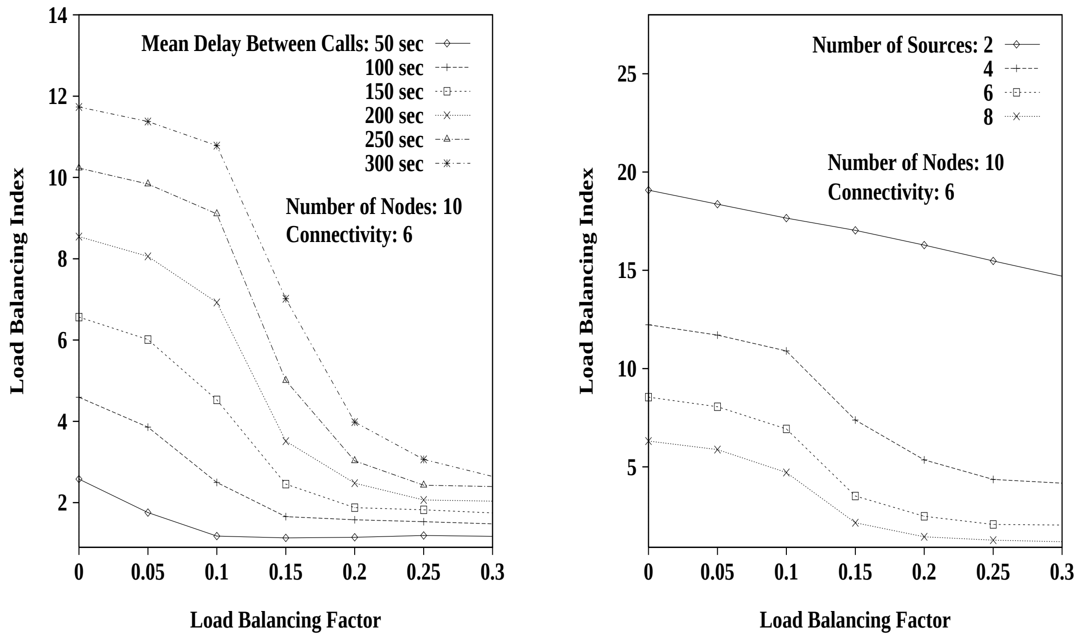


Figure 3.12: Load balancing performance with varying loading condition

balancing is measured. For the smaller values of the balancing factor, the routing algorithm performs better in high source loading situations; however for the large values of the factor (near 0.3), load balancing performance becomes almost insensitive to the inter call origination delay.

The second graph represents a situation where a network is loaded by varying number of source entities. With the increasing number of sources more network links become loaded which, in turn, improves the load balancing performance of the routing algorithm. It is to be noted that in both the graphs, the routing performance improves in the high network loading situations. The *Neighbour Node Load* routing heuristics is chosen for the results presented in Figure 3.12.

3.9 Summary

The subjects of this chapter were dynamic topology reconfiguration and call-by-call routing in a high speed connection-oriented LAN enviornment. A distributed reconfiguration algorithm has been presented where each node maintains the minimum hop tree connectivity information corresponding to all the physically reachable network interfaces. A call-by-call routing algorithm, developed on top of the topology reconfiguration protocol, was also proposed with multiple heuristics for optimising the end-to-end hop-count and network load distribution.

The design of a reconfiguration server was presented with the necessary implementation details. The performance of the proposed reconfiguration and the routing protocols have been evaluated using a simulation model. The simulator was also used for verifying the reliability and correctness of the reconfiguration protocol. In the presented system environment, the maximum reconfiguration latency was found to be approximately 70 ms for a network with 14 nodes, connected by 100 Mbits/s point-to-point full-duplex links.

Chapter 4

Radio Medium Access

The design and performance of a *medium access control* (MAC) layer protocol, suitable for the ATM based radio LANs, are presented in this chapter. Carrier Sense Multiple Access with Acknowledgement Error Detection (CSMA/AED) is a single frequency based protocol, proposed for ATM *cell*[1] transfer over radio. In conjunction with carrier sensing, the protocol uses a per-cell acknowledgement packet in order to reduce the cell loss rate by indirectly detecting the errors, caused either due to collisions or other causes of cell corruption during transmission. The results presented in this chapter are obtained from a simulation model, developed for investigating the protocol performance at the medium access layer and its effects on the end-to-end network performance.

4.1 Introduction

The purpose of the present design is to provide a medium access control protocol for efficient ATM cell transportation over radio. Most of the existing strategies [Nanda 91, Khan 91, Buchholz 91] rely either on multiple carrier frequencies or on slot reservations, based on some specific scheduling policies. While the reservation based protocols offer a better control on radio resource management, they suffer from poor granularity of *multiplexing*, which is a primary requirement of the ATM traffic. With these schemes, when a mobile entity migrates from one transmission cell to another (these events are known as *handovers* [Chia 91]), it is necessary to reserve a new slot for the mobile entity within the new transmission cell; the associated reservation delays deteriorate the handover performance which, in turn, causes application layer service degradation. The spread spectrum and CDMA techniques provide significant signal quality advantages in the presence of interference and multipath effects which are typical of radio medium [Cooper 83, Viterbi 91]. However, a major weakness of CDMA for multiservice environment is that for a given system bandwidth, spectrum spreading limits the peak user datarate to a relatively low value [Raychaudhuri 94].

[1] Not to be confused with the *transmission cell* of a radio entity.

In order to overcome these problems, a single frequency based random access strategy is proposed. The provision of random access without any reservation scheme makes the protocol suitable for statistical multiplexing, which is necessary for ATM cell transportation. Unlike in the wired mediums, collision detection over radio is not reliable because of the large difference between the transmitted and the received signal strength. The detection procedure is further complicated because of the presence of *hidden terminal* [Tobagi 75] effects in the radio networks. To solve these, an indirect collision detection mechanism is adopted where each fixed size ATM cell is acknowledged by its receiver entity. Absence of an acknowledgement packet therefore indicates the loss of the corresponding transmitted cell because of a collision or a transmission error. The proposed CSMA/AED [Biswas 92] protocol is further augmented for an adaptive control of the MAC layer congestion, caused due to the asymmetric loading from the *base* and the *mobile* (for definition, see Section 4.2) entities.

The following section outlines the radio LAN model including the details of an example physical layer. The descriptions of the proposed MAC layer protocol and its extensions for adaptive congestion control are included in Section 4.3. An example wireless data-link layer packet format, for encapsulating the wired network ATM cells, is presented in Section 4.4. Finally, the simulation model and the protocol performance, obtained from that model, are furnished in Sections 4.5 and 4.6.

4.2 Network Model

The present wireless mobile network is composed of a wired high speed *backbone* and a wireless (radio) interconnect. The wireless part is partitioned into several *transmission cells*, with a control point, known as a *base station* (BS), in each of the cells. The base stations, which are attached to the wired backbone, provide communication gateways between the radio network and the backbone interconnect. Any relocatable entity, known as a *mobile unit* (MU), communicates with any other network entity through the base stations and the backbone network. Although it is possible for two neighbour[2] MUs to communicate directly over radio, in order to reduce the higher layer design complexity such MU-to-MU traffic is always shipped through the backbone network.

The present medium access control protocol is aimed for a radio network[3], which uses a carrier frequency of 2.45 GHz (the same frequency is used for both the up and the down links) with 10 MHz radio channel bandwidth[4]. Using this frequency, in conjunction with the QPSK (Quadrature Phase-Shift Keying) modulation scheme, it is possible to achieve the radio switching rates of up to 10 Mbits/s. The actual transmission power per radio

[2]Two entities, with physical layer radio connectivity, are referred to as neighbours with respect to each other.

[3]The wireless ATM LAN, being developed [Porter 94] at Olivetti Research Ltd. Cambridge.

[4]This is currently an unlicenced band in UK available for short-range data links within buildings.

entity can be varied to optimise the frequency reuse and is restricted to 100 mw, which is sufficient for realising the *pico-cells* [Lawton 91] of radius around 10 metres.

The pico-cell profiles (both size and shape) within a building are highly irregular and unpredicatable because of the complicated terrain and attenuation characteristics of the building materials. For a fixed placement of a BS, the corresponding transmission cell profile may even vary with time, depending on the relative location and movement of the objects within the cell. To mitigate the unreliability problem, caused by this unpredictable pico-cell characteristics, an overlapping cell strategy is introduced where an MU is likely to be seen simultaneously by multiple base stations. *Pico-cell overlapping* is also used as the key component while designing the low-latency handovers, described in Chapter 7. However, in a single carrier frequency scenario the cost paid for this is a reduction of overall frequency reuse efficiency.

A range of ATM networks (Cambridge Fast Ring [Hopper 88], Cambridge Backbone Ring [Greaves 90], Fairisle network [Leslie 91] and Olivetti Switch Fabric [Milway 94]), currently operational within the Olivetti Research Ltd. and the Cambridge Computer Laboratory, can be used as the backbone of the proposed wireless LAN. In order to implement the internetworking functionalities, it is necessary to partition the backbone network into multiple administrative or geographical domains. One such partitioning and the associated addressing schemes are presented in Section 6.2.2.

4.3 Medium Access Control (MAC)

CSMA/AED is an extension of the *carrier sense multiple access* scheme with an acknowledgement error and collision detection mechanism. The reception errors in radio are usually caused by the noisy channels and the various kinds of collisions. As mentioned earlier, in radio it is difficult to detect the regular collisions reliably by comparing the incoming and the outgoing signals. The detection problem becomes worse in a *hidden collision* situation, as illustrated in Figure 4.1. Both the entities 'A' and 'C' are simultaneously sending packets to a third radio entity 'B'. Since 'A' and 'C' are out of the transmission ranges of each other, the collisions at the radio entity 'B' remain *hidden* from both the transmitter entities.

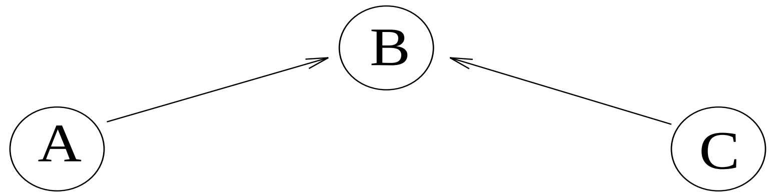


Figure 4.1: Occurrence of a hidden collision

4.3.1 Protocol

An acknowledgement is used for each transmitted packet in order to detect the mentioned errors. A packet, consisting of a preamble, a data-link layer PDU and a CRC code, is termed as a *MAC layer information cell.* Figure 4.2 shows a MAC layer frame which indicates the combined duration of the information cell transmission time (T_{cell}), the maximum waiting time for an acknowledgement packet (T_w) and the acknowledgement packet transmission time (T_{ack}). Note that the radio data-link layer provides the necessary wireless encapsulation for the wired ATM cells; description of an example data-link layer PDU and the corresponding acknowledgement packet formats are presented in Section 4.4.

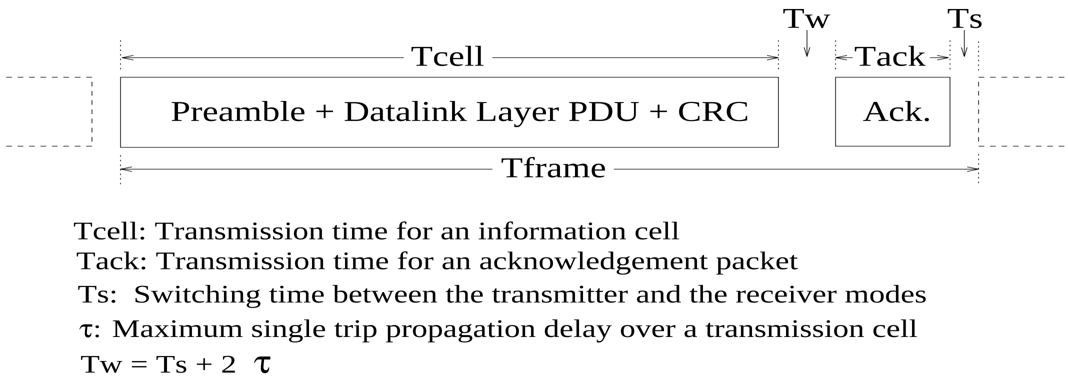


Figure 4.2: A MAC layer frame

When an entity intends to send an information cell, it senses the channel and waits until it is free. Once free, the cell is transmitted with necessary data-link layer information and a suitable CRC code. After transmission, the source waits for a specific time period T_w to receive an acknowledgement from the receiver entity. Figure 4.3 explains why the waiting time T_w is chosen as $T_s + 2\tau$.

If the source fails to receive an acknowledgement within a stipulated time period, it assumes that the information cell has been corrupted either due to a collision (direct or hidden) or because of an error caused by the noisy channel. In case of a failure a retransmission of the cell is to be arranged according to an appropriate back-off strategy [Metcalfe 76]. Depending on the specific traffic type, there can be an upperbound of the number of possible retransmissions of a cell. It should be noted that no *sliding window* mechanism is provided and that implies that the source may not wait for the acknowledgements for multiple outstanding information cells.

When a radio entity successfully receives an information cell, it checks the data-link PDU header and the error checking code in order to decide whether to reply or not. If necessary, the receiver sends an acknowledgement packet identifying the received information cell. Although no restriction is specified, the minimum possible acknowledgement packet size should be used in order to improve the channel utilisation efficiency. A mechanism for handling

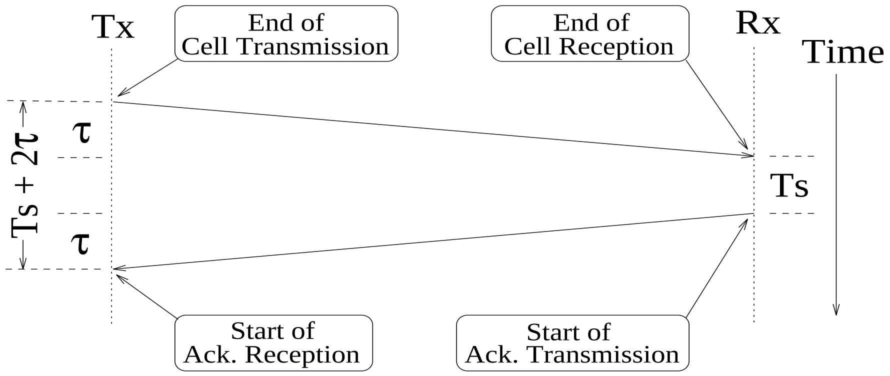


Figure 4.3: Waiting time for acknowledgement

cell duplications, caused by corrupted/lost acknowledgements, is outlined in Section 4.4.

In the above scheme, if a source entity starts sensing the channel in the middle of a frame (i.e. while some other entity is transmitting a cell), the sensing station will transmit its own information cell as soon as the channel becomes free; almost certainly, such a transmission would cause the cell to collide with the acknowledgement of the previous frame. This happens because of the lack of any synchronisation between the old and the current frames. In order to achieve certain degree of *frame synchronisation*, the protocol is modified as shown in Figure 4.4.

After waiting for a while, when a sensing source entity finds a channel free, in order to minimise the chance of a collision, it should defer the cell transmission till the end of the current frame is detected. The required amount of deferment is $T_{ack} + T_s + 2\tau$, which is the maximum interval between the end of an information cell transmission and the end of the corresponding MAC layer frame. In case when the last detected signal corresponds to an acknowledgement packet, no deferment of transmission is necessary, because the end of the acknowledgement packet indicates a frame boundary anyway. For a collision signal, no deferment of transmission is necessary because of a different reason: since the source entity cannot make out the nature of the colliding signals, it decides to restart the frame synchronisation and starts cell transmission immediately. Frame synchronisation however needs some enhancements of the hardware capabilities in order to detect the type of the sensed signals. One way of differentiating between the information cells and the acknowledgement packets is to use two different transmission datarates.

When a source entity starts sensing the channel during the interval between the end of a cell transmission and the start of an acknowledgement transmission (marked as T_w in Figure 4.2), it finds the channel free and immediately starts transmitting a cell; this cell may collide with the acknowledgement of the previous frame. Note that the present frame synchronisation scheme is not immune from this rare form of collisions.

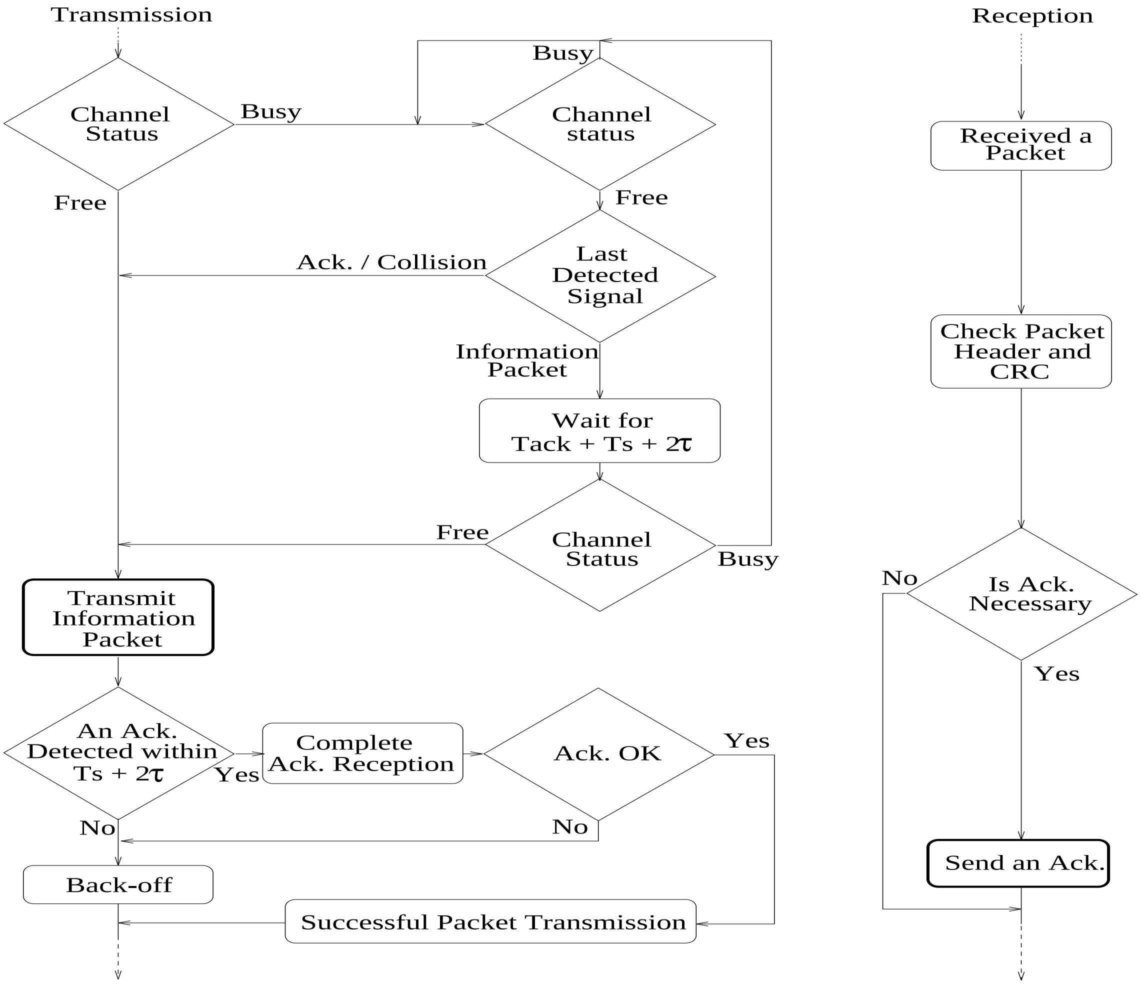


Figure 4.4: MAC layer protocol with inter frame synchronisation

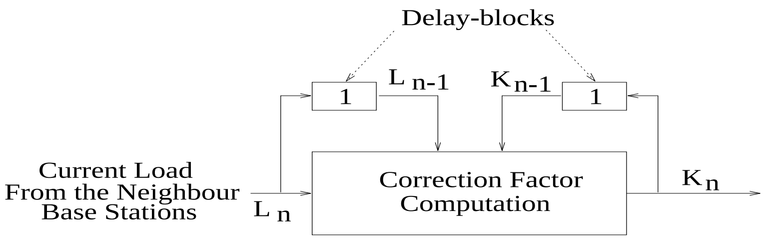
4.3.2 Adaptive Back-off

Assuming only end-to-end (via backbone) MU traffic, if L represents the average load (bits/s) from each of the mobile units in the network then the mean radio traffic load from a base station can be expressed as $L(N_m/N_b)$, where N_m and N_b represent the number of MUs and BSs respectively. Since a network is expected to contain more mobile units than the base stations (i.e. $N_m > N_b$), a BS needs to deliver more radio traffic than its mobile counterpart. In the absence of any priority mechanism for the base stations, this asymmetric loading may cause a downlink (BS to MU) congestion in the radio part of the network. Presence of such congestion is further verified from the results presented in Section 4.6. In order to overcome this problem, the following measures can be taken.

- Variable Persistency: Transmission persistence factor[5] for the base stations can be chosen to be larger than the one for the mobile units. This provides higher channel access priority to the base stations. Guided by the load within a transmission cell, the BS and MU persistency factors can also be tuned adaptively.

- Adaptive Back-off Delay: Both static and dynamic allocations of the asymmetric back-off delays, for the base and mobile entities, can be used as another way of assigning higher channel access priority to the base stations.

In order to maintain implementation simplicity, the second approach is chosen for the current simulation work. Under this scheme, soon after a transmission failure an *exponential* back-off delay [Metcalfe 76] is computed for both the mobile and the base entities. A base station defers its cell retransmission according to this computed delay. For an MU however the back-off delay is further modified by a correction factor K, tuned adaptively according to the radio traffic load from the neighbour base stations. The computing procedures for this adaptive tuning of the correction factor are depicted in Figure 4.5.



$K_0 = 1$ and $K_n >= 1$ for all n
if ($L_n > L_{n-1}$) then $K_n = K_{n-1} + 1$
if ($L_n < L_{n-1}$) then $K_n = K_{n-1} - 1$
if ($L_n = L_{n-1}$) then $K_n = K_{n-1}$
Effective Delay = Exponential Back-off Delay x Kn

Figure 4.5: Functional schematic for the adaptive correction factor computation

An increasing trend in the neighbour base load causes the correction factor to increase and that, in turn, prioritises the base stations by increasing the MU cell retransmission delays. Similarly, with diminishing neighbour base load, a mobile units gets a better share of the radio bandwidth by reducing the value of its correction factor. An effective implementation of this scheme requires each MU to keep a record of the number of information cells received form the neighbour base stations during a specific period of time. To reduce the computation overhead, the correction factor can be updated once in a specific interval (e.g. say after each 5000 cell transmissions, as done in the current simulation), instead of computing it for each cell retransmission. Performance of this scheme is presented in Section 4.6. Note that

[5]When a source entity decides to start transmitting an information cell, after carrier sensing it does so with a probability ‘p’; this number ‘p’ ($0 < p <= 1$) is known as the persistence factor.

the current mechanism attempts to alleviate only the MAC layer congestion problem by assigning dynamic priorities to the base and the mobile entities. The conventional upper layer flow and congestion control schemes are also necessary for end-to-end services over radio.

4.4 Data-Link Layer

Although the data-link layer details are not essential for the current MAC layer protocol description, for the sake of completeness, an example radio data-link packet format is presented in this section. The different fields of the packet are not yet standardised and that is why no specific lengths are mentioned about them; the field lengths will depend very much on a specific implementation. Figure 4.6 illustrates the radio specific information required for encapsulating a 48 octet ATM payload into a wireless ATM cell.

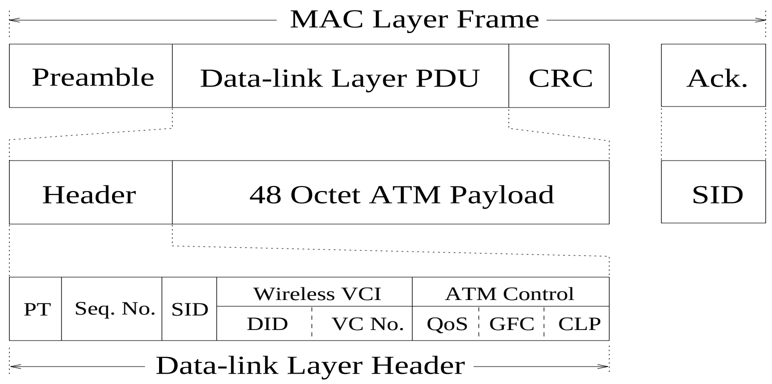


Figure 4.6: An example radio data-link layer packet format

A data-link layer header contains the usual ATM control fields [NCLATM 92, ATMUNI 93], carrying the QoS, GFC and the CLP information. The control field *Payload Type* (PT), indicating the service type (e.g. supervisory/control, CBR, VBR, data, ..), is also added at the front in order to enable the base stations to allocate resources and to perform various control and management operations.

The SID and the DID in Figure 4.6 stand for *source identifier* and *destination identifier* respectively. Each of these identifiers should uniquely identify a radio entity over the entire network. One way of achieving this is to use the hardware identifier of an entity as its SID and DID. This approach however requires enormous data-link packet headers for accommodating the large hardware identifiers. Another solution, as proposed in [Porter 94], is to use a combination of a permanently assigned *domain identifier* and a temporarily assigned short *radio entity identifier* as the SIDs and the DIDs. The upper layer management overheads, for the allocation and recycling of the domain and the radio entity identifiers, are deemed to be justified when the advantages of having shorter radio data-link packets are considered.

As in the wired ATM networks, the virtual circuit identifier (VCI) is allocated at the destination so that the VCI space can be partitioned at the receiver end in order to optimise the table lookup. Since radio is a multiple access medium, it is necessary to put the DID in the VCI field in order to specify the receiver of a cell. The DID, along with a virtual circuit number, constitutes a wireless VCI, which may be compressed [Raychaudhuri 94] in order to keep the header size manageable.

Since no sliding-window is supported by the current MAC layer protocol, the source identifier is the only information necessary for acknowledging a packet. This requires a data-link layer packet to have an SID field. The sequence number is necessary for avoiding the packet duplications, caused by the corrupted and the lost acknowledgement packets. During each packet transmission, the radio interface of the source entity puts this sequence number, which is monotonically incremented in hardware after each cell transmission (but not after retransmissions). At the receiver end this number is used to discard duplicate cells once they have been read from the interface.

4.5 Simulation Model

A radio network was simulated for investigating the CSMA/AED protocol behaviour at the medium access layer and its effects on the end-to-end network performance. A connection-oriented approach is assumed where a set of virtual circuits are setup at the beginning of a simulation run and the quality of traffic transportation through those connections are studied in the steady state.

4.5.1 Network Topology

The base stations are regularly arranged in a three dimensional grid, as illustrated in Figure 4.7. Each horizontal plane corresponds to a floor of a building. A uniform random distribution is chosen for placing the mobile units within the base station grid. The transmission characteristics are assumed to be uniform within a spherical pico-cell, with a BS at its centre. In most of the cases, a high degree of cell overlapping is used for studying the effects of the population density of the radio entities; it is also ensured that a mobile unit is always seen by at least one base station. Note that the mobile units are assumed to be static during a simulation run. This assumption is quite reasonable in the sense that the total simulation time required for the end-to-end performance evaluation (of the order of seconds) is much smaller than the average grain of mobility of the MUs. Since the current simulation work attempts to study only the effects of the radio MAC layer protocol behaviour on the end-to-end performance, instead of simulating explicitly, a simplified G/D/1 queueing model is used for representing the backbone network.

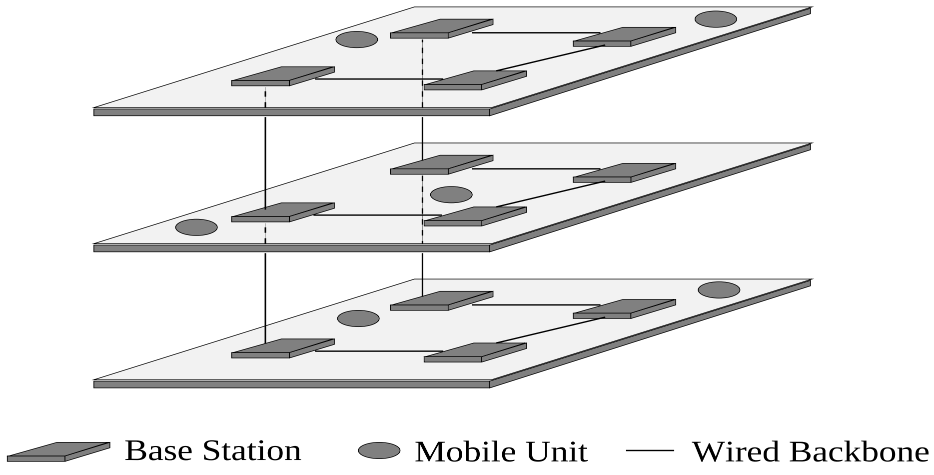


Figure 4.7: Placement of the base and the mobile entities

4.5.2 Routing

A per-connection routing scheme (note that only the MU-to-MU connections, through the backbone, are considered) is incorporated within the current simulation model. When a mobile unit intends to setup a connection, it chooses the neighbour base station with the minimum radio traffic through it. Although this requires higher layer MU management activity for monitoring the neighbour base stations' traffic conditions, the policy is adopted in order to distribute the radio traffic load, as much as possible. Once the backbone receives the connection request through the chosen base station, a distributed *mobile connection manager* entity selects the other end base station for forwarding the call to the destination mobile unit. The load distribution heuristics is again applied while choosing this remote-end base station. After receiving the call setup packet from the backbone network the destination MU transmits a reply which is relayed back to the originating mobile unit. The detailed connection management and the routing schemes, proposed for the present mobile network architecture, are presented in Chapter 7.

4.5.3 Traffic Model

At the beginning of a simulation run, each mobile unit establishes a virtual circuit with another randomly chosen mobile unit. This random destination selection strategy, along with the mentioned load balancing routing heuristics, ensures an overall uniform load distribution throughout the network. Only the mobile-to-mobile traffic via backbone, with one connection per MU, are considered in the current model. Two different traffic generator models are used. In the *synchronous* case, the blocks of cells are generated at a constant bit rate (e.g. CBR audio and video) and then pumped into the network. The second category, known as a *reciprocal arrival process* [Greaves 90], corresponds to the RPC (remote procedure call) kind of traffic in a distributed computing environment.

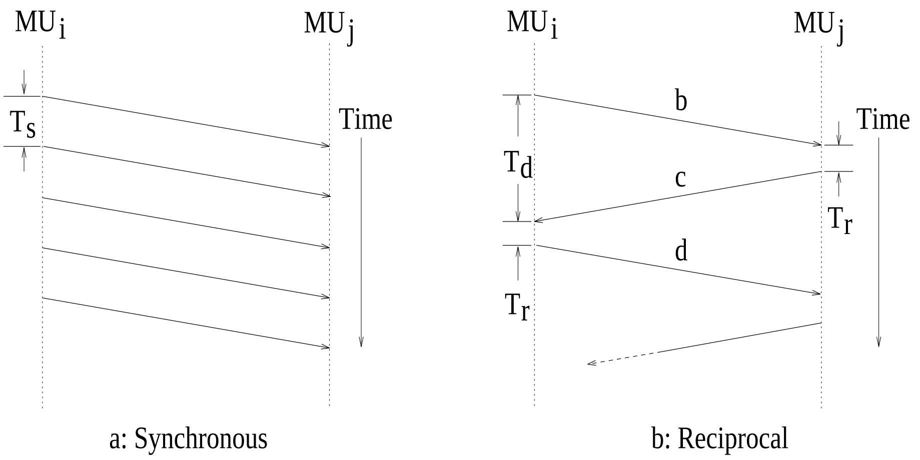


Figure 4.8: Traffic generator model

The time duration T_s, in Figure 4.8, corresponds to the inter block transmission time for synchronous traffic. In a reciprocal case, when the destination mobile unit MU_j receives a block 'b' from MU_i, the former sends a reply block 'c' back to MU_i after a *response delay* of T_r. A similar action is taken by MU_i after it receives the block 'c'. Inter block transmission time T_s determines the load exerted on the network by a synchronous traffic stream. Load in a reciprocal scenario is controlled by the mobile unit response time T_r.

Item	Value
Backbone Network Bandwidth	100 Mbits/s
Wireless ATM Cell Transmission Rate	10 Mbits/s
Wireless Acknowledgement Transmission Rate	3 Mbits/s
Normalised Pico-cell Size	0.65
Bit Error Rate	10^{-6} - 10^{-3}
Wireless ATM Cell Size	53 Octet
Acknowledgement Packet Size	2 Octet
Transceiver Switching Time (T_s)	5 μs
Minimum Back-off Delay	One MAC Layer Frame
Block Size for Synchronous Traffic	1
Block Size for RPC Traffic	3

Table 4.1: Parameters, specifying the network, MAC layer and the traffic generator

The baseline parameters, describing the network environment, MAC layer and the traffic model specific details, are summarised in Table 4.1. Note that two different radio transmission rates are used in order to enable a radio entity to distinguish between an information cell signal and an acknowledgement packet signal; this is required by the frame synchronisation operation of CSMA/AED protocol, as described in Section 4.3.1. The pico-cell size is normalised by the separation distance between two adjacent base stations. Other than where specified, a pico-cell size of 0.65 is used for all the presented performance graphs.

4.6 Performance Results

The simulation results, presented in this section, are aimed mainly towards the evaluation of the end-to-end performance of the described network, operating under the proposed MAC layer protocol. An event driven simulator was written in *C*, in order to obtain the following performance figures. For each simulation, the program was allowed to run for a sufficiently long *warming up* period before the statistics collection took place. All the presented throughput figures are normalised by the raw radio transmission bandwidth which is 10 Mbits/s. Similarly, the delays and mobile response times are normalised by the information cell transmission time ($T = 42.4\mu$s) of a 53 byte cell. Unless otherwise stated, all the presented graphs correspond to a network containing 48 base stations.

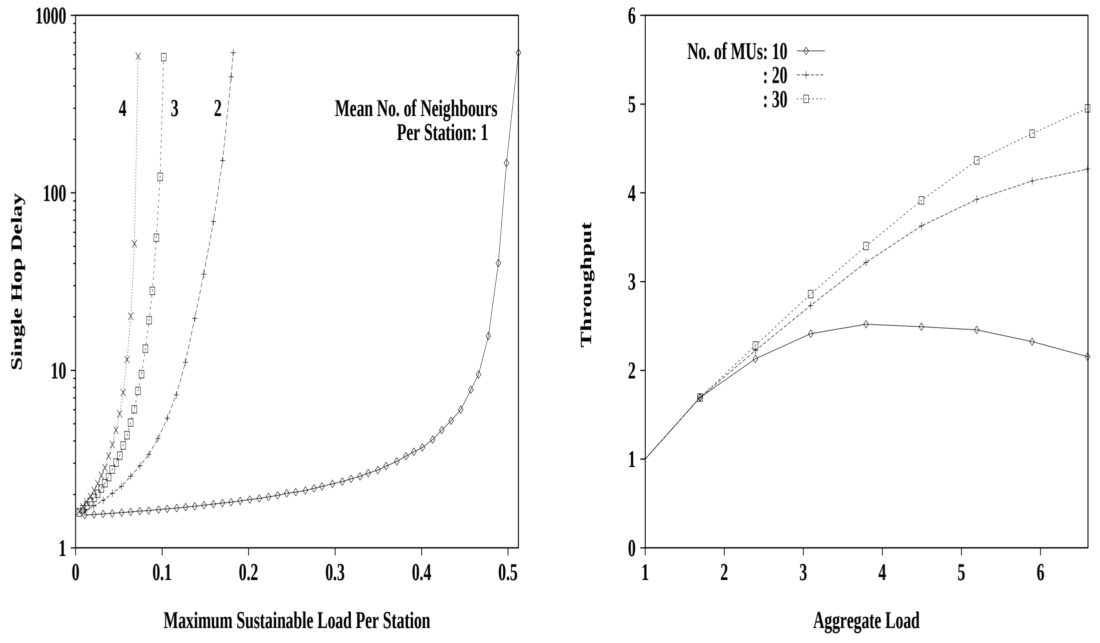


Figure 4.9: Single hop MAC layer delay and end-to-end network throughput for synchronous traffic

The first graph in Figure 4.9 shows the variation of the mean single hop delay with the maximum sustainable throughput (i.e. load is equal to throughput) for each of the participating radio entities. The presented delay figures indicate the sum of the queueing latency at the MAC layer and the actual medium access time. A synchronous traffic generator, with adjustable inter block arrival period, is used for this measurement. The load versus delay characteristics are studied for the different station densities. An example of the limiting case, with one neighbour, would be a situation where an MU is the only resident within the transmission cell of a base station. For the same type of traffic (synchronous), the end-to-end network throughput is shown in the second graph of Figure 4.9. The results are collected with varying number of mobile connections (one connection per MU) which are setup at the beginning of the corresponding simulation run.

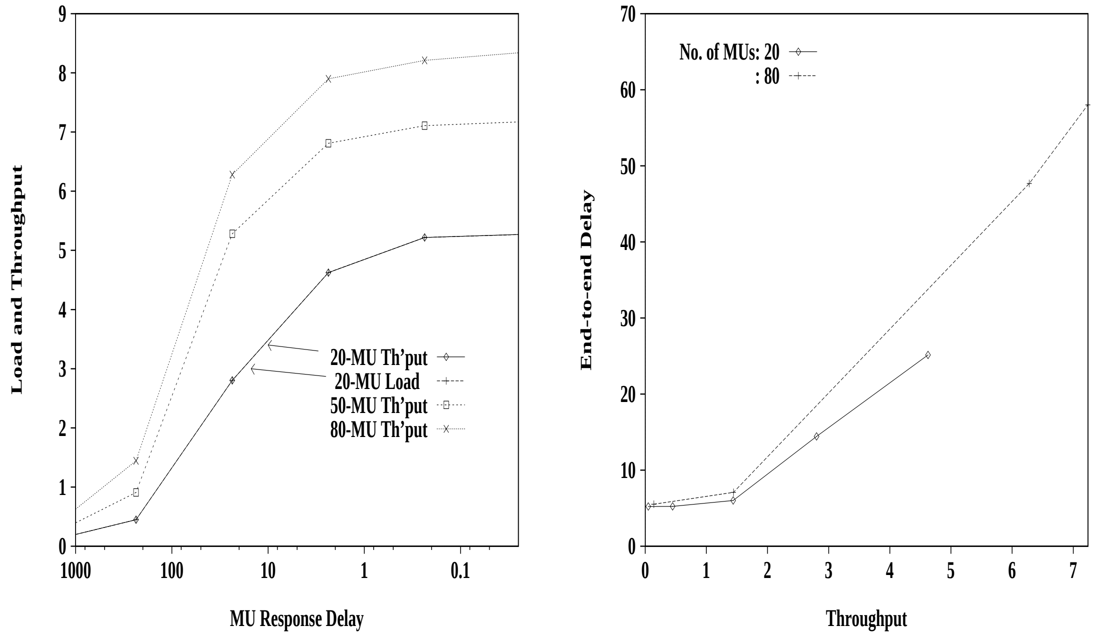


Figure 4.10: Delay and throughput for RPC traffic

The end-to-end network performance for the reciprocal RPC traffic are presented in Figure 4.10. Unlike in the synchronous case, the wait-and-go approach for the reciprocal traffic provides an implicit end-to-end flow control which, in turn, keeps the network stable by always ensuring that the applied load on a connection would always be the same as its throughput. This is verified from the throughput curves in Figure 4.10. As explained in Section 4.5.3, for a specific topology the maximum attainable throughput (and that is equal to the maximum sustainable load) is determined by the mobile unit response delay T_r. In the presence of 50 mobile units, for instance, the maximum attainable throughput is almost 7 times the raw radio bandwidth which is 10 Mbits/s for the present model. The corresponding MU response delay is approximately 4.24μs (i.e. $0.1T$). Once saturated, the network maintains stability by preventing the users from pumping any more traffic into it. It is worth noticing the difference between the maximum steady state throughput for the synchronous and the RPC types of traffic. The delay figures for the RPC traffic are also presented in Figure 4.10. With 80 active connections, a near maximum normalised throughput of 7.5 can be achieved with a decent end-to-end delay of $55.5T$ which is 2.35 ms. Note that with 20 mobile units, the maximum attainable throughput causes the corresponding delay curve to end abruptly.

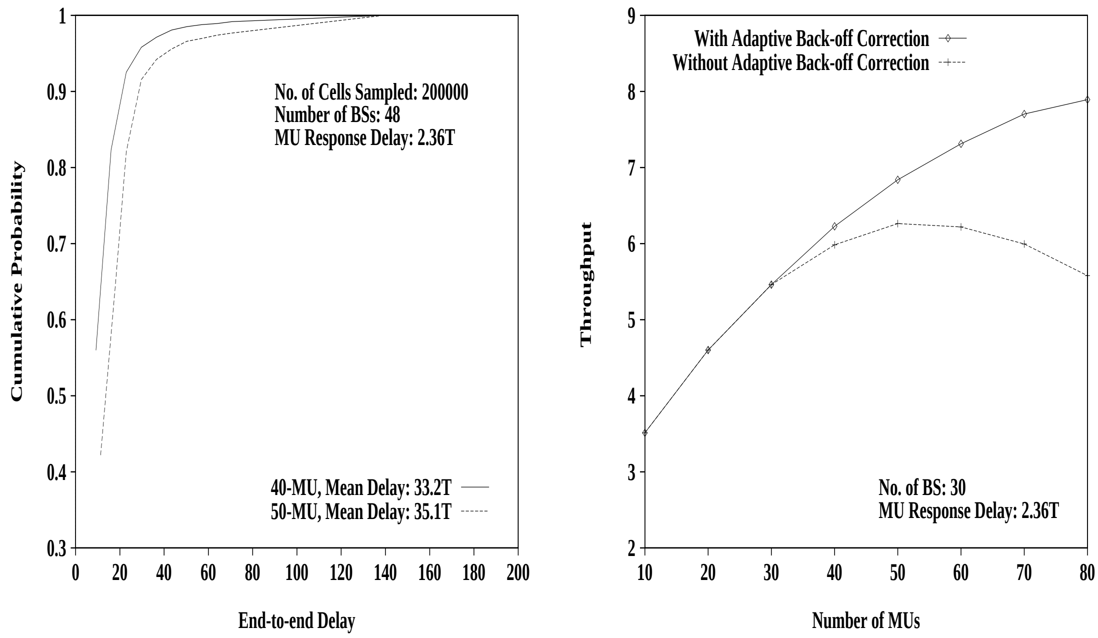


Figure 4.11: Jitter and MAC layer congestion control performance

The variation of the end-to-end cell transfer delay, referred to as *jitter*, is an important performance criterion in a realtime traffic environment. The jitter, expressed as the cumulative probability distribution of the end-to-end delay, is shown in the first graph of Figure 4.11. Each of the curves plots the probability of a cell being delivered within a specific time duration. For example, in a network with 40 MUs one would expect 95 percent of all the cells through a connection to be delivered with a maximum delay of $27T$ which is 1.14 ms. The networks with larger number of mobile units suffer from worse delay variance, caused by the increased number of MAC layer collisions.

The aspects of the MAC layer congestion, caused by an asymmetric base and mobile loading, were discussed in Section 4.3.2. The effectiveness of the proposed congestion control mechanism, based on an adaptive back-off delay *correction*, is illustrated in the second graph of Figure 4.11. For less than 30 mobile units in the network, there is no noticeable performance difference between the *with* and the *without* correction situations. This is because there exists no uneven medium access competition between the base and the mobile entities. For higher number of mobile units however the CSMA/AED *with* adaptive back-off correction scheme distinctly outperforms its *without correction* counterpart. The curve, corresponding to the without correction case, also illustrates the severity of this congestion. Note that the *with correction* version of the protocol is used for obtaining all the other curves, presented in this chapter.

The performance of a inherently unreliable wireless network depends a great deal on the error situation of the wireless environment. Such physical layer unreliability is usually expressed as the *bit error rate*, which represents the probability of an error during a single

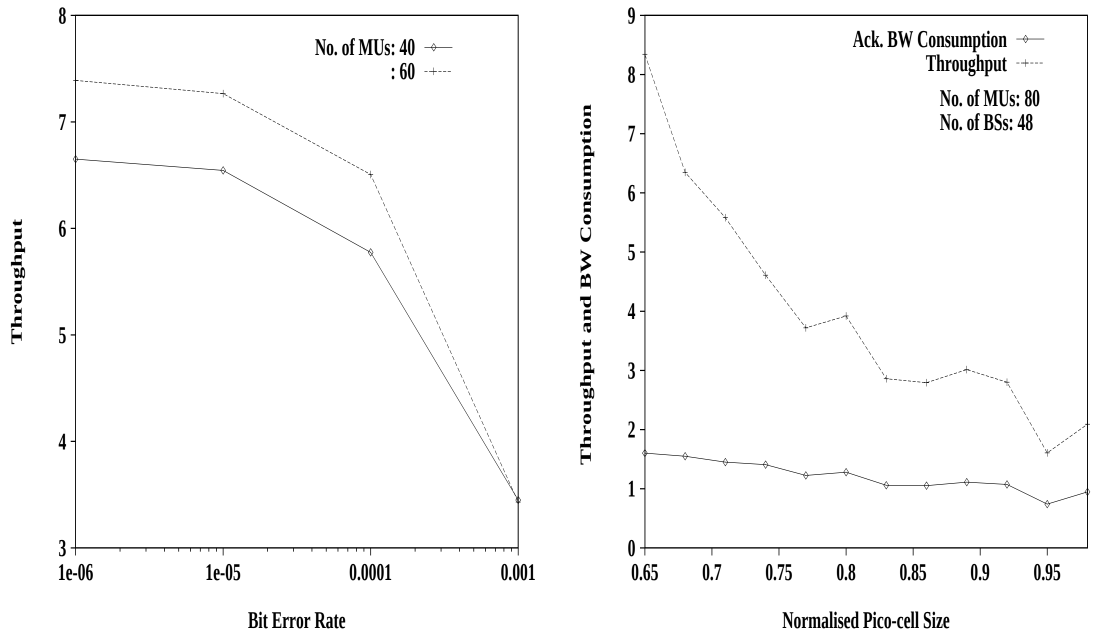


Figure 4.12: Effects of bit error rate and pico-cell size

bit transmission. The throughput variation, with different bit error rate scenario, is shown in the first graph of Figure 4.12.

The second graph of Figure 4.12 illustrates the effects of the pico-cell size on the network throughput. In the present high cell overlapping scenario, with increasing pico-cell size the average number of neighbours per radio entity also increases. This gives rise to a higher collision probability, which reduces the end-to-end network throughput because of the degraded MAC layer performance. A high collision rate however reduces the requirements for the acknowledgement transmissions and that, as indicated in Figure 4.12, curtails the radio bandwidth consumption by the acknowledgement packets. Since the MUs are placed randomly, an increase of the pico-cell size does not increase the average number of neighbours uniformly. This accounts for the irregularities of the curves in the second graph of Figure 4.12.

4.7 Summary

A carrier sensing based radio medium access control protocol was presented in this chapter. The performance of the protocol in a connection-oriented network environment, has been studied with the aid of a discrete event simulator. In the described network model, it was shown that the asymmetric loading, caused by the unequal numbers of the base and the mobile entities, gives rise to a MAC layer congestion problem. An adaptive back-off strategy for the mobile units was proposed as a partial solution of this congestion problem.

An example data-link layer encapsulation detail, including the corresponding wireless VCI format, were also presented in this chapter.

Although the obtained delay, jitter and the throughput characteristics comply with the basic requirements of the multiservice traffic, a number of features of the CSMA/AED protocol are identified as the potential candidates for further augmentation. Incorporating a MAC layer traffic priority structure would further enhance the protocol capabilities for transporting multiservice traffic. Introduction of capture [Cidon 85] and the adaptive power control with a suitable *antenna diversity* scheme may also prove to be useful for the CSMA/AED performance enhancement. Another interesting approach would be to incorporate some synchronisation strategy for the base stations, where by exchanging synchronisation messages through the wired backbone network, the base stations can reduce the number of collisions in radio. However, too much synchronisation may convert the protocol into a reservation based scheme, which is not desirable. It is also worth investigating the effects of the irregular pico-cell shapes and the distributions on the performance of the presented medium access protocol.

Chapter 5

Flow Control Model

This chapter presents a hybrid queueing model for evaluating the performance of a window based flow control mechanism, operating in a connection-oriented radio LAN environment. Each end-to-end connection is modelled using a queueing representation of the radio data-link, transport and the application layer protocol entities. The simulator, written for CSMA/AED protocol in Chapter 4, is used to generate the radio data-link layer model. A multichain closed queueing network model, representing the end-to-end network connections, is solved using an approximate Mean Value Analysis (MVA) technique. A novel technique for improving the convergence speed of the solution is also developed in this chapter. The presented model is primarily used for studying the end-to-end connection performance and to investigate the location and the patterns of network bottleneck with varying application characteristics. The results obtained from this model can be used for choosing the optimum window size, for a given application and the other network specifications.

5.1 Introduction

The present model is developed with an aim for investigating the performance of a per-connection sliding window protocol, operating in a packet radio environment, described in Chapter 4. A simplified OSI protocol stack is considered, where the radio data-link, transport and the application layers are modelled individually for each end-to-end connection. CSMA/AED, which provides an error free service to the upper layers, is used as the medium access and the data-link layer protocol. Since in a connection-oriented environment most of the network layer signalling is done in an out-of-band [McAuley 90] fashion, very little in-band protocol processing is required in the network layer. In order to reduce the model complexity, this lightweight network layer and the usual transport layer functionalities are encapsulated into a single transport layer queue. The tasks, involving connection setup/termination and flow control are modelled by the application layer queues. Note that the current model does not take handovers and the other mobility related issues into account.

The mean delays obtained from the simulator, used for investigating the performance of the CSMA/AED channel access algorithm (see Chapter 4), are used directly as the service times of the radio data-link layer queues in the present end-to-end connection model. In order to model the per-connection sliding window flow control, a modification of the technique described in [Reiser 79] has been used. In [Reiser 79], a queue service time is assumed to be independent of the throughput of all the virtual circuits passing through that queue. It also assumes that the service profiles of no two queues interact with each other under any circumstances. In a multihop radio environment however the service latency of a radio data-link layer queue is affected by the throughput of each individual connection passing through it. Because of the MAC layer interference between two *neighbour* radio entities, the service time of a radio data-link layer queue is further affected (see Figure 4.9) by the aggregate throughput of the connections, passing through its neighbour radio entities. To cope with this interdependence of the link layer queue service profiles, a modification of the technique, presented in [Reiser 79], is proposed in [Biswas 93]. In the present hybrid approach, the radio data-link and the end-to-end connection performance are modelled separately. This has the advantage that it is then possible to use the same end-to-end modelling technique with different data-link layer models, derived from various channel access algorithms.

Organisation of the rest the chapter is as follows: A description of the flow control mechanism, operating in a wireless virtual circuit environment, is presented in Section 5.2. Section 5.3 is devoted to a description of the queueing model and the analytical technique used for solving it. Finally, an example network model is solved numerically and the results are presented in Section 5.4.

5.2 Flow Control

The present model assumes a connection-oriented wireless network environment, as described in Chapter 4. Although a per-connection routing strategy is employed, no specific assumptions regarding the connection routing policies are necessary. Two example virtual circuits are pictured in Figure 5.1 where the virtual circuits VC_1 and VC_2 interfere within the base station BS_m and within the radio space shared by BS_m, MU_h and MU_j. Note that the packet transfer from one base station to another may involve multiple wired hops, depending on the relative geographical locations of the end mobile units. The present model deals only with the MU-to-MU connections.

A sliding window flow control [Crosby 91] mechanism is used for reducing end-to-end congestion within the radio LAN. By assigning a fixed window size to each virtual circuit, this mechanism limits the total number of packets within a VC at any point of time. The flow control is implemented only on an end-to-end basis where the end mobile units keep track of the total number of outstanding packets within a virtual circuit. Each MU simultaneously maintains multiple virtual circuits, through which the packets are sent with a *piggybacking* acknowledgement scheme. When MU_j, in Figure 5.1, receives a data packet from MU_i on VC_2, instead of sending a separate acknowledgement packet, a part of the next data packet

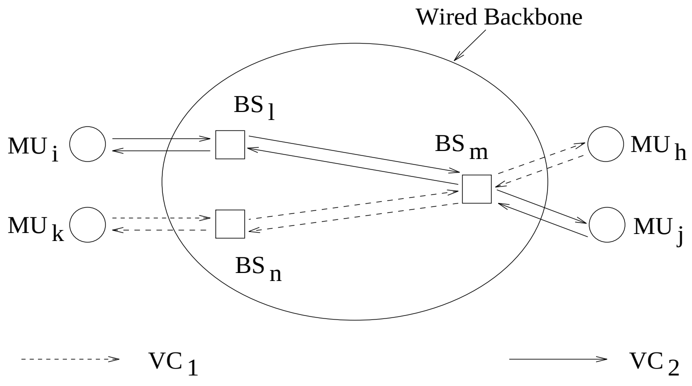


Figure 5.1: Virtual circuits through radio

to MU_i is used for acknowledging the last received packet from MU_i.

5.3 Modelling End-to-end Flow Control

An end-to-end queueing model of the mentioned flow control scheme is developed in this section. The purpose of the model is to evaluate the network performance on a per connection basis and to study the location and pattern of network bottleneck with varying application and flow control specifications. The present model does not address the signalling related issues; it only considers the steady state flow controlled packet transfer through the pre-established end-to-end virtual circuits. Considering a reliable data-link layer protocol (CSMA/AED), it is assumed that no packet loss/error is present above that layer. An infinite buffering capacity is assumed at all the intermediate base and the terminal mobile entities. It is also ensured that the number of VCs per mobile and base entity is uniform throughout the network. This homogeneity is necessary while applying the data-link layer simulation results into the present end-to-end model.

5.3.1 Queueing Model

Figure 5.2 illustrates a layered representation of the virtual circuit VC_2, shown in Figure 5.1. The packet handling within a mobile unit is modelled as three separate protocol layers. Both the medium access and the logical link control tasks are represented by a single *radio data-link* layer. Similarly, the network layer and the transport layer protocol functionalities are encapsulated within the *transport* layer. The lightweight network layer (assuming out-of-band signalling and the absence of in-band multiplexing), in the present connection-oriented environment, justifies this merging of those two protocol layers. Note that the presented flow control scheme operates on an end-to-end basis and it is solely controlled by the *application* layer entities.

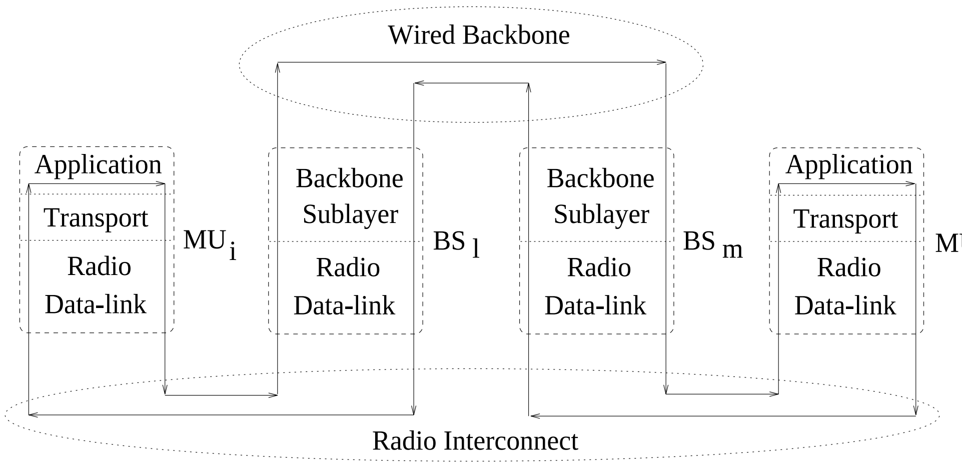


Figure 5.2: Layered model of a virtual circuit

The base stations provide gateway services between the wired backbone and the radio interconnect. The functionalities of these base stations are modelled by a radio data-link layer and a *backbone* sublayer. Since the present model attempts to investigate only the effects of the flow control strategy from the end mobile units' point of view, instead of modelling explicitly, the entire backbone interface (including the relevant protocol layers) and the upper layers of the radio protocols in a base station are encapsulated within a simplified *backbone* sublayer.

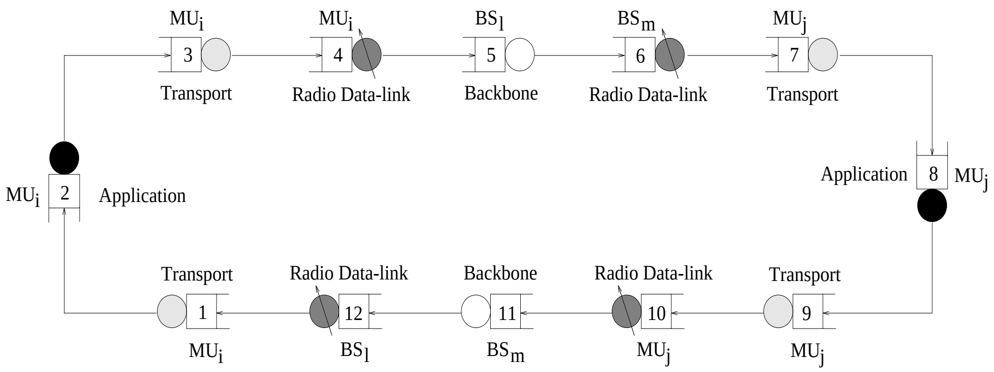


Figure 5.3: Closed queueing network model for a virtual circuit

Figure 5.3 illustrates the queueing network representation [Murata 88] of the virtual circuit VC_2, shown in Figures 5.1 and 5.2. The total number of packets within a closed chain, representing a virtual circuit, is equal to the sliding-window size of the applied flow control scheme. To represent the process abstraction of the application softwares, the application queues are modelled as the delay nodes. Apart from relaying packets between the data-link layers and the application layers, a transport layer also offers the necessary per connection segmentation and reassembly services. A mobile unit transport layer is modelled as a simple FCFS queue with a constant service time which corresponds to the fixed length packets, used in the current model. A radio data-link layer is modelled by using the throughput-delay curve, shown in Figure 4.9. It is represented as a delay node where the service time depends on the total instantaneous flow through the node. The backbone sublayer is modelled as an FCFS queue with an exponential service time distribution.

5.3.2 Analysis

The present solution scheme is based on the multi-chain mean value analysis (MVA) technique, proposed in [Reiser 79]. In order to cope with the flow-dependent service times of the radio data-link queues, the following modified version of the MVA is used.

Each virtual circuit is represented by a closed chain, containing 12 different queues, as shown in Figure 5.3. The formal definition of the network is as follows.

N:	The total number of queues within the network
R:	The total number of closed chains
i:	The index for representing the queue i where $i = 1, \ldots, N$
j:	The index for representing the chain j where $j = 1, \ldots, R$
W^j:	The population of chain j; it also represents the window size of the virtual circuit j
$R(i)$:	The set of chains passing through queue i
$Q(j)$:	The set of queues, visited by chain j
$[W]$:	The population vector for all the chains: $[W^1, \ldots, W^j, \ldots, W^R]$; most of the queue parameters depend on this vector
$[W - e^j]$:	This represents the vector: $[W^1, \ldots, W^{j-1}, W^j - 1, W^{j+1}, \ldots, W^R]$
τ_i^j:	The mean service time of a chain j packet at queue i ($j \in R(i)$); initially, these parameters are considered to be independent of the population vector $[W]$
$n_i^j(W)$:	The mean chain j packet population at queue i ($j \in R(i)$)
$n_i(W)$:	The mean packet population at queue i; this can be written as $\sum_{j \in R(i)} n_i^j(W)$
$\lambda^j(W)$:	The mean throughput of chain j
$\lambda_i(W)$:	The mean throughput of queue i; this can be written as $\sum_{j \in R(i)} \lambda^j(W)$
$t_i^j(W)$:	The mean queueing time (waiting and service) of a chain j packet at queue i ($j \in R(i)$)
$\sigma_i^j(W)$:	The second moment of the chain j service time distribution at queue i ($j \in R(i)$)
$\rho_i^j(W)$:	The chain j utilisation at queue i ($j \in R(i)$)
$\rho_i(W)$:	The utilisation of queue i; this can be written as $\sum\limits_{j \in R(i)} \rho_i^j(W)$
$f_i^j(W)$:	The fraction of chain j throughput at queue i ($j \in R(i)$); $f_i^j(W) = \lambda^j(W)/\lambda_i(W)$

In the following derivation, only three types of queueing entities, namely, the *delay nodes*, the *exponential service time queues* and the *constant service time queues*, are considered.

The mean queueing time (derived in Section B.1) for a chain j packet at queue i can be expressed as

$$t_i^j(W) = \begin{cases} \tau_i^j & \text{delay node,} \\ \tau_i^j + \sum\limits_{k \in R(i)} \tau_i^k n_i^k(W) & \text{exponential service time,} \\ \tau_i^j + \sum\limits_{k \in R(i)} \tau_i^k [n_i^k(W) - \frac{1}{2}\rho_i^k(W)] & \text{constant service time.} \end{cases} \tag{5.1}$$

Applying the *arrival instant theorem*[1], Equation 5.1 can be rewritten as

$$t_i^j(W) = \begin{cases} \tau_i^j & \text{delay node,} \\ \tau_i^j + \sum\limits_{k \in R(i)} \tau_i^k n_i^k(W - e^j) & \text{exponential service time,} \\ \tau_i^j + \sum\limits_{k \in R(i)} \tau_i^k [n_i^k(W - e^j) - \frac{1}{2}\rho_i^k(W - e^j)] & \text{constant service time.} \end{cases} \tag{5.2}$$

Application of the *Little's Result* [Kleinrock 75] to the chain j ($j = 1, \ldots, R$) packets gives rise to the equation

$$\lambda^j(W) = \frac{W^j}{\sum\limits_{i \in Q(j)} t_i^j(W)}. \tag{5.3}$$

Similarly, for chain j ($j = 1, \ldots, R$), application of the same principle to queue i ($i \in Q(j)$) yields

$$n_i^j(W) = \lambda^j(W) t_i^j(W). \tag{5.4}$$

The Equations 5.2, 5.3 and 5.4 can be solved recursively with the starting values of $n_i^j(0) = 0$ and $\rho_i^j(0) = 0$, for $i = 1, \ldots, N$ and $j = 1, \ldots, R$. This straightforward recursive solution has an extremely high operation count of $\prod_{j=1}^R W^j$. For instance, in a 16 chain network with 10 customers per chain the operation count would be 10^{16}. In order to reduce this complexity, the following approximate MVA technique, with an iterative solution algorithm, is used. After rearranging its terms, Equation 5.2 can be written as

$$t_i^j(W) = \begin{cases} \tau_i^j & \text{delay node,} \\ \tau_i^j + \sum\limits_{k \in R(i)} \tau_i^k n_i^k(W) - \sum\limits_{k \in R(i)} \tau_i^k [n_i^k(W) - n_i^k(W - e^j)] & \text{exponential,} \\ \tau_i^j + \sum\limits_{k \in R(i)} \tau_i^k [n_i^k(W) - \frac{1}{2}\rho_i^k(W)] & \\ \qquad - \sum\limits_{k \in R(i)} \tau_i^k [n_i^k(W) - n_i^k(W - e^j)] & \\ \qquad + \frac{1}{2} \sum\limits_{k \in R(i)} \tau_i^k [\rho_i^k(W) - \rho_i^k(W - e^j)] & \text{constant.} \end{cases} \tag{5.5}$$

[1]In a closed multichain queueing network [Reiser 80] with product form solution, an arriving packet observes the equilibrium solution of the queueing network with one less packet in the arriving packet's chain.

The following heuristics are applied for all k ($k = 1, \ldots, R$) where $k \neq j$:

$$[n_i^j(W) - n_i^j(W - e^j)] \gg [n_i^k(W) - n_i^k(W - e^j)]$$

and

$$[\rho_i^j(W) - \rho_i^j(W - e^j)] \gg [\rho_i^k(W) - \rho_i^k(W - e^j)].$$

These are based on the assumption that the chain with one less packet is mostly affected [Reiser 79]. Ignoring the terms (for all $k \neq j$) in the right hand sides of the above two inequalities with respect to their left hand counterparts, Equation 5.5 can be rewritten as

$$t_i^j(W) = \begin{cases} \tau_i^j & \text{delay node,} \\ \tau_i^j + \sum\limits_{k \in R(i)} [\tau_i^k n_i^k(W)] - a_i^j(W)\tau_i^j & \text{exponential,} \\ \tau_i^j + \sum\limits_{k \in R(i)} \tau_i^k [n_i^k(W) - \frac{1}{2}\rho_i^k(W)] - a_i^j(W)\tau_i^j + \frac{1}{2} b_i^j(W)\tau_i^j & \text{constant,} \end{cases} \tag{5.6}$$

where the error terms $a_i^j(W)$ and $b_i^j(W)$ are defined as

$$\begin{aligned} a_i^j(W) &= n_i^j(W) - n_i^j(W - e^j) \\ b_i^j(W) &= \rho_i^j(W) - \rho_i^j(W - e^j). \end{aligned} \tag{5.7}$$

In order to obtain the values of $a_i^j(W)$ and $b_i^j(W)$, it is necessary to solve the single chain MVA equations for each chain j ($j = 1, \ldots, R$). It is shown in Appendix B that the effects of the background load (from the other interfering chains) can be taken into account by modifying the expression of the mean service time of a chain j packet at queue i ($i \in Q(j)$). The modified service time $\widehat{\tau_i^j}$ is expressed as

$$\widehat{\tau_i^j} = \begin{cases} \tau_i^j & \text{delay node,} \\ \dfrac{\tau_i^j}{1 + \lambda^j(W)\tau_i^j - \sum\limits_{k \in R(i)} \lambda^k(W)\tau_i^k} & \text{otherwise.} \end{cases} \tag{5.8}$$

The mean queueing time (waiting and service) for a chain j packet at queue i is

$$\widehat{t_i^j(w)} = \begin{cases} \widehat{\tau_i^j} & \text{delay,} \\ \widehat{\tau_i^j}[1 + \widehat{n_i^j}(w-1)] & \text{exponential,} \\ \widehat{\tau_i^j}[1 + \widehat{n_i^j}(w-1) - \frac{1}{2}\widehat{\lambda^j}(w-1)\widehat{\tau_i^j}] & \text{constant,} \end{cases} \tag{5.9}$$

where w is the varying population size of chain j; $w = 1, \ldots, W^j$. Note that $\widehat{t_i^j(w)}$ represents the queueing time in a single chain network. Application of the *Little's Result* to the single chain j (over the whole chain and for queue i on chain j) yields the equations

$$\widehat{\lambda^j(w)} = \frac{w}{\sum\limits_{i \in Q(j)} \widehat{t_i^j(w)}} \tag{5.10}$$

and

$$\widehat{n_i^j(w)} = \widehat{\lambda^j(w)}\widehat{t_i^j(w)}. \tag{5.11}$$

The Equations 5.9, 5.10 and 5.11 can be solved for chain j, by using the MVA algorithm for a single chain closed queueing network. With the starting values of $\widehat{\lambda^j(0)} = 0$ and $\widehat{n_i^j(0)} = 0$ for all i ($i \in Q(j)$), the quantities $\widehat{t_i^j(w)}$, $\widehat{\lambda^j(w)}$ and $\widehat{n_i^j(w)}$ are computed successively for the values of $w = 1, \ldots, W^j$. The next step is to compute the error parameters $a_i^j(W)$ and $b_i^j(W)$ from Equation 5.7.

So far, it has been assumed that the service times τ_i^j are independent of the population vector $[W]$. The results of the simulator, presented in Figure 4.9, show that the service time of a radio data-link queue can be expressed as $f(\lambda)$, where λ is the total throughput of all the virtual circuits passing through the queue. Given that the packet size is constant for all the VCs, a general expression for the queue service time can be written as

$$\tau_i^j(W) = \begin{cases} f(\sum\limits_{j \in R(i)} \lambda^j(W)) & \text{radio data-link queue,} \\ \tau_i^j & \text{otherwise.} \end{cases} \tag{5.12}$$

If a queue is of radio data-link type then the total throughput (λ) of the queue is to be computed first. Using this λ, it is then possible to find out the value of $\tau_i^j(W)$ from the radio medium access layer model in Section 4.6. Figure 4.9 illustrates the relevant throughput versus delay characteristics of the radio data-link delay servers.

Iterative solution algorithm.

1. For all j ($j = 1, \ldots, R$) initialise $\lambda^j(W) = \eta$ (e.g. $\eta = 10^{-2}$) and $n_i^j(W)$ for all $i \in Q(j)$. While initialising, the condition $\sum_{i \in Q(j)} n_i^j(W) = W^j$ for all $j = 1, \ldots, R$ should be satisfied.

2. Using Equation 5.12, compute the service times τ_i^j for all i ($i = 1, \ldots, N$).

3. For all the chains j ($j = 1, \ldots, R$), run single chain MVA using the Equations 5.8, 5.9, 5.10 and 5.11. The error parameters $a_i^j(W)$ and $b_i^j(W)$ can be computed by applying the single chain MVA results into Equation 5.7.

4. Compute the values of $t_i^j(W), (i = 1, \ldots, N; j \in R(i))$, from Equation 5.6.

5. The throughput for each chain j ($j = 1, \ldots, R$) is to be computed by using Equation 5.3.

6. Compute the packet population $n_i^j(W)$, ($i = 1, \ldots, N; j \in R(i)$), using Equation 5.4.

7. If the current iteration number is n, a convergence error factor is defined as

$$\pi = \sum_{j=1}^{R} [\lambda^j\{n\} - \lambda^j\{n-1\}] + \sum_{i=1}^{N} \sum_{j \in R(i)} [n_i^j(W)\{n\} - n_i^j(W)\{n-1\}]. \tag{5.13}$$

   If the condition $\pi < \epsilon$ (e.g. $\epsilon = 10^{-6}$) is satisfied then the iteration process has to be stopped; otherwise repeat from step 2.

To cope with the delayed convergence problems of the present scheme, a modified solution algorithm is presented in Appendix B.

5.4 Numerical Results

The proposed *hybrid* modelling technique is used in a numerical study of the window based flow control, operating in an example radio network. As required by the simulation and the end-to-end queueing model, a homogeneous traffic density is ensured by deploying equal number of base and mobile units within the radio LAN. Let N_b, N_m and N_c represent the number of base stations, mobile units and the virtual circuits in the network. Since each virtual circuit passes through two mobile and two base entities (see Figure 5.1), the number of virtual circuits through any base or mobile unit can be expressed as $N_d = 4N_c/(N_b+N_m)$. The virtual circuit density N_d represents the degree of interference between the different VCs in a network.

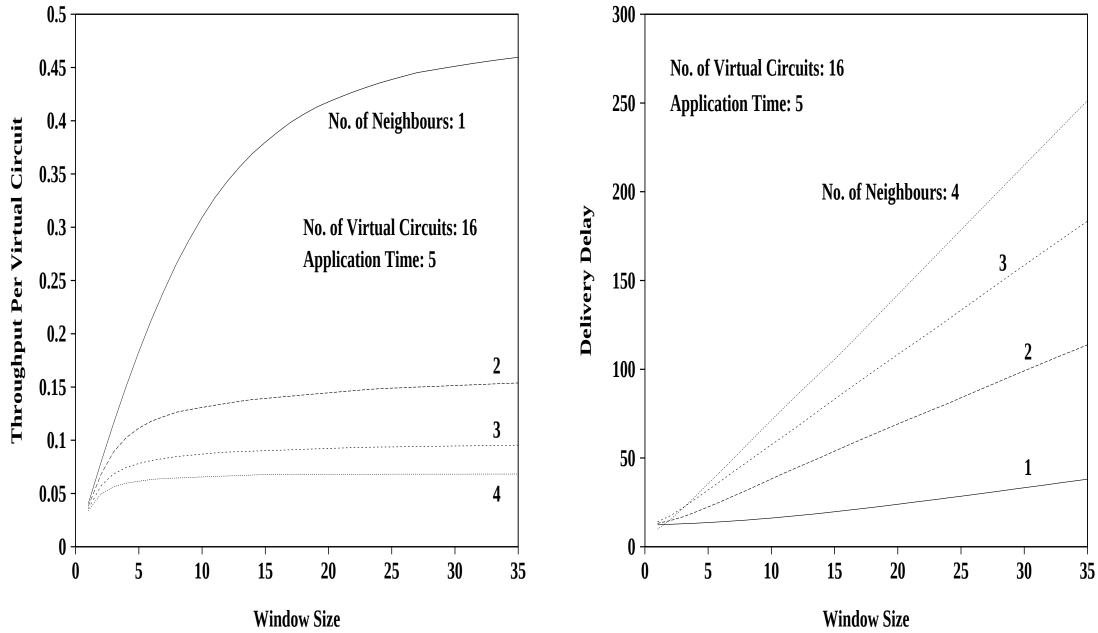


Figure 5.4: Throughput and end-to-end delay versus window size

For the results presented in this section, a network is chosen with $N_b = N_m = 32$ and $N_c = 16, 32, 48$ and 64, which correspond to the virtual circuit density $N_d = 1, 2, 3$ and 4 respectively. The queues, modelling a virtual circuit, can be classified into four different categories. The transport queue processing time is set as $2T$, where T represents the radio transmission time of a 530 octet packet. With a raw radio switching rate of 10Mbits/s (see the network model in Section 4.2), this transmission time T is equal to 424 μs. The radio MAC layer performance results (shown in Figure 4.9), for 1, 2, 3 and 4 neighbours per radio entity, are directly used for modelling the radio data-link queues. Considering a high speed backbone network, comprising of the switching elements connected by 100 Mbits/s point-to-point links, a service time of $0.25T$ is chosen for the backbone queues in Figure 5.3. A wide range (from T to $80T$) of the application layer queue service times is chosen in order to study the location and the nature of network bottlenecks. All the presented delay figures are normalised by the radio layer packet transmission time T. Similarly, the throughput is expressed in terms of the number of packets delivered in time T.

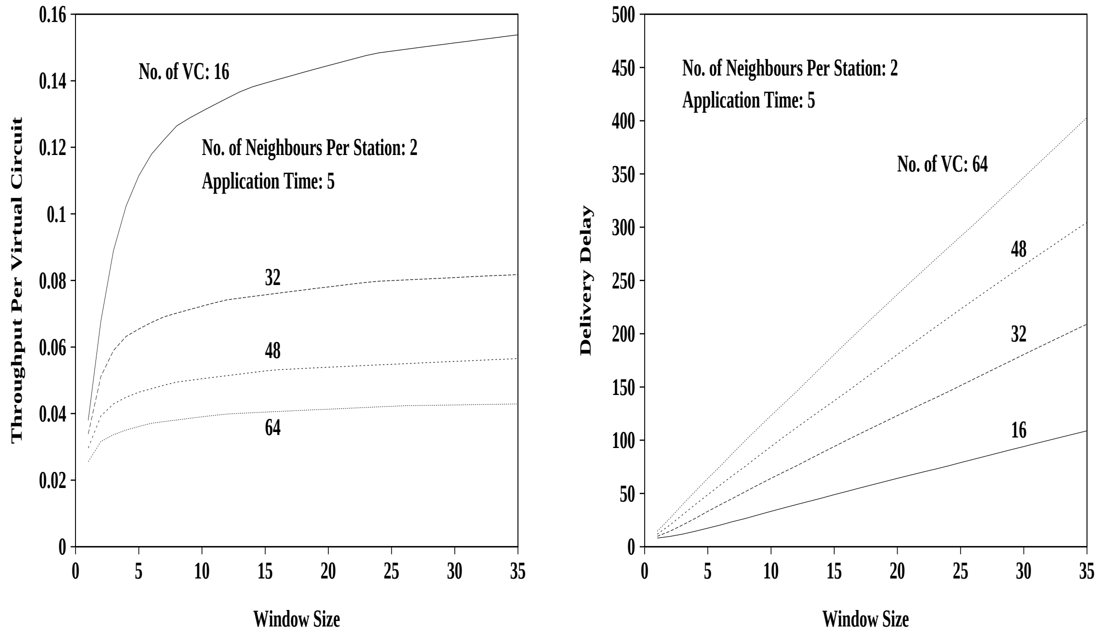


Figure 5.5: Effects of variable traffic density

Figure 5.4 demonstrates the effects of window size variation on the end-to-end network delay and throughput with different radio entity population densities. The delivery delay is defined as the time interval between the instant of departure of a packet from the source mobile unit's application queue and the time of its arrival at the destination mobile application queue. With increasing window size, the packet population within a virtual circuit grows monotonically; the increased queueing delays, caused by this population growth, account for the linear and monotonic increase of the delivery delay with sliding window sizes. The tolerable upper-bound of the delivery delay for a specific application determines the maximum window size that can be used for the flow control scheme. For the smaller values, the enlargement of the window size causes the arrival rate to all the queues in a chain to increase and that, in turn, increases the steady state throughput for the virtual circuit. For the larger window sizes however the increased end-to-end delay causes the inter packet transmission interval at the application queues to increase. This reduces the rate of increase of the chain throughput, and eventually causes the throughput to become insensitive to the window size. As expected, the densely placed radio entities show poor performance, which can be explained from the MAC layer performance graphs, shown in Figure 4.9. The effects of the varying virtual circuit densities on the network delay and the throughput are shown in Figure 5.5. Note that the patterns of the delay and the throughput variations are quite similar (for the same set of reasons) to the previous case.

The following four graphs present the network responses for different values of the application service time of the mobile users. Figure 5.6 illustrates the drastic throughput reduction with increasing application time, in a low virtual circuit density (e.g. for 16 VCs) scenario. For the higher VC densities (e.g. with 64 VCs), however, the reduction of throughput is not that significant. It is to be noted that for the very high response times of the application layer queues ($> 70T$), the throughput becomes almost insensitive to the virtual circuit densities. With increasing application time, the delivery delay reduces and it finally tends to saturate at a value less than $10T$. For the small application time cases, packets are quickly processed by the application layers queues and that, in turn, increases the arrival rate at the data-link layer queues. Increased arrival rate also increases the service delay of the radio data-link layer and most of the packets within a chain tend to accumulate there; this gives rise to a bottleneck at the radio data-link layer of the network. This bottleneck causes the delivery delay to be as high as $60T$ (when the application time is $10T$), for a 64 VC case with a window size of 5. For the slower applications, the delay performance improves significantly because of the reduced burden on the radio data-link layer queues.

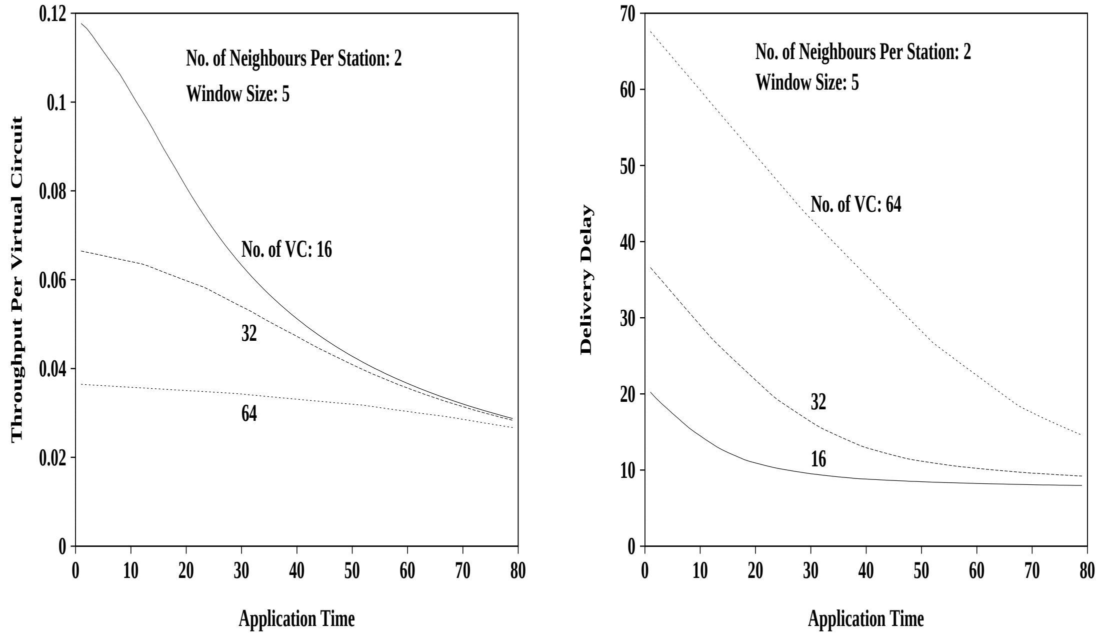


Figure 5.6: Throughput and delay variation with changing application response time

The first graph in Figure 5.7 demonstrates the effects of various kinds of bottlenecks on the components of the network delay. The single trip end-to-end delay is defined as the half of the round trip delivery delay. It includes the delays through the queues 2, 3, 4, 5, 6 and 7 in Figure 5.3. The single trip radio delay consists of the delays through the queues 4 and 6. As the network bottleneck gradually shifts from the radio data-link layer to the application layer (with increasing application time), the single trip radio delay reduces and finally stabilises at a small value. Initially, the rate of increase of the single trip end-to-end delay is very low because of the reducing radio link layer delays. However, for the very slow applications, with stabilised radio link layer delays, the single trip end-to-end delay increases linearly with the increasing application time.

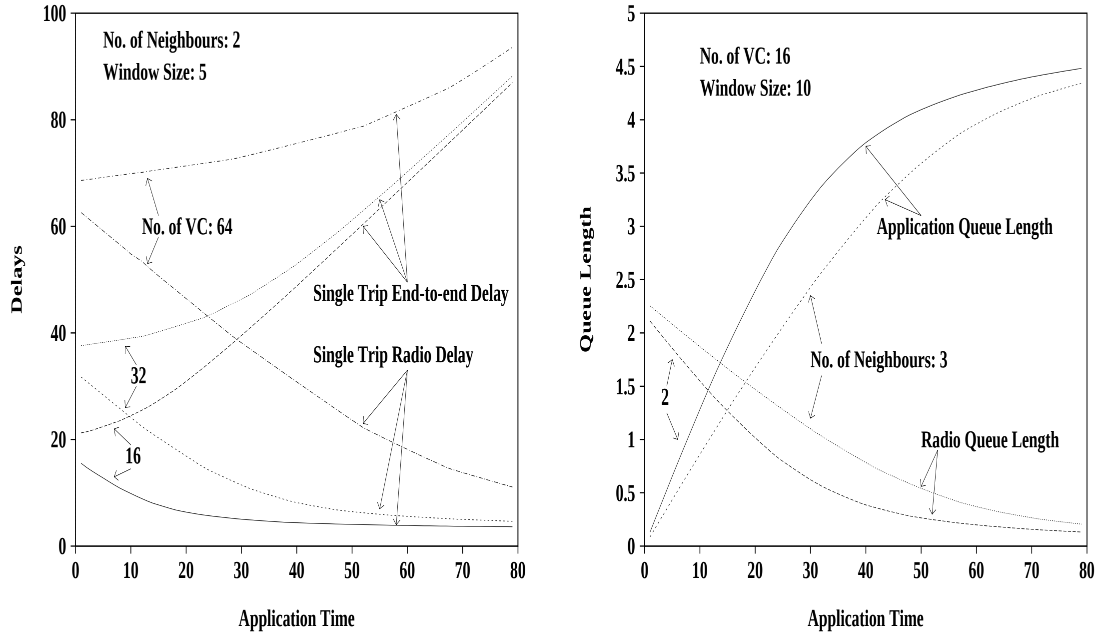


Figure 5.7: Delay breakup and buffer requirements

The buffer requirements at the application and at the radio layer are depicted in the second graph of Figure 5.7. When the network bottleneck is entirely within radio, the data-link layer queues (labelled as 4, 6, 10 and 12 in Figure 5.3) have a maximum length of 2.3 for a station density of 3. With increasing application time, the average radio queue length reduces and finally stabilises at a value less than unity. The application queue length increases as the network bottleneck drifts from the radio to the application layer. However, a fixed window size for each virtual circuit ensures that the application queue length also stabilises for a large value of the application service time.

5.5 Summary

The end-to-end performance of a multihop packet radio LAN, operating under a window based flow control strategy, has been investigated in this chapter. A hybrid modelling technique is used in which the radio data-link layer was modelled using a simulator and the simulation results are used in a higher layer queueing model for the end-to-end performance evaluation. The end-to-end queueing model for each virtual circuit has been solved by using an approximate MVA technique with an augmented iteration scheme, for faster convergence.

The results, in terms of network throughput and end-to-end delivery delay, obtained from this model can be used for choosing the optimum window size, for a given application and the other network specifications. The impact of the mobile unit application service times, on the network performance, were found to be quite significant. It was observed that the increase of the application service time reduces the delivery delay over radio (by shifting the network bottleneck from the radio data-link layer to the application layer), at a cost of reduced overall throughput. Although the model starts with an assumption of infinite buffer length at each queueing node, the results confirmed that the buffer requirements are always limited by the maximum window size.

Although the presented numerical example corresponds to a regular network topology, it is possible to apply the proposed modelling technique for the networks with random topologies. This however requires a more complicated and exhaustive model for the radio data-link layer. Considering the large difference (one order) between the raw switching rates of the backbone and the radio networks, a simplified model of the former has been considered. In the situations where the backbone network speed is comparable to that of the radio, an explicit and more detailed model for the backbone has to be developed.

Chapter 6

Mobile Location Management

This chapter outlines the design of a location server, tailored for a connection-oriented radio local area network environment. It is described how the location information, regarding the mobile users, is efficiently partitioned and exchanged between the network entities in order to strike a balance between the location update and the query latencies. The presented design introduces a *representative* based strategy which offers the tracking functionalities required by the mobile connection management service, described in Chapter 7.

6.1 Introduction

In a ubiquitous computing environment [Harter 94], location information about the mobile users and resources is extensively used by the distributed location dependent and independent applications. This information is also used by the radio network entities responsible for signalling and management operations. To cater for the needs of such location sensitive applications, it is necessary to provide a service which can efficiently manage the location information about the relocatable network entities and resources. Any user, either mobile or stationary, should be able to send queries to the location server. Typical queries [Imielinski 92a] range from simple requests such as "where is the relocatable resource X?" to the more complex ones like "what about the location of all the relocatable resources, with feature Y, in the local administrative domain?"

One extreme approach for locating a mobile[1] entity is to execute a network-wide *paging* [Badrinath 92] operation. This involves broadcasting the question "where are you?" network-wide and waiting for a sensible reply from the relevant entity. The other extreme policy is to inform the location server about any change of location information, whenever the entity relocates itself. When the former strategy suffers from an extremely high search latency, the latter one provides fast query responses at a cost of enormous location update traffic. An intermediate policy is adopted in the current design, where the location infor-

[1]The terms *mobile* and *relocatable* are used interchangeably throughout this chapter.

mation about a mobile entity is optimally distributed in the network to achieve a balance between the location update complexity and the search latency. In order to accomplish improved reliability and search performance, the location server is designed in a distributed fashion where the information about a mobile entity is stored with different degree of accuracy at different location server nodes.

Setting up a transport or network layer connection with a mobile entity involves locating the entity and routing the connection to it. Once the connection is established and communication starts through it, any change of location of the mobile entity may require significant management and signalling overhead [Keeton 93] for maintaining the network layer (for connection-oriented networks) and, in general, the transport layer connectivities. In the current scheme, an intermediate agent for each mobile entity is used for insulating the rest of the network from the effects of its mobility. Such an agent, known as a *representative*, is required to keep a replicated copy of the latest location information of its client mobile entity. Although a representative is relatively less mobile than its client, it migrates when the client mobile moves from one geographical/administrative domain to another. More about the representative migration procedure is described in Section 6.9 and the details about representative based connection management are discussed in Chapter 7.

The following section describes the structure of the radio LAN for which the location server is designed. Sections 6.3 and 6.4 specify the service requirements of the location server and its functional interdependence with the other network services. The design details of the location server, including the search and update procedures, are presented in Sections 6.6, 6.7 and 6.8. Section 6.9 presents the representative management and migration related details of the location server. Finally, an augmentation of the location server, based on the mobility statistics of each individual mobile entity, is proposed in Section 6.10.

6.2 Radio LAN Architecture

A preliminary description of the wireless LAN architecture is presented in Chapter 2. The components of the network, relevant to the location server design, are re-specified in this section.

6.2.1 Network Model

The system consists of a high speed wired network extended with a wireless component. Any computing entity which has a radio link on it and can relocate itself, is termed as a Mobile Unit (MU). An MU always communicates with the rest of the world through the wired *backbone* network; the gateway entities between the radio and the wired networks (See Figure 6.1) are known as Base Stations (BSs). Although it is possible for two neighbour[2]

[2] Any two entities, with physical layer radio connectivity, are referred to as neighbours with respect to each other.

MUs to communicate directly over radio, in order to reduce the higher layer design complexity, such MU-to-MU traffic is always shipped through the backbone network. Apart from the MUs and the BSs, there are some Stationary Units (SUs) in the network which do not actively take part in wireless communication but can be involved in the location management activities.

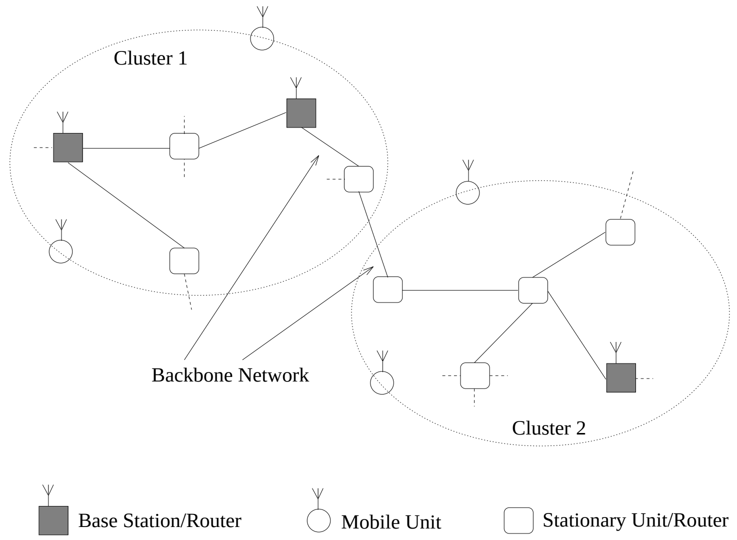


Figure 6.1: Cluster based partitioning of the wireless network

The presented location server is designed for a mobile LAN where a single microwave radio frequency is used by all the base and mobile units as the physical medium of information transportation. No specific assumptions are made regarding the chosen Medium Access Control (MAC) protocol at this stage. Transmission coverage area of a BS is typically referred to as a *transmission cell*[3]. The network consists of large number of small transmission cells, often referred to as pico-cells. The pico-cell profiles (both size and shape) within a building are highly irregular and unpredicatable because of the complicated terrain and attenuation characteristics of the building materials. For a fixed placement of a BS, the corresponding transmission cell profile may even vary with time, depending on the relative location and movement of the objects within the cell.

To mitigate the unreliability problem, caused by this unpredictable pico-cell characteristics, an overlapping cell strategy is introduced where an MU is likely to be seen simultaneously by multiple base stations. Pico-cell *overlapping* is also used as the key component while designing the low-latency handovers, described in Chapter 7. However, in a single carrier frequency scenario the cost paid for this is a reduction of overall frequency reuse efficiency. By carefully adjusting the placement and the transmission power, wireless connectivity between two geographically adjacent base stations can be avoided.

[3]In the current design, the *transmission cell* sizes of the BSs and the MUs are assumed to be equal; this indicates equal transmission power of the base and the mobile entities.

Although the current design principles apply to both datagram and connection-oriented networking environment, a connection-oriented network layer (see Section 6.3), like ATM, is assumed for the materials presented in the rest of this chapter.

6.2.2 Partitioning and Addressing Scheme

In order to reduce the complexity of topology reconfiguration and routing in the backbone network, a single level partitioning scheme was introduced in Chapter 3. The same policy is extended for the wireless world, where the network is partitioned into a group of *clusters* (as shown in Figure 6.1) and each network entity is a member of a specific cluster. The BSs and the SUs in a cluster are expected to be closely placed in terms of physical connectivity and geographical proximity. Each mobile unit is permanently registered with a particular cluster and it shares the same cluster address space with all other entities in that cluster. The MUs of a particular cluster can visit the other clusters, without changing their original cluster of registration. It is most likely that an MU is registered with a cluster into which it spends most of its time. In the current design, it is necessary to change the network layer address of an MU to alter its cluster of registration; this is done by the network administrator, only when a mobile user plans to change, for instance, the working and/or residential place permanently.

The entire network address space is partitioned into a number of cluster address spaces, each of which is subdivided into a mobile-cluster address space and a fixed-cluster address space. All the stationary entities, including the BSs and the SUs, belong to the fixed-cluster address space and the MUs belong to the mobile-cluster address space. Each mobile unit is assigned with a globally unique and permanent network layer address and that is independent of the instantaneous location of the MU. The conventional four octet IP/MSNL [Comer 91, McAuley 90] network addressing scheme is assumed where a part of the address represents the cluster-id and the rest of it represents the host-id[4]. For the fixed-cluster address space the most significant bit of the host-id part is always *zero* and the same bit is always *one* for the mobile-cluster address space. This makes the fixed and the mobile cluster address spaces distinguishable, but equal in size.

With the current addressing policy, it is possible to resolve the cluster address of an entity from its network layer address. The cluster, which an MU is registered with, is known as its **Home Cluster** and the others are referred to as **Foreign Clusters** with respect to that MU. Similarly, all the MUs which are registered with a cluster are designated as its **Domestic MUs** and the ones which are visiting from outside are known as the **Visiting MUs** with respect to that cluster. Because of the overlapping of the base station *transmission cells*, it is often possible that a mobile unit is seen by more than one cluster simultaneously. Although useful for enhanced reliability, this feature of cluster overlapping makes the location server design more complicated and challenging.

[4]Although the term host-id is used, the owner of the identification does not necessarily have to be a host, in strict sense. As a matter of fact, it can be any entity having a network layer identity.

Since each MU has a globally unique network layer address, standard IP name service [Comer 91] is valid for the present network scenario. The *names*, instead of the *addresses*, will be used in rest of this chapter, for identifying the mobile units.

6.3 Service Interdependence

A layered representation of the major wireless network functionalities is pictured in Figure 6.2. Although the wired network services are completely independent of the wireless counterpart of it, the latter has a complicated dependency on the backbone network services, above the data-link layer.

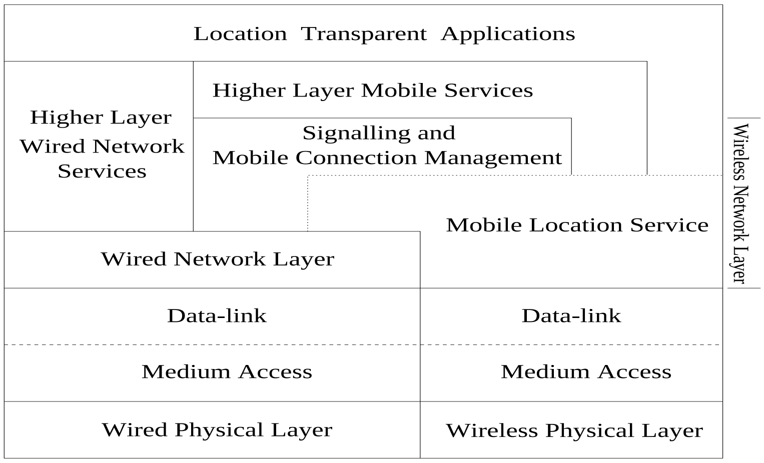


Figure 6.2: Layered representation of the network services

In order to support mobility, the *wireless network* layer is divided into two sub-layers, namely, a *location service* sub-layer and a *connection management* sub-layer. The mobile location service relies on various functions of the *wireless data-link* and the *wired network* layers. The latter is used for transporting location information within the wired network, both during the update and the search operations. It is important to note that the functionalities offered by the mobile location service are used by all the higher layers, including the location transparent applications. Mobile routing, signalling, connection setup/teardown and handovers are the primary functions of the connection management layer which needs support from the location server.

6.4 Required Functionalities

6.4.1 Location Information

The location server is usually contacted to receive the instantaneous location information about a mobile unit. Depending on the spatial granularity, mobile location information is classified into the following categories.

- High Resolution Location Information (HRLI): Contains the address of the base stations which are seeing the relevant mobile unit.
- Low Resolution Location Information (LRLI): Points to the clusters which are currently being seen by the MU.

Apart from the BS addresses, an HRLI entry contains records about the age of the information and the quality of the radio links between the base stations and the relevant MU, at the time of recording. Depending on the location of a querying entity, a location information can be divided into the following partitions.

- Local: Part of the location information which belong to the cluster of the querying entity.
- Remote: The rest of it, belonging to the remote clusters, with respect to the querying entity.

An interested client can contact the location server for a particular amount (either local or remote or both) of location information with a specified degree of spatial granularity. Such a query, regarding a single MU, is designated as a *primary* location query. An example of a *primary* query is when the mobile connection manager asks the location server about the current location of a mobile unit so that the former can successfully route a connection to the relevant MU. Multiple *primary* queries, regarding one or more mobile units, constitute a *secondary* location query. Teleporting [Richardson 94], an intelligent location dependent application, for instance, rely on a *secondary* location query to find out the most suitable X terminal to relocate a user's X session, depending on the current location of the user.

6.4.2 Mobile Connections and Location

Reliable management of a mobile connection[5] requires the location information about its end mobile units during the entire lifetime of the connection. While setting up a mobile connection the source and the intermediate network entities need to know about the location

[5] Any connection with an MU at one end is referred to as a mobile connection.

of the destination MU in order to route the connection successfully. Once the connection is setup, any change of location information of an end MU should be disseminated throughout the network in order to maintain the connection and continue reliable data exchange with the relocating mobile unit. This implies that all the involved network entities have to be aware of the mobility and changing location information of the relevant MU.

A per-mobile unit software agent (known as a *mobile representative* (MR)) is used for partially hiding the effects of mobility from the rest of the network. Any application, interested in transaction with an MU, should negotiate with its representative and the rest is handled by the corresponding MR. This requires a representative to keep complete and up-to-date location information about its client MU. It is illustrated in Figure 6.3 that a connection to an MU always passes through its mobile representative. When the mobile unit relocates itself from one base station to another (from BS_1 to BS_2 in the figure), only the parts of the connection from the mobile unit to its representative are affected. The rest of it remains unaltered, as long as the representative remains stationary. The details about the representative based connection management is presented in Chapter 7.

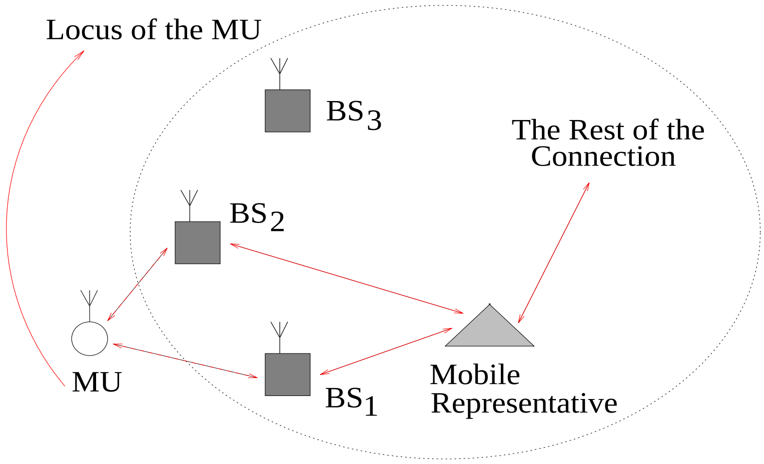


Figure 6.3: Use of *representative* for mobility insulation

The placement and mobility of the representatives are two crucial design decisions. One extreme approach is to nail down the representative process on a fixed network entity, preferably in the home cluster of the client MU. Although it makes mobility of an MU completely transparent to the network, the scheme suffers from the following shortcomings.

- Since each mobile connection has to pass through the home cluster of the destination mobile unit, this may give rise to non-optimal routing and network resource utilisation.
- When an MU is residing quite far from its home cluster, any small change of its location information has to travel a long way before it reaches to the representative. Apart from degrading the update latency, this will produce enormous amount of location update traffic.

- A failure of the fixed host-entity[6] of a representative will cause the termination of all the connections, involving its client MU.

At the other extreme, an MR closely follows its client MU and usually resides within its geographically nearest possible stationary entity. Since the location server has to track the representatives just like their client MUs, while yielding a better update traffic performance, this approach fails to ensure the *mobility hiding* feature of the representatives. An intermediate measure is taken in the current design where an MR is arranged to be much less mobile than its client MU. A representative migration takes place only when its client MU relocates to a new cluster and this requires the location server to keep information about the representative locations. More about the representative migration management is discussed in Section 6.9.

6.4.3 Performance Requirements

The current design aims to achieve the following performance goals.

- Storage Size: The amount of location information, stored in the network, should be as small as possible.
- Query Latency: The location server should reply to a location query with minimum possible latency. Query latency depends both on the communication delay and the location database search time.
- Update Cost: The average network bandwidth usage, incurred by a location update procedure, should be kept minimum.
- Update Latency: The update procedures should be fast enough to maintain the required level of reliability of location information. Reliability is defined as the probability with which a location information indicates to the correct location of the relevant entity. Both the querying entity and the location server should be able to specify the degree of reliability in the relevant queries and replies.

Splitting up the location management functionalities between the wired backbone and the wireless network is another crucial design issue. Considering the scarcity of low-bandwidth radio resources, most of the management operations are implemented within the wired network.

[6] A network entity with sufficient computing resources for running a representative process.

6.5 Mobile Representative

The mobile representatives are software entities, implemented as a collection of independently scheduled threads [Dixon 91], as shown in Figure 6.4. In order to take part in connection management (subject of the next chapter), each MR maintains a database, containing the details of all the mobile connections passing through it. It is also necessary for a representative to maintain an up-to-date version of the location information regarding its client mobile. Figure 6.7 shows the format of the information stored within the location database of a representative. The signalling and management connections are used for exchanging information with the entities involved in location and connection management.

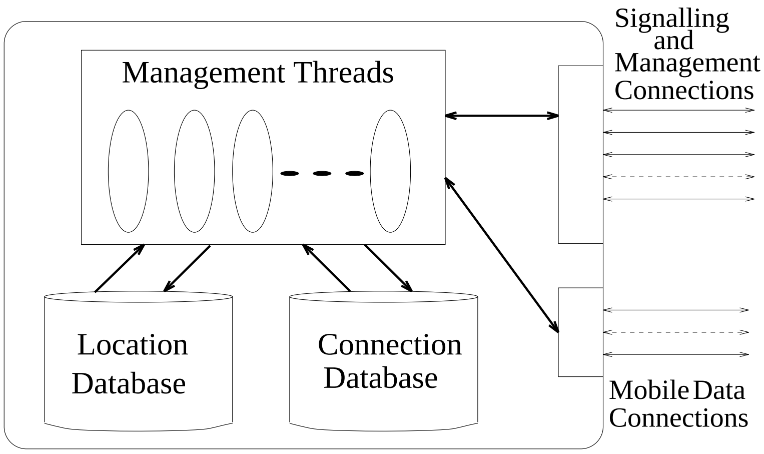


Figure 6.4: Components of a migratory mobile representative

6.6 Location Server Architecture

The location server is designed in a distributed fashion where each cluster requires to maintain a location management entity, known as a *Cluster Location Server* (CLS).

6.6.1 Cluster Location Server

This is a per-cluster management entity and is responsible for the location management activities within its host cluster. The host-id part of a CLS machine address is fixed within the cluster and it belongs to the fixed-cluster address space. This implies that a random entity can always reach to the relevant CLS if the former knows about the address of the corresponding cluster.

In order to exchange location information, each CLS requires to maintain connections with

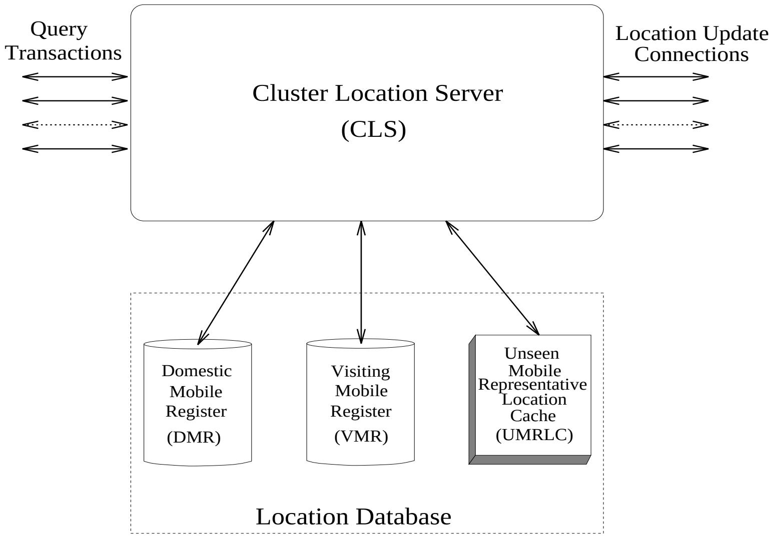


Figure 6.5: Components of a Cluster Location Server

the local[7] base stations and the remote CLSs. Location information in a CLS is stored within three separate databases, namely, the Domestic Mobile Register (DMR), the Visiting Mobile Register (VMR) and the Unseen Mobile Representative Location Cache (UMRLC). More about the databases are presented in Section 6.6.3. When a CLS crashes, usually no location service is available in the corresponding cluster; if a querying entity however knows the address of any of the neighbour clusters, the former can access location service from the CLS of that neighbour cluster. Because of the base stations' *transmission cell* overlapping, it is often the case that an MU is simultaneously seen by multiple clusters. In such situations, crashing of a CLS does not necessarily impair the quality of location service available to the client applications, running on the mobile unit.

6.6.2 Offered Service Interface

A CLS needs to supply information of the following categories.

- Mobile unit location.
- Mobile representative location.

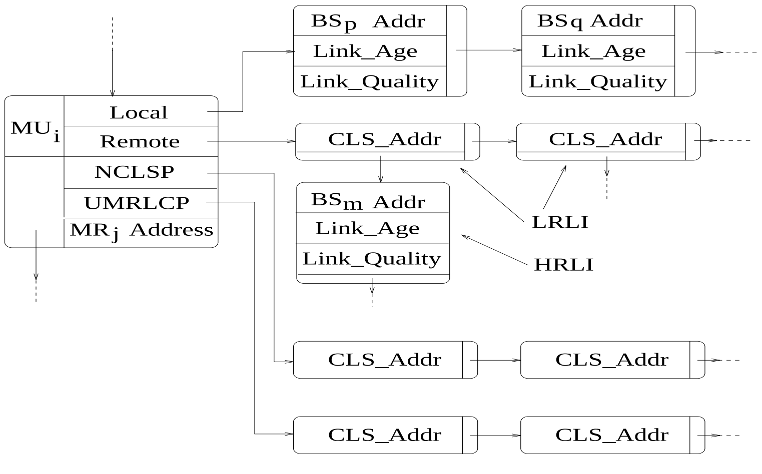
Location of a representative is represented by the complete network layer address of the host-entity in which the representative process is currently residing. An MU location information, on the other hand, can be expressed by the visiting cluster addresses (Low Resolution

[7]From now onwards the term *local* is used for referring to the entities within the same cluster; anything which belongs to some other cluster is indicated as *remote*.

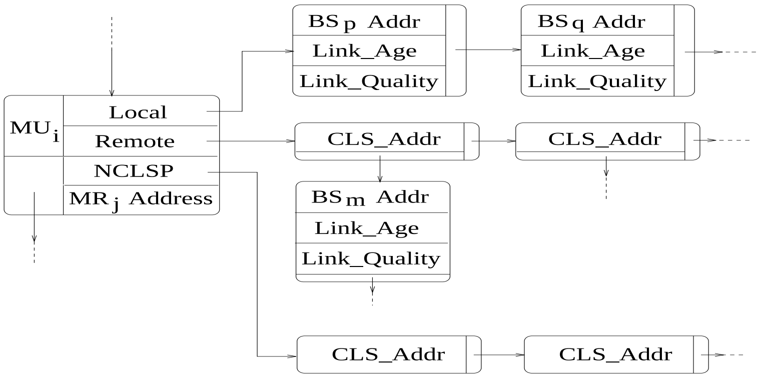
Location Information) or the visiting BS addresses (High Resolution Location Information), as classified in Section 6.4.1. It can further be partioned into the local and the remote components with respect to the location of the querying client.

6.6.3 Location Database in CLS

Figure 6.6 illustrates the format of the location database, maintained within a cluster location server.



(a) DMR



(b) VMR

(c) UMRLC

Figure 6.6: Location database within a CLS

In order to offer the services, mentioned in the previous subsection, the following MU location information needs to be stored within the location databases of the cluster location servers.

- Domestic Mobile Register (DMR) of a CLS keeps complete low resolution location information (LRLI) about all the domestic mobile units of the local cluster.
- If the mobile unit MU_r is currently seen by the local cluster, then the CLS maintains both local and remote high resolution location information of MU_r. If MU_r is domestic then the information is stored in DMR, otherwise in VMR.
- If the host-entity of a representative MR_j is in the local cluster, then the CLS should maintain the complete (both local and remote) high resolution location information of the client mobile unit MU_j. This is necessary because MR_j is required to be informed by the local CLS about the current location of MU_j.
- Depending on the cluster overlapping profile, it is possible for a mobile unit MU_s to see multiple neighbour clusters simultaneously. In that situation, all such clusters are expected to keep the complete high resolution location information of MU_s. Such a cluster's location server CLS_v maintains a list of Neighbour Cluster Location Server Pointers (NCLSP) in its DMR, so that CLS_v can inform the other *co-sighted*[8] clusters, pointed by NCLSP, about the local HRLI of MU_s.

A cluster location server also keeps the following information regarding the current location of the mobile representatives.

- For a domestic mobile unit MU_d, the CLS should keep (in DMR) the current host-entity address of the representative MR_d. This is valid even when MU_d is not residing in its home cluster.
- The address of the current host-entity of the representative MR_v is maintained in the VMR when the client mobile unit MU_v is visiting the local cluster.
- Whenever CLS_k acquires representative location information regarding an *unseen*[9] mobile unit MU_u, from its home cluster CLS_h (more about it is discussed in Section 6.7), CLS_k keeps this information in an Unseen Mobile Representative Location Cache (UMRLC) for future references. After supplying this information, CLS_h adds the address of CLS_k in a list of Unseen Mobile Representative Location Cache Pointers (UMRLCP in DMR), so that the former can notify CLS_k to flush the corresponding entry of its UMRLC, when necessary. Special query-search and update mechanisms are adopted (in Sections 6.7 and 6.8) to reduce the chance of having inconsistency of the UMRLC entries.

[8] Other neighbour clusters which are currently being seen by the relevant MU.

[9] A mobile unit which is neither domestic nor visiting.

6.6.4 Representative Database

One of the prime functions of a mobile representative is to partially hide (see Figure. 6.3) the effects of mobility of its client MU from most of the other network entities. This needs the representative to keep a replica (see Figure. 6.7) of the complete (both local and remote) high resolution information about the location of its client mobile unit. A primary version of it is maintained within the local CLS which, with the assistance from the other relevant CLSs, is responsible for building up the replicated database in the corresponding representative. This approach of replication considerably expedites the representative migration procedure.

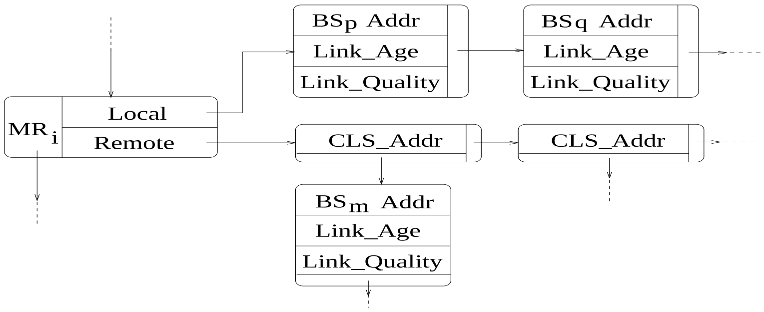


Figure 6.7: Location information maintained by a representative

6.7 Resolving Queries

A client usually sends its location queries to the local cluster location server. However, if the client application is running on a mobile unit which is simultaneously seeing more than one cluster, the application should be able to choose one of the accessible cluster location servers to send its query to. Upon reception of such a query, the CLS first attempts to resolve it by searching the local databases. If the local search fails, it consults with the relevant remote CLSs to acquire the necessary information for the querying client. If a location query cannot be resolved by the local CLS it is assumed that the requisitioned location information is currently not available and the client makes no further attempt to access the information from any other CLS. The query searching procedure is summarised in Figure 6.8; failure of a location query search at $level_i$ causes a CLS to forward the query to $level_{i+1}$. The detail descriptions of the searching procedure are presented in the following subsections. It is to be noted that the degree of location information caching within a client entity is completely application dependent and that is outside the realm of the location server's control.

Since the queries with different combinations of local, remote, high and low resolution location information cause different searching overheads for a CLS, the querying application should clearly specify the amount of information it needs to be supplied regarding the location of the MU of interest. An application, for instance, may be interested only in the local HRLI and the remote LRLI of a mobile unit, but not in the remote HRLI; this may

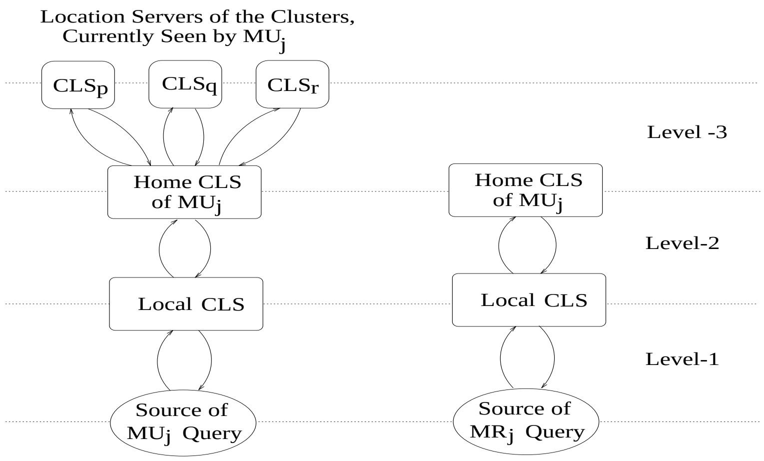


Figure 6.8: Multilevel location query search

require significantly lesser amount of searching than if the complete location information were requisitioned. The following algorithms outline the general searching schemes where the received queries are assumed to have the maximum information requirement. It is a matter of putting additional *'if'*, *'then'* and *'else'* clauses to tailor the algorithms for variable requirement queries.

6.7.1 Mobile Unit Location Query

On reception of a location query regarding MU_j, the location server CLS_h executes the following algorithm. Note that the received query may have originated in the local or in a remote cluster. In the second case, it is actually sent by another CLS which has failed to resolve the query locally.

if (MU_j is a domestic mobile unit){

  if (MU_j is currently residing within its home cluster){
    /* CLS_h should have the complete local and remote HRLI */
    reply to the query with required amount of location information;
    return;
  }

  /* MU_j is not residing within its home cluster */
  if (MU_j is visiting some other cluster){
    /* since MU_j is domestic, CLS_h should have the remote LRLI in DMR */
    send queries to all the CLS(s) of the cluster(s) currently visited by MU_j;
    /* each of those visited CLSs is expected to have complete HRLI */

```
        if (at least one valid reply is received within specified time){
            process the information received from all the CLSs;
            reply to the original query with required amount of information;
            return;
        }
        else{ /* all the queries have failed or timed out */
            reply failure to the original query;
            return;
        }
    }

    /* MU_j is not visiting any other cluster; it cannot be traced */
    reply failure to the original query;
    return;
}
else{ /* MU_j is not domestic; either visiting or unseen */

    if (MU_j is visiting the local cluster){
    /* a local entry for MU_j is found in the VMR of CLS_h */
    /* CLS_h should have the complete local and remote location information */
        reply to the query with required amount of information;
        return;
    }
    /* MU_j is an unseen mobile unit */
    send a query to the home cluster location server of MU_j;
    if (a valid reply received within specified time){
        reply to the original query with required amount of information;
        return;
    }
    else{ /* query to the home CLS timed out or a failure is indicated */
        reply failure to the original query;
        return;
    }
}
```

6.7.2 Representative Location Query

The following algorithm outlines the searching scheme executed by a cluster location server CLS_h when it receives a query regarding the current host-entity address of the representative MR_j.

```
if (MU_j is a domestic mobile unit){
```

```
    if (an entry for MR_j is found in the DMR){
        reply to the query with the address of the host-entity of MR_j;
        if (the query has originated from a cluster location server CLS_n)
            put the address of CLS_n in the list pointed by UMRLCP field of DMR;
        return;
    }
    else /* no MR_j entry is found in DMR */
        reply failure to the query;
        return;
    }
}
else{ /* MU_j is not domestic; either visiting or unseen */

    if (MU_j is visiting the local cluster){
        if (an entry for MR_j is found in VMR){
            reply to the query with the address of the host-entity of MR_j;
            return;
        }
        else{
            reply to the query with a failure indication;
            return;
        }
    else{ /* MU_j is an unseen mobile unit */
        if (an MR_j entry is found in UMRLC){ /* cache hit!! */
            reply to the query with the address of the host-entity of MR_j;
            return;
        }
        else{ /* cache miss!! */
            send a query to the home cluster location server of MU_j;
            if (a valid reply received within specified time){
                put the acquired MR_j address into UMRLC;
                reply to the original query with the host-entity address of MR_j;
                return;
            }
            else{ /* query to the home CLS has failed or timed out */
                reply failure to the original query;
                return;
            }
        }
    }
}
```

Although not specified, both the algorithms require some additional cluster hop-count checks for avoiding the possibilities of cross-cluster infinite query loops.

6.8 Location Update

Most of the location management functions are implemented over the wired backbone network. A mobile unit does little more than broadcasting self addressed and time-stamped location beacons periodically over its wireless link. Each neighbour base station extracts information from those beacons to create and maintain a Base Neighbour Table (BNT), as pictured in Figure 6.9.

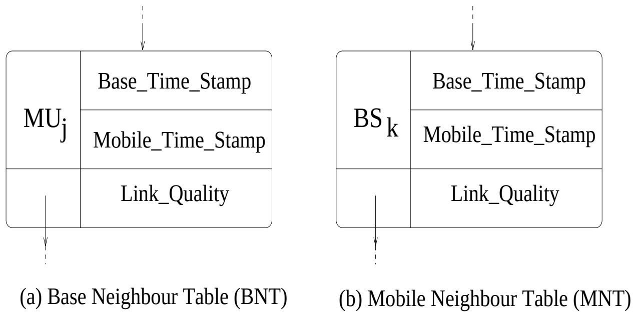


Figure 6.9: Base and mobile neighbour table entries

Similar location beacons are also broadcast by the base stations, which the mobile units use for constructing their Mobile Neighbour Tables (MNTs). This indicates that an MU always knows about its neighbour base stations; in order to find out more about its own location, a mobile unit needs to contact a cluster location server. Apart from the neighbour address, each table entry contains the fields indicating the time when it was last updated. Another field records the quality of the radio link during the reception of the last relevant beacon. All the BNT and the MNT entries time out in order to detect any possible breakage of the radio links. Any significant change of a BNT entry (see next subsection) triggers a network-wide update cycle to create and update the location databases as specified in Section 6.6.3.

6.8.1 Relocation Events

Each one of the following events causes the location server to execute an update cycle.

- The addition or deletion of a BNT entry in BS_t: This indicates an arrival or departure of a mobile unit in the *transmission cell* of BS_t. Since a power switching of a mobile unit causes a location update cycle in the network, it is sensible for an MU to broadcast location beacons even when the unit is temporarily withdrawn from service. This is a tradeoff decision between MU power consumption and the amount of update traffic generated by the location server.

- Significant change of wireless link-quality: During the relocation of an MU within a base station's *transmission cell*, if the quality of the wireless link with the corresponding BS changes by more than a threshold value[10], an update cycle is executed by the location server. Location information with radio link-quality is widely used during the setup of a mobile connection (see Chapter 7) where an appropriate base station is chosen on the basis of the link-quality with the destination mobile unit.

6.8.2 Update Connections

The location update messages generated in a base station always need to be sent to the local cluster location server. This requires a base station to setup and maintain a connection with the local CLS. Similar update connections (see Figure 6.5) are also maintained by a CLS with the other relevant cluster location servers. Soon after booting, a BS sets up an update connection with the local CLS. A cluster location server however sets up such a connection with a remote CLS only when an update message needs to be exchanged between them.

Although no specific assumption is necessary for the network layer protocol, a connection-oriented approach is taken during the implementation of the prototype server, described in Chapter 8. The details of the location update messages and their usage policies are presented in Section C.1 of Appendix C.

6.9 Representative Management

The creation and migration procedures of a representative are supervised by the home cluster location server of the client mobile unit. The cluster, in which an MR would reside, is selected by the home CLS and the particular representative host-entity is selected by the CLS of the chosen cluster. It is an essential criterion that an MU must be seen by the cluster which is chosen as its representative's residence. While choosing an appropriate cluster for MR_j, the home CLS of MU_j uses one of the following possible heuristics; however, these are used only when the mobile unit MU_j is simultaneously seen by multiple clusters.

- Choose the cluster where the mobile unit was seen most recently. This is a useful approach when not much information is known about the MU's *mobility profile*[11].
- Select the cluster where maximum number of base stations are currently being seen by the client mobile unit. This strategy is useful for minimising the average distance between the representative and all the base stations, seen by the MU (see Figure 6.3).
- Choose the cluster which appears to be most suitable from the *mobility profile* of the MU. One policy, for instance, is to choose the cluster which was visited most often during the last few minutes/hours/days.

[10] This can be used as a parameter to control the granularity of location change detection.

[11] Mobility profile is specified [Badrinath 92] as the temporal and the spatial locality information about the movements of an MU.

Once a cluster is selected, the CLS of the chosen cluster considers the following criteria to pickup a host-entity for the representative.

- It is necessary for the host-entity to be located in a fairly central position with respect to the cluster base stations. A variation of this policy would be to give more priority to the BSs which are currently seeing the client mobile unit.
- The chosen host-entity should have sufficient computing and communication resources left for a representative process.

The details of the representative creation, deletion, migration and the corresponding MR location update procedures are presented in Section C.2 of Appendix C.

6.10 Mobility Profiles

A mobile unit is expected to be found, with high probability, within a group of geographical locations. A user, for instance, would like to spend most of his working time within his own office and with few other specific colleagues. Similarly, a mobile unit is likely to be contacted very often, by some specific network entities. These indicate that the location information about a mobile and its representative is used by some clusters more frequently than the others. An augmentation of the location server, based on this spatial locality of the mobile movement and the location queries, can be implemented to improve its performance.

Per-MU Partitioning

Based on the mobility and query profiles, as mentioned above, the following per-MU partitions can be created on top of the scheme described in Section 6.2.2.

- Roaming Partition (RP): RP_j specifies the group of clusters where the mobile unit MU_j spends most of its time.
- Query Partition (QP): The clusters, originating most of the location queries regarding MU_j and MR_j, constitute the query partition QP_j of the mobile unit MU_j.

Based on the mobility statistics of an MU, its roaming and query partitions are dynamically defined by its home cluster location server.

The concept of roaming partition is useful while choosing an appropriate cluster to create a new representative process. As mentioned in Section 6.9, it is advantageous to choose a cluster from the mobile unit's roaming partition, for reducing both the query and update latencies. An augmented location update strategy which attempts to incorporate these newly defined partitioning schemes, is presented in Section C.4 of Appendix C.

6.11 Summary

A distributed location server design, for a wireless local area network environment, was presented in this chapter. A scheme, somewhere between the network-wide paging and the per-move-update, has been adopted to optimise the update and the search costs involved with location queries. The forms of mobility and their manifestations were described, along with the strategies with which the change of location information is selectively disseminated throughout the network. It was argued that by restricting the mobile entity location information within selective geographical partitions, both the update and the query searching costs can be optimised.

The concept of a per-mobile entity software agent, known as a *representative*, has been introduced. Use of such representatives for partial mobility hiding, in a connection-oriented wireless environment, was advocated. The process of migration and location management of the representatives has been incorporated within the design of the generic location server.

A novel network partitioning scheme, based on the mobility statistics of the individual mobile entities, has been proposed. It was argued that the location service can be further augmented on the basis of this new partitioning scheme which improves the query searching performance.

Chapter 7

Mobile Connection Management

A design of the network layer signalling and the connection management protocols for a virtual circuit based radio LAN is presented in this chapter. The presented services are implemented on top of the mobile location server, described in Chapter 6. The details of the network architecture and various services offered by the network, to support mobility, are already outlined in Sections 6.2 and 6.3. The per-mobile unit software agents, known as *representatives*, are used for designing the low latency and scalable handovers which are capable of supporting continuous media traffic. A set of schemes are proposed for supporting resource reservation and Quality of Service (QoS) guarantee in a connection-oriented (e.g. ATM) mobile networking environment.

7.1 Introduction

The purpose of the present design is to provide a ubiquitous computing environment where a user can have access to the location independent information and computing resources. Realtime multimedia applications [Minzer 89] have strict performance requirements such as delay, jitter, throughput and the reliability bounds like cell loss rate. To guarantee these performance limits, the connection-oriented approaches [Keeton 93] are found to be more efficient compared to the conventional datagram based networking. The presented design corresponds to such a connection-oriented network layer which is suitable for the wireless LAN, proposed in this dissertation.

User mobility, as described in Chapter 6, makes the network layer operations complicated, especially in terms of signalling, routing and connection management. When a mobile unit (MU) is *on the move*, the end-to-end connectivity of a transport layer connection involving that MU is maintained by performing a management operation known as a *handover* [Hellstern 93, Keeton 93, Chia 91, Earnshaw 92]; this involves online traffic rerouting and exchange of connection state information between the involved base stations and various other network entities. In the presented management scheme [Biswas 94a], the per-mobile unit representatives are used for supporting the connection setup, tear-down, routing, signalling and the per-connection quality of service guarantees. A low latency handover policy is carefully designed for supporting the requirements of the realtime continuous media traffic. Although a single radio frequency physical layer is assumed for the target LAN architecture (see Section 2.5), in order to maintain generality, no specific assumptions are made about the radio *medium access control* (MAC) layer protocol. An implementation of the current design, in an ATM networking environment, is presented in Chapter 8.

The following section outlines the management functionalities required to be supported by the designed network layer. The targeted performance criteria, for supporting realtime multiservice traffic, are summarised in Section 7.3. Sections 7.4, 7.5 and 7.6 specify the structure of the *multi-segmented*[1] mobile connections and the signalling protocols involving the creation and maintenance of those connections. A comprehensive description of the designed handover mechanism is furnished in Section 7.7. Various error situations and their possible remedies are proposed in Section 7.8. The extensions, necessary for augmenting the current design to support quality of service guarantee, are proposed in Section 7.9.

7.2 Supported Services

The end-to-end services are provided by a transport layer, built on the network layer functionalities, described in this chapter. The primary goal of the current design is to hide the effects of user mobility above the network layer, as much as possible. In order to achieve this, in addition to the other usual wired network services, the wireless network layer[2] offers the following functionalities [Teraoka 93].

- Mobility Transparent Addressing: The upper layers are provided with unique addresses for the mobile units. This implies that an application can unambiguously specify a mobile unit with a globally unique network layer address, which is independent of the MU's physical location. Such an addressing scheme is outlined in Section 6.2.2.
- Connection Setup and Termination: Clean interfaces are provided to the upper layers to setup and tear-down the point-to-point mobile connections[3].
- Mobile Routing: With an aid from the location manager, described in Chapter 6, the network layer provides an efficient mobile routing service.
- Handover: Although this can be seen as a transport layer function, the realtime requirements of a handover operation prompted it to be considered as a network layer service.

[1] A segment is a part of a mobile connection; more about it is described in Section 7.4.

[2] Note that a connection-oriented network layer is assumed in the rest of this chapter.

[3] Any connection with an MU at one end is referred to as a mobile connection.

- Quality of Service (QoS): In a connection-oriented network like ATM, most of the mentioned services need to guarantee QoS while transporting the performance-critical continuous media traffic. This requirement leads to the current design of a *segment caching* technique (presented in Section 7.4) for supporting low latency handovers and QoS guarantee. The statistical nature [Jung 93] of QoS guarantee in the mobile networks gives rise to the requirements for frequent dynamic renegotiation [Crosby 93] of the quality of service parameters during the mobile movements.

The following are the prime driving forces which shape the design of the signalling and mobile connection management schemes.

- Most of the network management burdens are to be put into the wired backbone. This is to use the scarce radio resources as efficiently as possible.
- Considering the inherent unreliability of a wireless medium like radio, most of the network state information are to be located into the wired counterpart of it. This also makes the protocols more immune from frequent failures and disappearance of the mobile entities.
- Since high quality handover is considered to be one of the most crucial requirements, most of the other services like connection setup, routing and call termination are optimised to provide the best possible handover performance.

Most of the radio resource management operations, including medium access, transmit power control and channel quality monitoring, are performed by the lower layer protocols and it is shown later that QoS guarantee by the network layer requires direct support from these lower layer protocols.

7.3 Performance Requirements

The merits of the mobile signalling and connection management scheme can be evaluated with respect to the following criteria [Hellstern 93].

- Service Quality: Indicates to the performance of the above mentioned services, offered to the users by the network. QoS support, scalability and the predictive resource reservation during handovers are few of the example service quality requirements.
- Resource Utilisation Efficiency: Indicates to the efficiency of the network resource usage and balancing of the signalling and management burdens over the wired and the wireless parts of the network. The bandwidth-space-time product [Keeton 93], for instance, measures the network resources (bandwidth, length and duration of a connection) allocated to an inactive segment (see Section 7.4) of a mobile connection. A low bandwidth-space-time product indicates improved resource utilisation efficiency.

7.4 Mobile Connections

In most of the existing connection-oriented network layer handover schemes [Keeton 93, Pollini 92, Chia 91], the new parts of the affected connection are setup during the actual handover process. This involves heavyweight *online signalling*[4] within both the wired and the wireless networks. To reduce this online signalling burden and the performance degradation incurred by it, a segment caching scheme is proposed. Here, the new parts of the affected connection, known as segments, are setup in a predictive manner, even before the handover operation is initiated. Although this approach makes the routing and the handover signalling more complicated, it removes some of the handover performance restrictions and allows QoS guarantee over the mobile connections by reserving resources predictively.

Depending on the end entities, the mobile connections are classified into the following categories.

- Mobile to Stationary: A connection, originated from a mobile unit and with a stationary destination entity.
- Stationary to Mobile: Originated from a stationary entity and destined towards a mobile unit.
- Mobile to Mobile: Both ends of the connection are tied with mobile entities; this is the most general form of a mobile connection.

The components of a mobile-to-mobile connection is pictured in Figure 7.1, where both the end mobile units are simultaneously seen by multiple base stations. A mobile connection can be seen as a set of wired and wireless *segments*, put together to achieve end-to-end connectivity. MU_j-BS_p, MU_j-BS_q, BS_r-MU_k and BS_t-MU_k are the wireless segments of the connection shown in Figure 7.1. Segments BS_p-MR_j, BS_q-MR_j, MR_j-MR_k, MR_k-BS_r and MR_k-BS_t belong to the wired backbone network. Although all the wireless segments are single-hop[5] in nature, the number of hops in a wired segment depends on the relative physical locations of the base stations and the host-entities of the end mobile units' representative processes. Note that each mobile connection passes through the representatives of its end mobile units. Any combination of the segments, providing an end-to-end connectivity, constitutes a *path*. A connection can also be seen as a bunch of multiple overlapping paths where only one of them can be *active* (the one which carries data) at one time. MU_j-BS_q-MR_j-MR_k-BS_t-MU_k is one of the four possible paths in the connection shown in Figure 7.1. Multiple paths are used for the implementation of low-latency handover control scheme, as explained in Section 7.7. It is to be noted that the multiplicity of the edge segments (MU-BS and BS-MR) and the paths in a mobile connection depends on the number of base stations,

[4]The signalling operations which are carried out during the handover procedure; relevant signalling, which takes place not during the actual handover operation, is referred to as *off-line signalling*.

[5]When a segment has to pass through only one data-link layer.

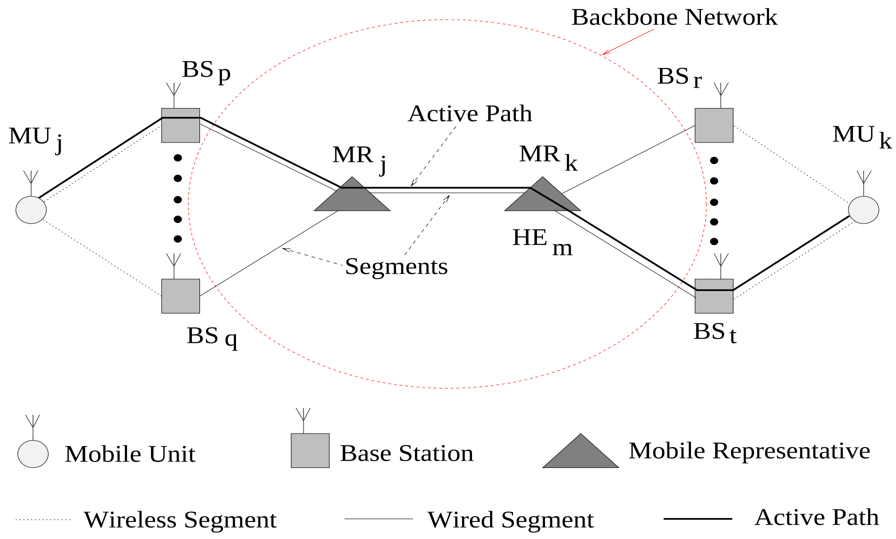


Figure 7.1: Segment based structure of a mobile connection

simultaneously seen by the end mobile units. During a handover, those edge segments are switched in and out of a connection by the MUs and their representatives. These switching operations are coordinated according to the MU relocation patterns from one BS to another.

For a mobile connection, each of the involved MU, BS and MR may have to maintain more than one segment at a time and this information is kept within the connection databases, as shown in Figure 7.2. These databases are created and updated during the mobile connection setup, connection deletion, segment setup, segment deletion and the handover operations. Note that the database shown in Figure 7.2(c) corresponds to the connection information stored within an MR, as pictured in Figure 6.4. It is necessary to identify a mobile connection uniquely within the network so that an entity (MU, BS or MR), involved with the connection, can recognise the relevant segments of it. One way of identifying (see Chapter 8 for more detail) a mobile connection is to use the tuple (*source, sequence-no*), where *source* indicates the network layer address of the source entity of the mobile connection and the *sequence-no* is a source specific parameter, incremented monotonically for each newly created mobile connection. Note that the *Mobile Connection Identifier* and the *Virtual Circuit Identifier (VCI)* are two entirely different entities; when the former is an identification of an end-to-end mobile connection, the latter is used for identifying a single hop virtual circuit [McAuley 90]. A connection segment is uniquely identified (Segment Id.) within an end entity, as a combination of the address of the other end entity (MU, BS or MR), the outgoing network interface identifier and the VCI of the first hop of the segment, as seen by the former. These connection databases are extensively used for the segment management and the handover operations, presented in the following sections.

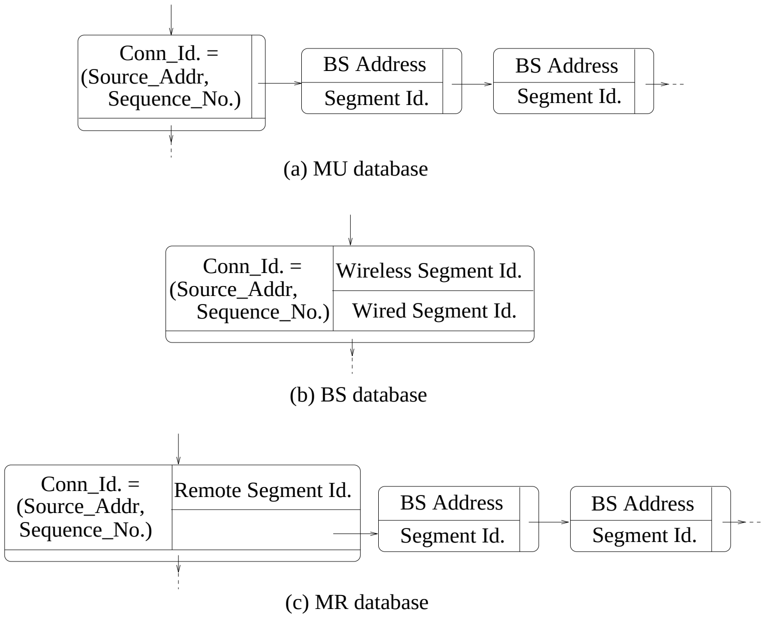


Figure 7.2: Connection information maintained by different network entities

7.5 Signalling

In the current design, both *metasignalling* and *signalling* are used at different stages of the protocol. Metasignalling packets [McAuley 90, Black 93] are messages about signalling, sent on a separate control circuit permanently configured rather than established using the circuit control procedures. Signalling however always takes place through a *signalling connection*, explicitly established between two end entities. Extensive handover signalling between the base stations and the mobile representatives requires them to setup and maintain signalling connections, as depicted in Figure 7.3. The following signalling operations are outlined in this section.

- Connection and segment setup.
- Mobile routing.
- Connection termination.

The signalling, related to segment management and handover, are presented in Sections 7.6 and 7.7.

7.5.1 Connection Setup and Routing

If either of the end mobile units of a mobile connection is simultaneously seen by multiple base stations, the connection may have multiple *paths* associated with it. While creating a new mobile connection, only the segments necessary to construct an *initial path* are setup and data transaction starts through it immediately. If necessary, the other feasible segments of the connection are created afterwards.

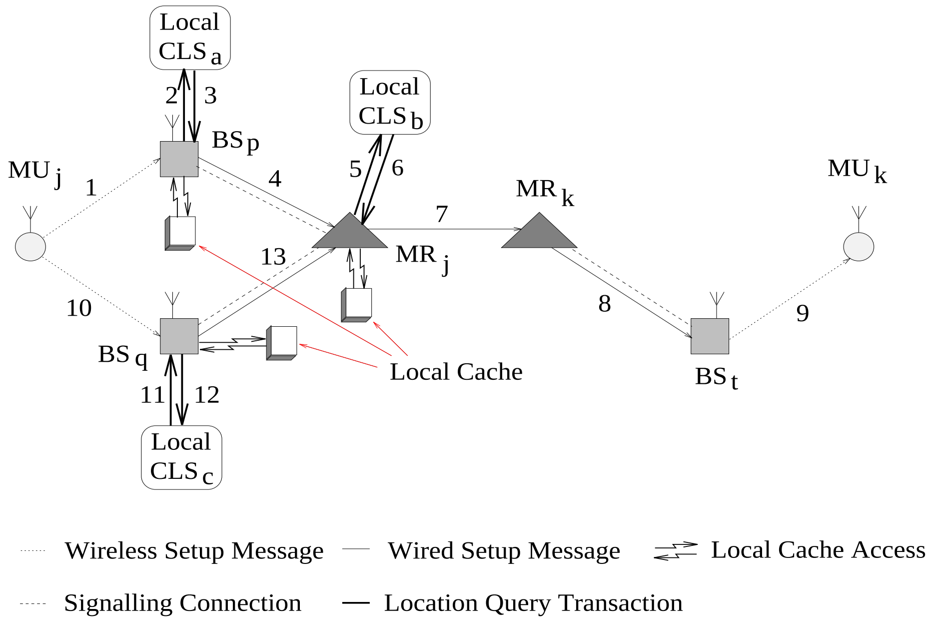


Figure 7.3: Mobile connection setup and segment management

Connection Setup

An example connection setup procedure is pictured in Figure 7.3 and the corresponding signalling sequence is depicted in Figure 7.4. A connection setup from MU_j to MU_k involves the following steps.

1. MU_j searches its MNT (see Section 6.8) and selects BS_p as the most favourable (based on the age and the current quality of the available radio links) base station to establish the initial path through it. The MU sends a metasignalling message (labelled '1' in the diagrams) to BS_p, mentioning the newly assigned identifier of the mobile connection and the VCI for this particular wireless hop.

2. Upon reception of that metasignalling message, BS_p sends a query (message '2') to the local cluster location server CLS_a, asking for the location information of the representative MR_j. CLS_a uses the algorithms, described in Section 6.7, to resolve

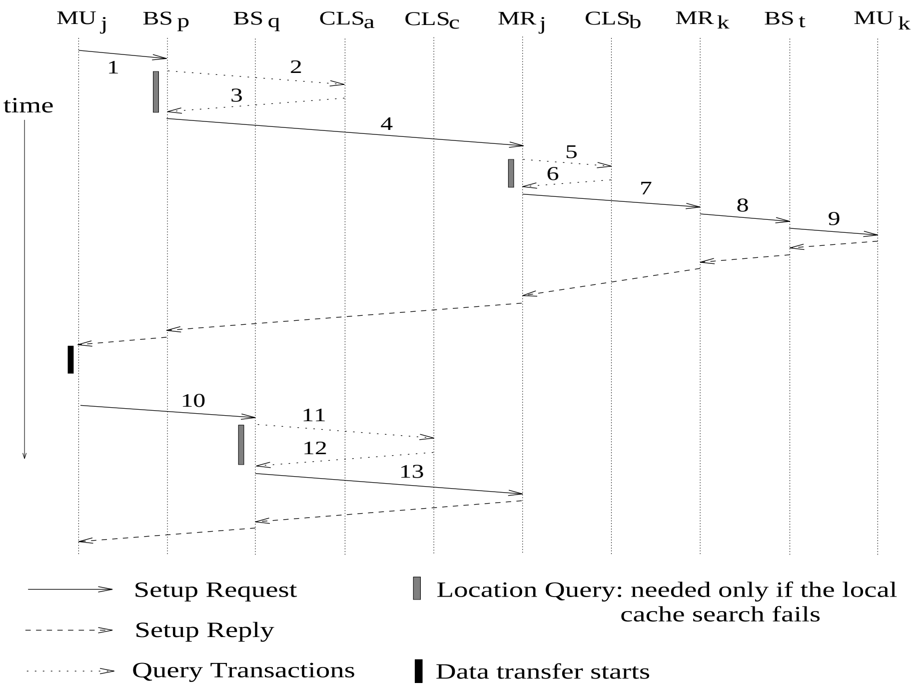


Figure 7.4: Signalling for connection and segment setup

the location query and sends a reply (message '3') back to BS_p. At this stage, BS_p forwards the connection setup request (message '4') to the representative MR_j.

3. The representative MR_j consults (messages '5' and '6') with the local location server CLS_b to resolve the location information about the representative of the destination mobile unit MU_k. On reception of the message '6' with the location information of MR_k, MR_j forwards the call setup request (message '7') towards MR_k.

4. After searching the MU_k location information, stored within the location database of MR_k (see Section 6.6.4), the latter chooses an appropriate base station BS_t and sends the setup message '8' towards it. Again, the age and the quality of the radio links are considered while choosing an appropriate base station.

5. Once BS_t receives the connection setup request (message '8') from MR_k, it tries to complete the connection by forwarding the request (message '9') to the destination mobile unit MU_k.

6. If the mobile unit MU_k is willing to accept the connection, it replies to the request back to BS_t. Similarly, BS_t forwards the reply back to MR_k and this relaying process continues until the reply goes back to the originator mobile MU_j. This completes the setup process of all the segments, necessary for creation of the initial path.

Soon after the initial path is established, the end entities start transferring data through it. The off-line segment setup/deletion procedures are described in Section 7.6. During a segment formation, the relevant MU, BS and MR entities construct and subsequently maintain the connection databases, as shown in Figure 7.2. It is necessary for a BS to have a handover signalling connection with an MR if there exists at least one mobile connection segment between them. This signalling connection is setup (using a standard wired network connection management scheme) by the BS whenever it detects the creation of a segment with the MR.

When the connection destination is a stationary entity (say SU_d; not shown in the figure), MR_j, in Figure 7.3, forwards the connection setup request directly to SU_d without any location query. Similarly, a non-mobile source entity SU_s originates and sends the connection request message to the representative MR_k (SU_s needs to consult the local CLS to know about the location of MR_k); the rest of the setup procedure remains unaltered. These indicate that when one of the end entities is non-mobile, a curtailed version of the described signalling protocol is necessary for the segment and the connection setup operations.

Routing

Unlike the datagram-oriented mobile-IP routing schemes [Teraoka 93, Ioannidis 91], a per-connection mobile routing policy is adopted. This approach is necessary for supporting the performance-critical multiservice traffic in a connection-oriented environment like ATM. Routing within the backbone network (for messages '2' through '8' and the corresponding replies) uses the wired network routing schemes. One such policy is presented in Chapter 3 and it is used for the prototype implementation, described in Chapter 8. The one-hop wireless setup ('1' and '9') and reply messages do not need any explicit routing to be performed.

It is to be noted that a combination of *source* and *hop-by-hop* approaches is used for routing a mobile connection. Source routing is implemented on a per-segment basis whereas within a segment, the connection is dynamically routed in a hop-by-hop fashion. This requires the base station BS_p, for instance, to put the network layer address of the MR_j host-entity in the connection setup message (labelled '4'), for short term source routing. Segment routing from BS_p to MR_j, however, is hop-by-hop and it is dynamically performed by using the backbone network routing policy.

Location Query and Caching

Each connection setup cycle involves at least one location query service to locate the necessary representative process. Two queries are needed when both the end entities are mobile. Messages '2', '3', '5' and '6' represent such location query transactions during the example connection setup phase. In order to reduce this per-connection query overhead, a local location cache is maintained within each BS and MR (See Figure 7.3). Whenever the location information about a representative is retrieved from a CLS, it is put in the local cache and is expected to be used during the subsequent connection setups, involving the corresponding mobile units. This caching scheme reduces the mean setup latency at an expense of increased setup-failure probability, caused by out-of-date cache entries.

One of the various possible flushing schemes is to delete the cache entries based on their age. Another approach requires a cluster location server to inform the local BSs and the MRs about any change of location information of the relevant representatives. While the first one is simple to implement, it suffers from severe cache-incoherence problems in a highly mobile environment. On the other hand, the second approach provides better reliability, but of course at a cost of increased cache maintenance traffic. Any unsuccessful use of an entry also prompts the cache owner to delete it.

Setup Failures

A mobile connection setup fails in the following error situations.

- A connection request or a reply message gets lost in the transit.
- A local cache contains a stale representative location information entry.
- The location service fails because of a crashed cluster location server or any other transient fault.
- The location information, provided by a CLS, becomes inconsistent because of the migration of a representative during a connection setup cycle.
- The network cannot support the quality of service requirements, associated with the connection request. More about it is discussed in Section 7.9.

Except for the last one, all the mentioned errors can be overcome by retrying a failed connection setup with an appropriate *back-off* strategy.

7.5.2 Connection Termination

A mobile connection needs to be closed in the following situations.

- One of the end entities intends to terminate the connection.
- Because of radio link failure or extreme mobile roaming, an end MU goes out of sight of the network.

Unlike the wired network layer connections [Forum 93], the termination of a mobile connection involves the deletion of all its associated segments. When the closure is initiated by an end entity, explicit terminate messages are sent to all the BSs and MRs which are involved with the relevant connection. A mobile representative monitors the states of all the mobile connections passing though it and may initiate a connection deletion cycle if it detects an unusual behaviour (e.g. MU goes out of sight of the network) from its client mobile unit. This improves the reliability of connection deletion and the associated garbage collection over the network.

A deletion also requires the connection databases within the MUs, BSs and the MRs to be cleaned consistently, in order to avoid any resource reservation by the nonexistent mobile connections. Soon after the last segment between a base station and a mobile representative is deleted, the BS terminates the handover signalling connection between them.

7.6 Segment Management

While data shipping continues through the initial path, additional segments are created and destroyed following the mobility pattern of the end MUs. The segment, connecting two mobile representatives, remains unaltered during the entire lifetime of a connection; only the ones, with an MU or a BS as one of the end entities, are created and terminated dynamically. The segment switching and the new path activations are done only during the handover operations. An off-line segment setup procedure, triggered by MU_j's relocation into BS_q's transmission cell, is described here (see Figures 7.3 and 7.4). It is assumed that the initial path MU_j-BS_p-MR_j-MR_k-BS_t-MU_k was already setup before.

1. Mobile unit MU_j sends a segment setup request (message '10' in the diagrams) to BS_q, mentioning the existing mobile connection identifier.
2. If the location information of the representative MR_j is not available within the local cache, then the base station BS_q retrieves (messages '11' and '12') that information from the local cluster location server CLS_c. It is important to note that the cluster location servers CLS_a and CLS_c can be the same, if BS_p and BS_q belong to the same network cluster. Once BS_q comes to know about the address of the current host-entity of MR_j, the former forwards the segment setup message (labelled '13') to the representative.
3. On reception of this setup message, MR_j updates the corresponding connection database and sends a segment setup reply back to the base station BS_q. Similarly, BS_q sends a reply back to the originator mobile entity MU_j. This completes the creation of the new segments MU_j-BS_q and BS_q-MR_j. While doing these, both BS_q and MU_j update the corresponding mobile connection databases with relevant information about the newly created segments.

A mobile connection segment is deleted under the following situations.

- When the corresponding mobile connection is terminated; this procedure was described in Section 7.5.2.

- When the mobile unit MU_j moves out of sight of the base station BS_p, it is necessary to delete the segments MU_j-BS_p and BS_p-MR_j, without affecting the current *active path* of the mobile connection. To optimise the performance however it may be advantageous to defer the deletion for a while, in case MU_j comes back within BS_p's transmission cell soon after it leaves.

While the first type of deletions are controlled by the end entities, a mobile representative initiates a segment deletion procedure, mentioned in the second case. In both the cases, it is necessary to clean the connection databases stored in the MUs, BSs and the MRs. Note that all these segment management procedures are described on a per connection basis. So whenever a mobile unit goes in or out of the transmission range of a BS, the segment management operations are performed by the network for all the existing mobile connections, involving the corresponding mobile entity.

7.7 Handover

As a mobile unit travels from one base station's transmission cell to another, the task of exchanging data between the wired network and the MU must be transferred to the new BS. In the current scheme, this functionality, known as *handover*, is implemented primarily in the network layer although any error at this layer will have to be recovered by the higher layer protocols. The higher layer error recovery depends largely on the type and the performance requirements of the transported traffic.

7.7.1 Design Objectives

The presented network layer handover design aims to achieve the following performance goals, as seen by the end users.

- Low latency: It is necessary to keep the service disruption time minimum during a handover process.
- Data integrity: The network layer protocol should try to minimise packet duplication, loss and out-of-ordering, caused by the handovers. Note that these requirements depend very much on the realtime feature of the applications and their corresponding traffic type.
- Scalability: The protocol should be able to manage a handover, irrespective of the geographical locations of the end entities of the affected connection.
- Reliability: A handover may occur with or without previous *hints*[6]. The protocol is required to meet the aforementioned performance criteria in both the situations.

[6]In a radio network, an MU can take advantage of the overlap between adjacent transmission cells to detect that it is entering into a new cell before it loses contact with its current neighbour BSs. This *hint*

The protocol also needs to be immune from oscillatory handover triggering, caused by intermittent changes of the radio link quality with multiple neighbour base stations.

- QoS guarantee: Although it is very hard to provide deterministic QoS guarantee in a mobile environment, quality of service can be statistically guaranteed by reserving resources predictively, according to the user mobility profiles.

In order to maximise the network resource utilisation, the handover design also needs to maintain the following performance standards.

- Low signalling traffic.
- Reduced handover buffering.
- Minimum possible segment (see Section 7.4) and data redundancy.

7.7.2 Triggering Situations

A handover can be triggered in any of the following situations.

- If a radio link, constituting a part of the *active path* of a mobile connection, breaks.
- When the quality of the radio link between an MU and a BS degrades consistently, the MU may like to handover all its connections to another BS with better radio link with it.
- To balance the load over wire and radio, the network decides to reroute an existing mobile connection.
- The change in radio link quality, in a transmission cell, may prompt the network to hand a connection over to a new BS for maintaining the guaranteed QoS standards.

Except for the first one, the handovers triggered in all the other situations are considered as *with hint* cases.

7.7.3 Protocols

One extreme approach for implementing handover is to setup a fresh connection through the new base station and reroute all the traffic through it; all these happen only *after* the handover process is initiated. The other extreme policy advocates creating redundant

allows the MU to setup new connection segments, before the actual path switching takes place. The term *hint*, in the current context, was first used in [Keeton 93].

end-to-end connections through all the neighbour base stations of an end mobile entity; these redundant connections are created *before* the actual handover takes place. After the handover process is initiated, the only task that remains is to choose the proper connection and reroute the traffic through it. While the first approach suffers from large handover latency (caused by the end-to-end connection setup delay) and, subsequently, enormous buffering requirements, the second one needs to reserve too much of excess network resources.

In order to strike a balance between handover latency and resource consumption, an intermediate approach is adopted where the necessary parts of a mobile connection, known as the *segments*, are created in advance; once a handover is initiated, the relevant segments are switched in and out of the mobile connection to maintain a healthy end-to-end *path* at any point of time. Unlike the extreme approaches, this scheme needs careful considerations regarding packet loss and out-of-ordering.

Although a *with hint* handover is explicitly initiated by one of the end mobile units, in case of a *without hint* handover, triggered by a radio link failure, all the involved mobile, base and the representative entities jointly initiate the handover. For a typical mobile connection, as shown in Figure 7.3, a handover caused by MU_j does not affect the connection beyond the representative MR_j. This feature makes one end handover completely independent of the other end, and so both the end entities can have simultaneous handovers[7] by executing the same algorithm, as presented here.

With Hint

The segment switching and the corresponding signalling sequence during a *with hint* handover process, initiated by MU_j, is pictured in Figure 7.5. It is assumed that the current *active path* of the mobile connection is passing through the base station BS_p and the segments j and l, involving BS_q, were already setup using the off-line segment management procedure, described in Section 7.6. MU_j continuously monitors the quality of the radio links for both the segments i and j. When it realises that the link quality of the segment j is getting consistently better than that of the segment i, the mobile assumes that it is approaching towards BS_q and it initiates a handover operation. The following steps are necessary to complete the handover procedure.

1. Soon after MU_j decides to initiate a handover, it stops sending traffic through segment i and puts the user data in a temporary buffer; however, it continues receiving through segment i. The mobile unit also sends an *initiate* message to the newly chosen base station BS_q.
2. When BS_q receives this message, it is expected to have both the segments j and l setup by the off-line segment management procedure.

[7]This is necessary when both MU_j and MU_k relocate themselves concurrently.

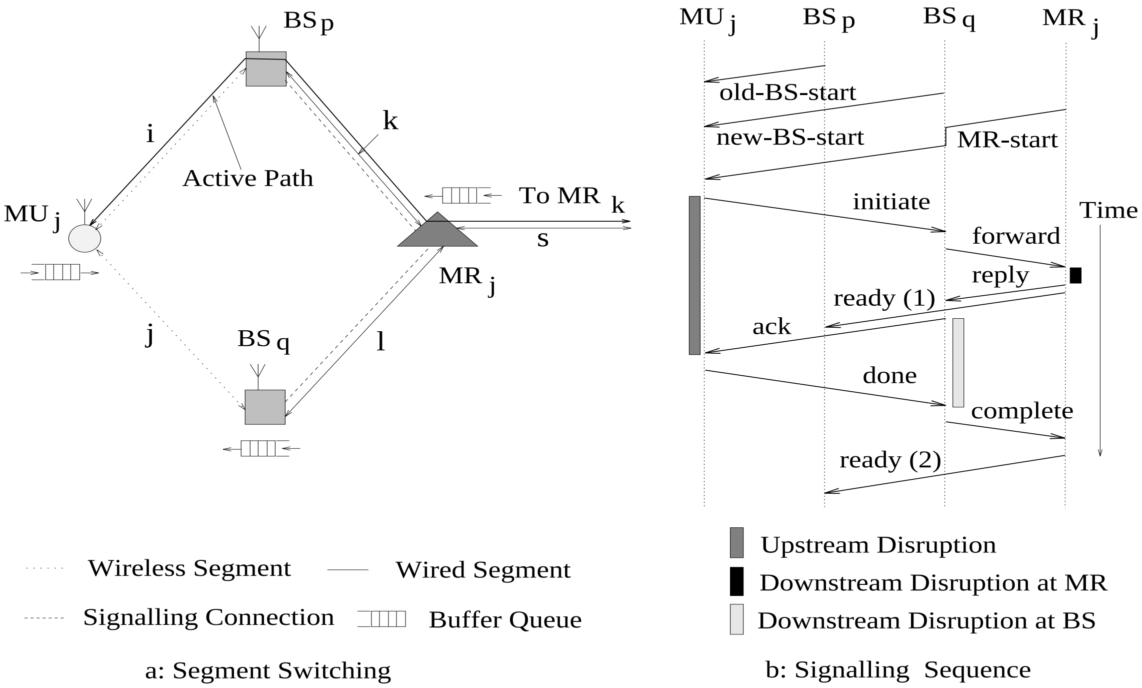


Figure 7.5: Signalling for a *with hint* handover

If those segments are not already setup, BS_q initiates the necessary segment management operations to set them up before proceeding any further with the ongoing handover process. Once the segments j and l are ready, BS_q puts a buffer in between l and j so that the traffic received through segment l can temporarily be stored into the buffer before dispatching it to the mobile MU_j. Until an explicit message (*done*) is received, the data stored in this buffer is not sent through segment j. It is to be noted that the buffer is only for one way traffic and no queueing is necessary for the stream from j to l. Once the buffer is properly set, BS_q sends a *forward* message to representative MR_j through the pre-established handover signalling connection.

3. Upon reception of the *forward* message, MR_j breaks the existing path by disconnecting segments s and k and constructs a new one by connecting segments l and s. Both way data forwarding through MR_j is interrupted only during the switching from segment k to segment l; the normal communication is resumed soon after the new path is activated through the segment l. After this segment switching and the new path activation is over, MR_j replies to BS_q with a *reply* signalling message.

4. BS_q simply forwards this to MU_j as an *ack* message. Note that BS_q is currently buffering all the traffic received through segment l but forwarding the data (if there is any) received through segment j.

5. Once MU_j receives the *ack* message, it stops receiving through segment i; then it clears the buffered data by sending it through segment j and starts transmitting and

receiving regularly through this segment. Afterwards, MU_j declares the completion of the handover operation by sending a *done* message to BS_q.

6. When BS_q receives the *done* message, it clears the queued traffic through segment j and starts regular data forwarding from segment l to segment j. The base station also sends a *complete* message to the representative MR_j, informing the latter about a successful completion of the handover procedure.

7. Finally, MR_j sends a *ready* message to the old base station BS_p so that the latter, depending on the quality of its radio link with MU_j, can either delete the segments i and k or it can prepare for future handovers. Two possible *ready* messages are indicated in Figure 7.5(b). Although *ready(2)* ensures better reliability, it increases the overall duration of the handover process.

When a *with hint* handover has to be initiated because of network load balancing or quality of service maintenance (see Section 7.7.2), it is more likely that a wired network entity, instead of an end MU, would initiate it. If one of the base stations (BS_p or BS_q) decides to initiate the process, then it sends an *old-BS-start* or a *new-BS-start* message to MU_j (see Figure 7.5), depending on who is originating it. Similarly, after taking a handover decision the representative MR_j sends an *MR-start* message towards the mobile unit MU_j. Upon reception of any one of these start messages, the MU performs a handover as described above.

Without Hint

The message descriptions for a handover, caused by a sudden radio link failure, is illustrated in Figure 7.6. A substantial part of handover signalling is common for the *with* and the *without hint* handovers. The following additional signalling steps are necessary for handling a *without hint* handover.

1. When a failure of the radio link with MU_j (see Figure 7.6) is detected by the base station BS_p, it sends a *failure inform* message to MR_j. Additionally, the base station initiates the deletion operations for the segments i and k.

2. On reception of this message, MR_j stops forwarding traffic through segment k and the corresponding data is buffered within a local queue. The data received through segment k, however, continues to be forwarded through segment s. From this point onwards MR_j acts very much like as in the *with hint* situation. The only difference is that when MR_j receives a *forward* message from BS_q, along with the other usual segment switching activities, it clears all the queued data and forwards it through segment l.

3. On the other side, upon detection of the link failure, MU_j starts the handover by sending an *initiate* message to the newly chosen base station BS_q. Unlike for the

with hint case, MU_j stops both transmission and reception through segment i and immediately starts reception over segment j.

4. The base station BS_q handles the *initiate*, *forward*, *reply* and the *ack* messages in the same way as described in the *with hint* situation. The only difference to be noted is that BS_q does not have to provide any buffering for the MR_j-to-MU_j traffic. This is because of the immediate reception switching of MU_j from segment i to segment j, mentioned in the last step.

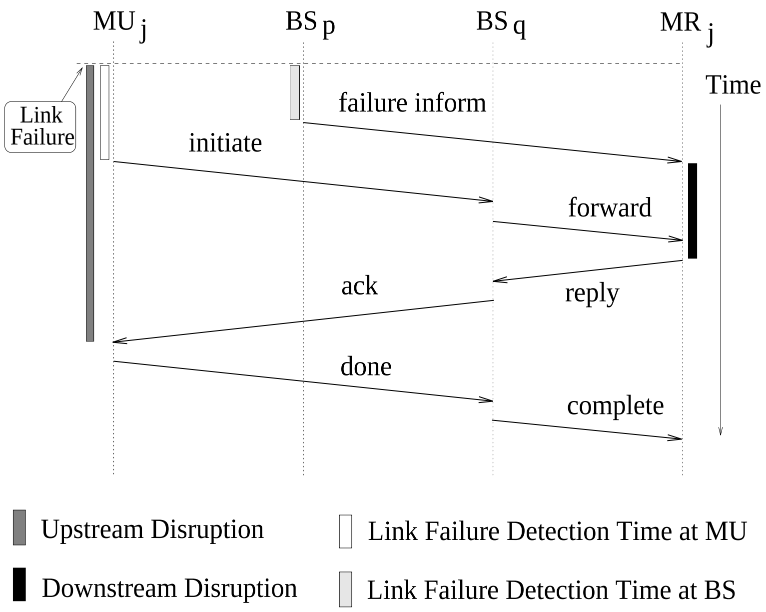


Figure 7.6: Signalling sequence during a *without hint* handover

Since BS_q does not do any traffic buffering, the *done* message is not really important in the current handover operation. This and the *complete* messages are optional and used only to improve the end-to-end consistency of a *without hint* handover. Since both the segments i and k are already deleted by the old base station BS_p, there is no necessity for MR_j to send a *ready* message to BS_p for updating its handover state (see the signalling for *with hint* case). It is worth noting that both the *upstream*[8] and the *downstream* handover latencies depend on the time required by the mobile and the base entities to detect the radio link failure. Reduction of handover latency therefore requires frequent broadcasting of the location beacons (see Chapter 6) in order to enhance the resolution of the radio link state detection procedure. Although the newly chosen base station does not contribute towards the downstream disruption, the MR counterpart of it is quite high because of the link failure detection delay and the *failure inform* message transit time.

[8]Represents the traffic from an MU to the wired backbone network. The reverse direction is referred to as *downstream*.

There exists an extreme situation where a radio link on the *active path* fails when no previously setup segment is ready for immediate switching. Considering the extremely low occurrence probability of such situations, under the present framework of dense transmission cell overlapping and the segment caching scheme, no signalling provisions are made in order to recover from those situations.

Packet Loss and Out-of-order

Unlike in the wired virtual circuit networks, there are potential chances of packet out-of-ordering during the handover of a mobile connection. AAL 5 [Leopold 93] does not specify any mechanism for recovery from the out-of-ordered ATM cells within a block. If the cells are out-of-ordered or lost, the block CRC will fail and depending on the corresponding traffic type the higher layer protocols will have to recover from it. This requires the network layer handover operation to take special care for minimising packet loss and out-of-ordering.

Both upstream and downstream packet out-of-ordering and loss are primarily caused by poor radio links and their unpredictable data-link layer delays. In case of a *with hint* handover, loss and out-of-ordering may occur in the following situations.

- Upstream: If the segment switching (from k to l, as specified in step '3' of the *with hint* handover procedure) by MR_j takes place before the last few packets, transmitted by MU_j through segment i, arrive to MR_j, those un-arrived packets will be lost; this happens when the poor quality radio link of segment i delays the arrival of those packets. Once arrived, any attempt to send them over segment s results in packet out-of-ordering.
- Downstream: After the segment switching is over, MR_j forwards all the downstream traffic to MU_j through segments l and j. If the last few packets forwarded to MU_j through segments k and i arrive at MU_j after the mobile unit has switched over to segment j for regular transmission and reception, those lately arrived packets will be lost. This is again caused by the large delay imposed by the gradually degrading radio link of segment i.

The packet loss problem is much more severe in the case of a *without hint* handover. Since MU_j needs a finite time to detect a radio link failure, it transmits quite a few packets over the non-existing radio link even after the physical breakage of the link. The number of packets, lost during this period, again depends on the granularity of the link failure detection procedure. Similarly, the downstream packets, forwarded by MR_j during the period between the actual link failure and MR_j's reception of the *failure inform* message, will be lost. The current handover protocol does not attempt to prevent these losses and it is assumed that, if needed, the higher layer protocols will arrange to recover the lost packets.

Performance Issues

In order to reduce the *online* signalling overhead of the handovers, a segment caching scheme is used where the segments of a mobile connection are predictively created and destroyed according to the end mobile movements. This helps to reduce the amount of *online* signalling (see Figure 7.5 and 7.6) and the latency of a handover. The possibility of packet duplications during a handover is avoided by not using the inherent broadcast facility of the radio medium; this however brings down the wireless resource utilisation efficiency. When a mobile unit relocates within the transmission cell of a new base station, there may be some registration delay depending on the specific MAC layer protocol and its reservation strategy. This delay worsens the handover performance by degrading the segment setup latency. Considering this, the single frequency based non-reservation type protocols (e.g. CSMA/AED [Biswas 92]) would outperform the reservation based strategies (e.g. dynamic TDMA [Raychaudhuri 94]) under the current handover scenario.

It is also to be noted that the current protocol does not distinguish between the inter and the intra cluster handovers; the only difference, seen by the applications, is the increased latency because of the larger geographical separations between the signalling entities in an inter cluster situation. However, the inter cluster location and the representative migration management operations are much more complicated compared to the intra cluster ones (see Chapter 6).

Isolation of the handover signalling at both the ends of a mobile connection makes the protocol scalable. Since the representative migration policy, in Chapter 6, ensures that an MR does not migrate as long as at least one mobile connection is passing through it, the handover signalling never gets affected by a non-stationary representative process.

It is important to note that MU_j, in Figures 7.5 and 7.6, exchanges all the handover signalling messages only with the new base station BS_q. Better statistical radio link quality of segment j compared to that of segment i (possibly, this is why the handover was initiated), improves the reliability of the scheme.

Handover Errors

A signalling message loss during a handover may lead to a connection maintenance failure. An initiator mobile unit is allowed to abandon an ongoing handover process, based on a specific *timeout* value. After sending an *initiate* message if the MU does not receive an *ack* back within a stipulated time period, it terminates the handover. Depending on the specific traffic performance requirements, it is possible for the MU to retry the handover through another suitable base station. Apart from the enhanced MU buffering requirements, such a retry scheme needs the other involved entities, like the base stations and the representatives, to keep the handover state information for each mobile connection passing through them.

When the mobile unit MU_j sees both the base stations BS_p and BS_q with almost equal clarity (see Figure 7.1), there exists a potential problem of oscillatory initiations of handovers, triggered by the intermittent changes in wireless link quality. To avoid this, it is necessary to implement a carefully chosen handover initiation policy. A *skeptic* based initiation strategy [Schroeder 91] is suitable for solving this problem, where a handover decision is based on not only the current radio link qualities but also on their statistical performance history.

7.8 Error Situations

The connections in a mobile environment are often susceptible to some errors which are normally absent in a fixed wired network. Few such error situations and their possible remedies are discussed in this section.

Mobile Departure: When a mobile unit goes out of sight of the network, the corresponding representative and all the associated signalling and data connections are removed from the network. This also requires tidying up and release of all the occupied network resources.

Setup Failure Due to Mobility: While setting up an initial path of the mobile connection MU_j-MU_k, through base station BS_p (in Figure 7.3), if the originator mobile unit MU_j goes out of BS_p's transmission cell, the setup-reply message will not be delivered to MU_j by the network. To avoid a setup failure in this situation, it is possible to send the setup-reply through a new base station which is currently seen by MU_j. Soon after receiving a setup-reply message from MR_k, if the representative MR_j realises that the connection originator MU_j is no more seen by the base station BS_p then MR_j redirects the reply message through a newly seen base station BS_q. It causes the initial path to be MU_j-BS_q-MR_j-MR_k-BS_t-MU_k. This recovery process requires some additional signalling to be done between MR_j and BS_q in order to exchange the relevant VCI information. A similar situation may also arise if the destination mobile unit relocates itself during an incoming mobile connection setup procedure.

Base Failure: Whenever a base station crashes, if there was an *active path* of a mobile connection passing through it, the corresponding end mobile unit needs to hand the connection over (as in the *without hint* case) to another suitable base station. Similarly, all the *passive* segments through the crashed BS will have to be deleted by the segment management service. If the crashed base station happens to be the host-entity of a mobile representative, the location server has to create a new representative, following the procedure described in Chapter 6.

Mobile Representative Failure: This terminates all the mobile connections, involving the client mobile unit. Any new connection setup to and from the mobile unit needs to be postponed till the location server detects this failure and re-creates the crashed representative somewhere else.

Location Server Failure: Although the failure of a cluster location server does not affect the existing mobile connections, it may prevent the new connections from being setup within the local cluster. Mobile representative migration procedures are also affected by a location server crash.

7.9 Quality of Service

Although ideally the wireless networks should provide the same QoS interface as the wired networks, in the context of mobility it is necessary to add and modify the existing quality of service parameters which are originally defined for the wired networks. After presenting the modified definitions, this section discusses about few design issues regarding QoS provisions in the present radio network environment.

7.9.1 Definitions

As defined in [Leopold 93], the following parameters are necessary for the characterisation of the QoS requirements in a wired network.

- Datarate and Traffic Characteristics.
- End-to-end Delay.
- Variation of Delay (jitter).
- Various Error Rates.

The mentioned errors include bit error, cell loss, cell insertion and a residual error frame rate of the AAL services. The network can guarantee these parameters in a *deterministic* or in a *statistical* [Kurose 93] manner. The following additions and modifications are proposed for the definition of quality of service in a mobile environment.

- Setup Latency: Unlike in a fixed network, mobility of the end entities may cause enormous connection setup delay caused by the error situations, described in Section 7.8. This gives rise to the requirement for the end entities of a mobile connection to specify the *setup latency* as a QoS parameter. This is particularly useful when a mobile unit, initiating a connection while seeing multiple base stations, realises that it is better to try through multiple BSs rather than waiting too long for a connection to be setup through a single base station.
- Mobility: A mobile connection originator should be able to specify its expected degree of mobility during the lifetime of the connection. Mere mention of the non-mobile[9] feature of a connection can be very useful for the network to utilise the resources efficiently, by, for instance, not creating any redundant segments for future handovers.

[9] When the end MU is not expected to move during the lifetime of a mobile connection.

The connections, expecting handovers, can further specify the following QoS parameters.

- Degree of Mobility: Helps the network, a great deal, to decide about the amount of redundancy in the reserved resources (e.g. segments, buffers, etc).
- Handover Latency: Depending on the class of traffic, a user may like to indicate the upper bound of the service disruption time, caused by the handover procedures.
- Blocking Probability: Indicates to the permissible probability of failure of a connection during a handover. The possible error situations are outlined in Section 7.7.3.

• Dynamic QoS Control: It should be possible to dynamically change the QoS parameters of an end-to-end connection [Leopold 93], using a suitable signalling protocol. Insufficiency of the available resources during a handover may prompt the relevant application to renegotiate the connection QoS parameters with the network. Because of the fast and statistical nature of the wireless network resource utilisation profile, dynamic QoS control is a more pressing requirement in a wireless network than its wired counterpart.

7.9.2 Providing QoS

Per mobile connection resource allocation needs support from both the wired and the wireless networks. A base station usually acts as the radio resource manager within its transmission cell; all the virtual circuits within the cell have to be admitted by the manager base station, depending on the current usage of the total available radio bandwidth. This requires support from an efficient MAC layer protocol with precise control on the bandwidth sharing among the participating radio entities. Although high transmission cell overlapping facilitates the low latency handovers, it poses restrictions, in terms of the control over the bandwidth distribution, on the MAC layer protocols.

While setting up a mobile connection, if an MU can see more than one base station, it needs to choose one of them, based on the QoS requirements of the connection being setup. This requires an MU to store the radio resource usage statistics, for all its neighbour base stations, into its Base Neighbour Table (BNT) (see Section 6.8). In the given situation, one of the available base stations is chosen, based on the QoS requirements and the stored usage statistics. Except for the final one, the setup procedures of all the wired segments of a mobile connection (see Figure 7.3) are independent of the radio resource usage within the transmission cells of the involved base stations. While setting up the last wired segment (MR_k-BS_t), the mobile representative MR_k faces a BS selection decision if the destination mobile MU_k is simultaneously seen by more than one base station. This again requires MR_k to store the radio bandwidth usage statistics, for the transmission cells of the BSs seen by MU_k. If any of the intermediate chosen entities fails to support the required QoS, it denies the connection with a negative reply. Depending on the specified upperbound of the setup-latency, a different route (if available) can be chosen by the previous entity, which has received the negative reply.

The segment caching approach, for supporting low latency handovers, gives rise to the requirements for predictive resource allocation. When the *active path* of a connection passes through a base station BS_p, it is useful to predictively reserve some bandwidth within the transmission cells of the geographical neighbour base stations of BS_p. Each such BS may reserve some resource for the unforeseen mobile connections which may be handed over from the neighbour base stations. It will also be useful to have a bandwidth allocation priority scheme where the existing connections are in a privileged position, compared to the newly setup ones.

7.10 Summary

A design of the network layer protocols, for a mobile radio LAN, was presented in this chapter. The per-mobile software entities, known as *representatives*, are used for handling user mobility during the setup and maintenance of a mobile connection. For convenient handover manipulation, each end-to-end connection is split into multiple segments and a segment caching scheme has been introduced for achieving reduced latency and improved data integrity during a network layer handover operation. The corresponding signalling protocols were presented with a description of the possible errors and their recovery strategies.

A suitable combination of *source* and *hop-by-hop* routing was shown to be an elegant way to implement the proposed segment caching policy. The interaction between the distributed location server (see Chapter 6) and the current connection manager, in order to create and maintain the mobile connections, has been presented. Finally, a set of quality of service parameters were redefined in the context of a mobile ATM environment and few schemes, for incorporating these modified QoS concepts into the current design, were proposed.

Chapter 8

A Prototype Implementation

This chapter describes the implementation details of the location and the connection management schemes, proposed in Chapters 6 and 7. An ARM-60 RISC processor based ATM *switch*[1] is used for constructing a point-to-point backbone of the wireless LAN. The same switches are also suitably configured to emulate the base and the mobile entities. WANDA [Dixon 91], a locally developed micro-kernel, is used as the software platform for all the described work. Experimental programmes, for evaluating the system performance, are outlined at the end of this chapter.

8.1 Introduction

The proposed location and connection management schemes are implemented in a high speed ATM networking environment. Olivetti four port ATM switches, running the microkernel WANDA, are used as the generic hardware resources for the current system implementation. The switches, connected by 100 Mbits/s full-duplex point-to-point links, provided an ideal platform to construct the backbone part of the wireless LAN, described in Section 6.2. The protocols, described in Chapter 3, are used for dynamic topology reconfiguration and routing within the wired backbone.

Switches, with ports configured (by software) as radio interfaces, are used as the base and the mobile entities where the software configured ports are programmed to emulate radio connectivities on them. The mobile services are built on top of the communication protocol, known as Multi-Service Network Architecture (MSNA) [McAuley 90]. In order to maintain portability and implementation simplicity, most of the mobile functionalities are implemented as user space processes.

The following section outlines the system environment in which the proposed mobile services are implemented. The hardware, used operating system kernel and the communication

[1] A bus-based 4x4 ATM switch, designed and developed at Olivetti Research Ltd., Cambridge.

protocols are discussed in that section. Sections 8.4.1 outlines the implementation details of the mobile location server and the associated *representative* management scheme. Mobile connection management service development is presented in Section 8.4.2. The experimental programme and the corresponding performance results are furnished in Section 8.5.

8.2 System Environment

8.2.1 Switch Architecture

Each Olivetti Switch [Milway 94] is an ARM-60 RISC[2] processor based machine, running at a clock speed of 24 Mhz. One Mbyte of on-board RAM is available for holding the kernel image, user processes and the other necessary information. Each switch contains four ATM ports, each with 100 Mbits/s full-duplex physical layer taxi [Taxi 92] link protocol running over co-axial cables. Each ATM port uses 32-bit wide FIFOs with a storage capacity of 64 standard 53 octet ATM cells[3].

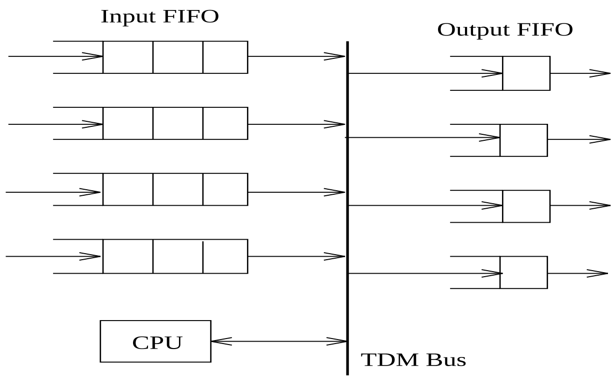


Figure 8.1: Architecture of the Olivetti Switch

Cells can be read from any input FIFO, and written to any number of the output FIFOs using the time division multiplexed bus, shown in Figure 8.1 [Doar 93]. The output FIFOs are much smaller than the input FIFOs and exist in order to cope with momentary delays in cells leaving the switch. A tag FIFO is used to maintain status information (valid, short, HEC error and corrupt data) about each cell as it enters the FIFO and is used as the cell comes to the head of the queue. The action of writing the last word of a cell causes the cell to be read out of the FIFO if no other cells are waiting (i.e. the switch does not operate in cut-through mode). Basic link level flow control mechanisms are supported using the in-band signalling capacity of the transmission interfaces.

[2]Reduced Instruction Set Computer.

[3]Not to be confused with the *transmission cells* of the radio entities.

8.2.2 Operating System

WANDA is a locally developed micro-kernel which supports a *lightweight* platform for running user programs. The kernel and the programs are cross-compiled using the Gnu *C* Compiler (GCC), on HP architecture, with optimisation turned on. Once compiled, the user programs are *spliced* on to the kernel for execution and the *spliced* image is downloaded from the boot-server during the booting operation.

A WANDA process, known as a *domain*, is an address space which is shared by a collection of independently scheduled threads. A single address space (*sas*) version of WANDA was ported by the author onto the Olivetti Switches. Compared to the virtual memory supported version, *sas* WANDA offers improved performance by avoiding the excessive data copying and the context switching overheads. A simple thread scheduling policy is used where a running thread can be preempted by another runnable thread with higher priority (statically assigned at the time of forking). Thread switching is optimised [Sreenan 93] by arranging for a thread which blocks to *spin* within the scheduler based on its wake up condition, thus avoiding a change of context in the case where it is resumed prior to any other activity. The ported version does not support any time-slicing; this ensures that a running thread would never be preempted by another thread of equal priority. The kernel itself is a collection of preemptable threads which includes background threads like device drivers. Inter thread synchronisation is provided by means of traditional *semaphore* operations.

An Inter Process Communication (IPC) mechanism, derived from the Berkeley UNIX socket abstraction, is provided by the WANDA kernel. It currently supports the Multi-Service Network Layer (MSNL) [McAuley 90] protocol, adapted for the ATM networks. Fairisle Data Link (FDL), which is one of the available link layer protocols supported by Multi-Service Data-Link Layer (MSDL), is used for the point-to-point links of the Olivetti Switches.

The kernel supports network and device access using I/O buffers. An I/O buffer is realised as a contiguous region of physical memory, mapped to the address space of its owner *domain*. During an IPC operation, an I/O buffer is de-mapped from the source *domain*'s address space and re-mapped to the address space of the destination *domain*. This strategy enjoys the advantages of reduced data copying by replacing it with the pointer exchange operations. In the ported version of WANDA, the largest I/O buffer is set to be 2048 bytes, of which 128 are reserved for kernel use. The device drivers for the switch ports are implemented as background kernel threads. A driver, if blocked, is woken up by a hardware interrupt (IRQ) generated by an incoming ATM cell. Once woken up, to optimise performance, the interrupt service routine services all the cells available in the Rx-FIFO of the relevant port. This partially mitigates the overhead caused by frequent interrupt services and the subsequent thread switching.

8.2.3 Networking Protocol

Multi-Service Network Architecture[4] (MSNA) is used as the generic ATM protocol for shipping data within the backbone and the emulated radio network. A comprehensive description of it is furnished in [McAuley 90] and [Dixon 91]. For the sake of completeness and the ease of continuity, the relevant features of MSNA are highlighted in this subsection.

One of the prime design philosophies of MSNA is to minimise the layered multiplexing, often found in the conventional communication protocols. The principal benefit of this approach is to be able to identify and schedule the relevant active entities (usually the processes and threads) in the end system with the minimum amount of protocol processing. Another driving force behind such a virtual circuit based network layer design is to dissociate data and signalling, in its fullest extent. There are three protocol layers which contribute to the MSNA internetworking service.

- MSDL and MS-Access (data-link level)
- MSNL (network layer)

MSDL defines *lightweight* virtual circuits, referred to as *associations*. A VCI pair defines an association over a particular data-link. The MS-Access layer is network specific and the currently supported networks are Cambridge Fast Ring [Hopper 88], Cambridge Backbone Ring [Greaves 90], Ethernet and the Fairisle [Leslie 91] Switch Fabric[5]. Multi-Service Network Layer (MSNL) extends associations into an inter networking environment. A set of single-hop associations are concatenated at the network layer to form what is known as a *liaison*[6]. MSNL does not impose any additional protocol overhead to the data streams; the MSNL PDU and SDU are equivalent and defined to be the size of the MSDL SDU, which is 48 octets for the FDL implementation. An MSNL service access point (SAP) comprises:

- A four octet MSNL identifier and
- A four octet MSNL port.

Although there is no theoretical restriction of one-to-one correspondence between an MSNL identifier and a single host, in the current implementation each switch is assigned a globally unique MSNL identifier during the RARP[7] [Comer 91] procedure before booting. An MSNL port, on the other hand, is an MSNL client specific entity which is allocated when the client

[4]MSNA was developed in Cambridge Computer Laboratory and an operational MSNA internet currently exists within Olivetti Research Ltd. and Computer Laboratory.

[5]Note that the Fairisle Data-Link (FDL) is used for the current set of experimentation with Olivetti Switches.

[6]A *liaison* is often referred to as an MSNL connection.

[7]Reverse Address Resolution Protocol.

registers an interest in listening for a connection or when it requests setting up an MSNL connection. Out-of-band *metasignalling* mechanism is used for creation and destruction of the MSNL connections.

Multi-service segmentation[8] and reassembly (MSSAR) is the only higher layer protocol in the version of MSNA, used for the current implementation. MSSAR enables the users to transfer variable size blocks over MSNL connections.

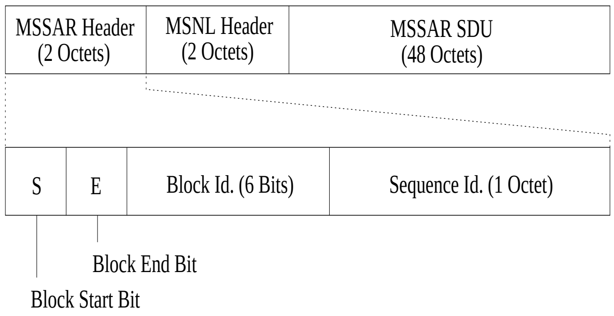


Figure 8.2: Format of an MSSAR PDU for FDL

The start and the end bits, shown in Figure 8.2, indicate the first and the last cells of a block. The block identifier field starts at zero when a virtual circuit is created and it is incremented (modulo 64) by unity after each block transmission. This ensures that at least 63 consecutive blocks have to be dropped before there is the possibility of one block being incorrectly assembled. The sequence identifier starts with the number of cells in the corresponding block and counts down to one in the last cell. This provides the block size information to the receiver end, on reception of the first cell of a block. Note that the present implementation of MSSAR does not support out-of-ordering of the ATM cells; any such event causes a termination of the reassembly of the current block at the receiver end.

8.2.4 Emulation of Radio Links

Because of the unavailability[9] of any working radio link at the time of these experimentations, the described switches were used for emulating both the base and the mobile entities. All the four ports of a switch, emulating a mobile unit, can be configured as radio interfaces. However, one port is usually kept reserved for the wired network for the control, management and the experimental measurement purposes. A base station switch may have a maximum of three ports configured as radio; at least one port is used for connecting it with the wired backbone network. A radio port of a mobile unit is connected to another radio port of a base station by a co-axial cable. Figure 8.3 shows how a mobile unit is con-

[8]Not to be confused with the mobile connection *segments*, introduced in Section 7.4.

[9]Although a radio link was under investigation in the Olivetti Research Ltd., it could not be used because of its experimental nature at the time of the present experimentation work.

nected to multiple base stations for emulating the *transmission cell* overlapping situations, mentioned in Section 6.2. For the mentioned 4x4 switches, a mobile unit can simultaneously see at most four base stations and a BS with three radio ports can see at most three mobile units at a time.

When two radio ports are physically connected by a co-axial cable, both the end entities (a BS and an MU) jointly maintain a state machine representing the changing behaviour of the quality of the corresponding radio link. Dynamics of the link quality state, including disconnection, is controlled by a pre-loaded model [Porter 94], obtained from user mobility experiments carried out in an in-building environment. Large scale relocation is simulated by manual connection and disconnection of the co-axial cables from the radio ports. More about the radio link state model is discussed in Section 8.4.5. A *tunnelling* [Roe 93] approach is used for emulating the radio *link layer* over a co-axial cable, connecting two radio ports. Whenever two radio ports are physically connected together, an MSNL (using FDL) connection is created and subsequently maintained over the relevant ports; the connection is referred to as a *Tunnelled Radio Connection* (TRC). For each pair of radio ports only one TRC exists at one time. This network layer connection is used for exchanging the radio link layer PDUs between the two end entities. From this point onwards, this underlying emulation model will be assumed without any explicit reference to it.

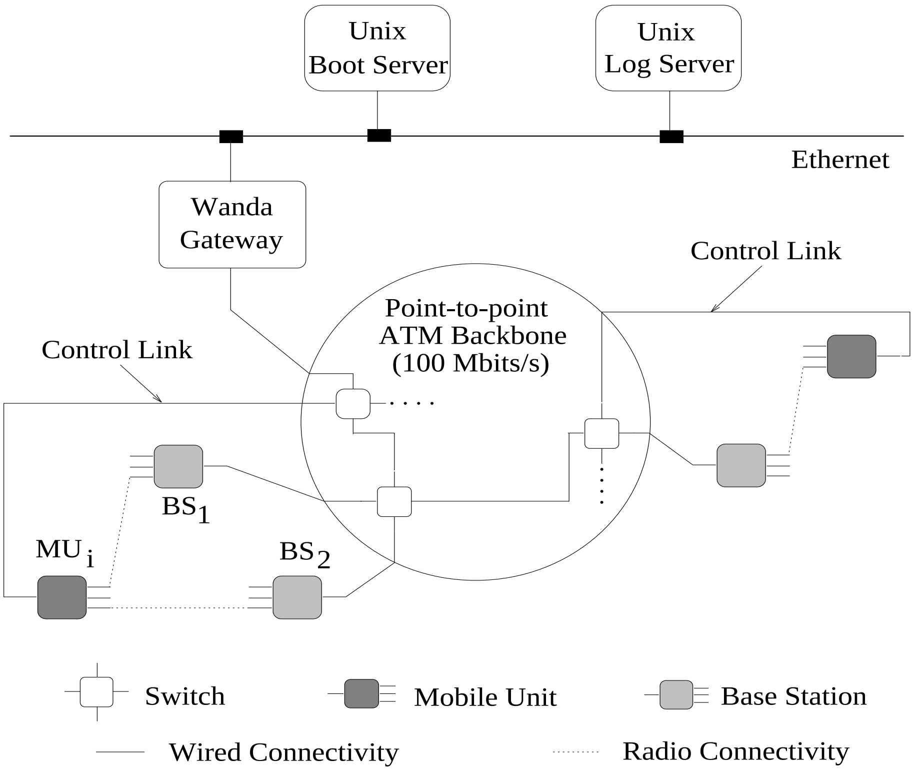


Figure 8.3: Configuration of the prototype LAN

8.3 LAN Configuration

The prototype LAN configuration is shown in Figure 8.3. A private network, comprising of base stations, mobile units and a point-to-point ATM backbone, is built, using the Olivetti Switches, described in Section 8.2.1. The addressing and the clustering schemes, mentioned in Section 6.2.2, are used for partitioning the backbone network. A Motorola 68030 based WANDA machine is used as the MSNL gateway [Dixon 91] between the prototype LAN and the Ethernet. To permit easy interconnection of ATM networks, the ATM protocol is also carried across the Ethernet, so that an Ethernet frame is filled with ATM cells in a format which enables rapid cell forwarding at Ethernet-ATM gateways.

All the switches, including the emulated base and mobile units, boot from a *boot-server* running on a UNIX workstation. Depending on the type of a *boot-client* (BS, MU or ordinary switch), the *boot-server* provides an appropriate boot-image, containing the relevant user processes *spliced* onto a WANDA kernel. It is to be noted that one port of each mobile unit switch is connected to the backbone network by a *Control Link*. These links are used for booting and MU statistics collection by the *log-server*, running on another UNIX machine. Since the switches do not have any local file system, all the required statistics from a switch are sent to the *log-server*, through an MSNL connection, setup at boot time.

8.4 Implementation Details

8.4.1 CLS and Mobile Representatives

A single cluster version of the proposed location service (presented in Chapter 6) is coded for the current prototype implementation. Each CLS is realised as a collection of threads, with one of them responsible for creating location update connections with the local base stations. Soon after its creation, the thread starts listening on a *well-known* MSNL port for incoming connection requests from the local base stations; upon reception of such a request, a location update connection is established with the requesting BS. This happens only if no such connection already existed with the same base station. Another dedicated thread, within a CLS, is used for sending *keep alive* messages over all the existing location update connections. Any end entity of such a connection may time-out if the other end stops sending *keep alive* messages through the connection. A message passing scheme is used for creation and maintenance of the location server; the details of the messages are furnished in Appendix C.

In the current implementation, the CLS address is fixed within a particular cluster; this exempts a location client from consulting any directory service in order to locate the local location server node. The location update connections between the CLS and the local base stations are also used for the representative management operations. The proposed representative architecture and its message based management strategies (presented in Chapter 6) remain unaltered during this implementation.

8.4.2 Mobile Communication Interface

In order to support the mobile applications, a *mobile communication interface* is developed in the user space. A structural overview of the system software components and their functional placements are shown in Figure 8.4. An extra layer of user space code, comprising of the library routines shown in Table 8.1, is added for handling the mobility related activities, including handover.

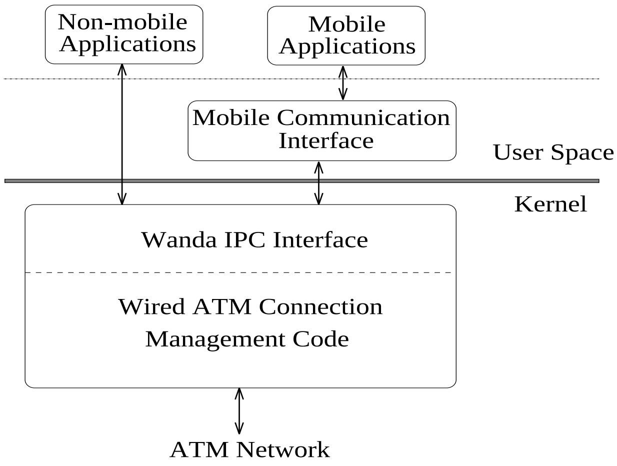


Figure 8.4: Overview of the system software components

Note that the current implementation of the mobile communication paradigm supports only the MSNL protocol family. The segment[10] management and the handover related operations are kept hidden from the end mobile applications, by using the functions listed in Table 8.1. An incoming mobile connection can be established in two different ways. In the first approach, an application (essentially a WANDA thread) registers its interest in receiving an incoming connection by invoking the *MobileOffer* routine. When a matching call request arrives, the application thread can check the details of the remote end point, including its peer address. Based on this information, the thread may or may not *Accept* the connection. *MobileListen*, on the other hand, forces an invoking thread to *Accept* a matching incoming connection, without any checking provisions. Other routines perform the operations, similar to their corresponding non-mobile counterparts, initially developed for WANDA IPC interface. Although the current set of interfaces are sufficient to offer the mobile communication infrastructure, it is envisaged that another layer of library routines would be useful for hiding *mobility* completely from the user applications.

[10]The terminologies, introduced in Chapters 6 and 7, are used throughout this chapter.

buffer: WandaIOBuffer mobile_socket: MobileCommunicationDescriptor address: WandaIPCAddress mobile_socket = GetMobileSocket(PROTOCOL_TYPE) MobileBind(mobile_socket, address) MobileConnect(mobile_socket, address, timeout) MobileOffer(mobile_socket, timeout) MobileAccept(mobile_socket) MobileListen(mobile_socket, timeout) MobilePeerName(mobile_socket, &address) MobileClose(mobile_socket) MobileSend(mobile_socket, buffer) buffer = MobileRecv(mobile_socket, timeout)

Table 8.1: Library routines for handling mobility

8.4.3 Connection Modules and Datapath

Mobile Module

An invocation of the *GetMobileSocket* routine creates a collection of resources, referred to as a *mobile module*. A pointer to the module, known as a *mobile_socket*, is returned to the invoking thread and the thread uses this *mobile_socket* as a descriptor of the created mobile connection.

It is allowed for a mobile module to maintain multiple sockets in a socket list, as pictured in Figure 8.5. Each such socket, with the information (e.g. identifier, status etc) associated with it, corresponds to a segment of the mobile connection. For each socket, a pair of data *pumping threads* are used for sending and receiving I/O buffers through the relevant connection segment. It is important to note that at any one time only one segment can have access to the queues and to ensure that, each socket needs to maintain a set of status and other thread control information associated with it. Two intermediate queues provide the necessary buffering of the user data between the application and the network sides. Required thread synchronisation is achieved by using two semaphores, as shown in Figure 8.5. Synchronisation is primarily required between the application side threads (invoking *MobileSend* and *MobileRecv* routines) and the buffer pumping threads.

For each mobile module, a dedicated manager thread performs the segment and the handover management operations associated with the relevant end of the corresponding mobile connection. Any significant change of the radio link status causes the manager thread to be activated; if the link status changes too frequently, the generated events are buffered

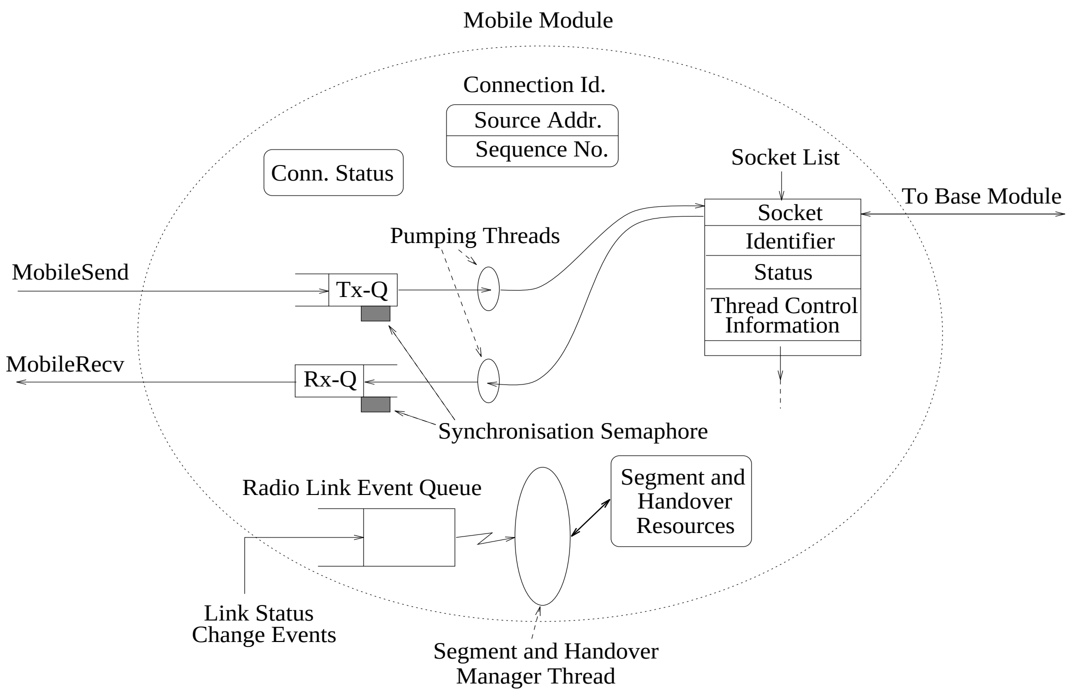


Figure 8.5: End *Mobile* module of a connection

within the *radio link event queue*. More about the manager thread operations are discussed in Section 8.4.5.

Since no handover and segment supports are needed for the stationary end entity of a mobile connection, the corresponding mobile module does not contain the management thread and its related resources. Only one socket is needed and the application is allowed to send and receive I/O buffers directly through the socket, without going through the buffer queues.

Gateway and Base Modules

An intermediate stationary node, which is not a base or representative entity, creates a user space *gateway module* to pass through a mobile connection. Such a module is shown in Figure 8.6. Once the necessary MSNL connections are established (identified by the two sockets in the figure), the data received from the left sided socket is forwarded to the right sided one and vice-versa. Two threads are employed for performing these data forwarding operations. Since the used version of the WANDA scheduler does not support *time slicing* and the threads block during a data reception operation (on *WandaIPCRecv* system call), no user space synchronisation is needed for those two forwarding threads.

When a base station receives a mobile connection or a segment setup request (see Section 7.5.1), it creates a *base module* as shown in Figure 8.6. The structure of a base module is similar to that of a gateway module, with some additional buffers needed for handover op-

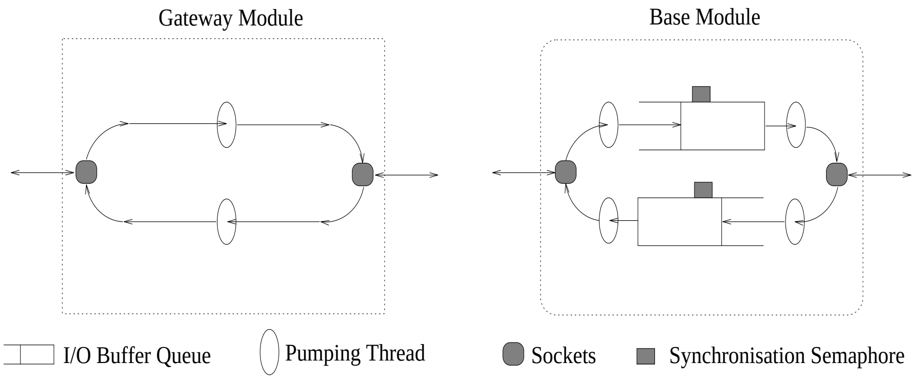


Figure 8.6: *Gateway* and *Base* modules

erations. The threads which put I/O buffers in and out of a queue, need to be synchronised and a pair of semaphores are provided for that purpose.

Representative Module

The components of a *representative module* are depicted in Figure 8.7. Similar to the mobile modules, multiple sockets can exist for handling segments at the mobile side of a representative module. A segment switching involves activation and de-activation of the relevant pumping threads; the necessary handover buffering is done in the I/O buffer queues.

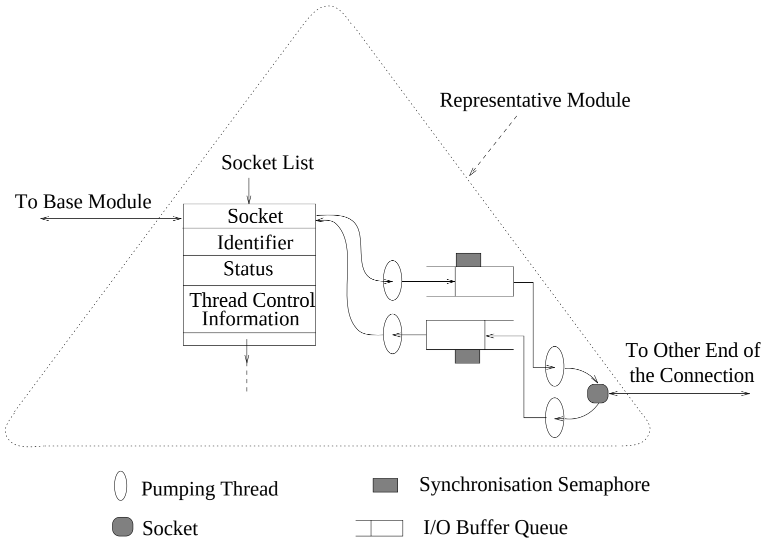


Figure 8.7: Structure of a *Representative* module

The gateway, base and the representative modules spawn from a common *seed thread*, known as a *WildListener*, which is created soon after booting of a stationary unit. The WildListener thread listens for the incoming connections on a *wildcard* MSNL port. Any incoming connection, destined to some other network entity, is accepted by this thread and depending on the connection type (mobile or non-mobile) and stage (if it is a mobile connection) it creates one of the gateway, base or the representative modules. Any further activities regarding that connection are handled by the appropriate newly spawned module. The original WildListener thread continues listening for the future incoming connections.

Datapath

One end of a mobile connection *datapath*, formed by the mobile, base and the representative modules, is shown in Figure 8.8. Note that, at any one time only one base module is active in transferring data from the mobile module to the representative module and vice-versa.

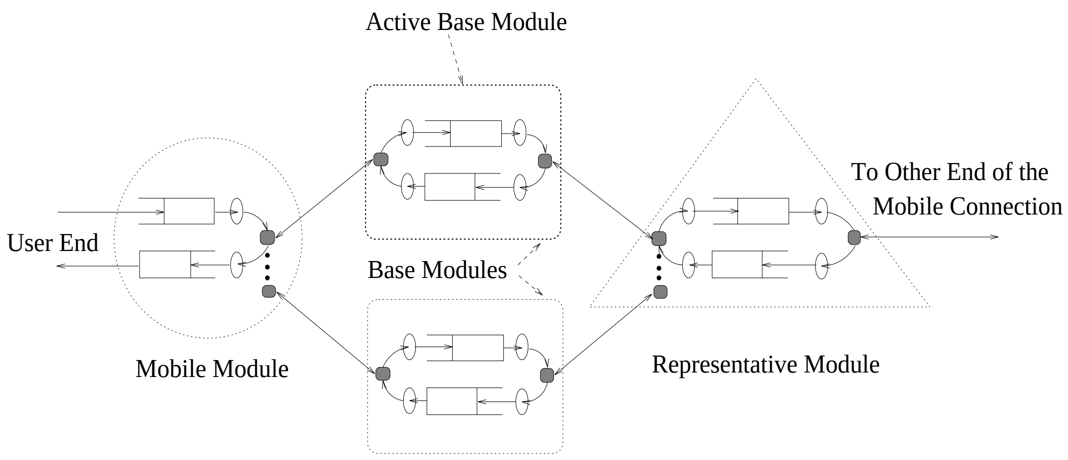


Figure 8.8: End-to-end datapath of a mobile connection

Implementing all the components of the datapath in user space is quite expensive both in terms of system resource and delay/jitter performance. In order to maintain an accepted performance level, in the current implementation a WildListener thread is allowed to create a maximum number of modules. The closure of each of the modules needs careful consideration regarding the release of the occupied resources. Since a module operation involves multiple threads of execution, the threads have to agree about whose responsibility it is to free the resources associated with the relevant user level module. A general approach is taken where the threads in a module exit sequentially after updating the corresponding status information and the last survivor thread releases the module occupied resources before exiting.

8.4.4 Metasignalling and Routing

A *metasignalling* approach is used for setting up and closing down the mobile connections. It was explained in Section 7.5.1 that a combination of *source* and *hop-by-hop* policies is used for routing a mobile connection. This gives rise to the necessity of adding an additional 4 octet field in the FDL setup metasignalling packet, in order to specify the address of the end-entity of a segment; this field is referred to as *source routing destination*. The WANDA kernel routing operation is modified accordingly so that a router always chooses the appropriate outgoing network interface based on the *source routing destination* field, instead of the ultimate destination of a connection. For a non-mobile connection however the setup *metasignalling* message is always routed towards the ultimate destination of the connection. The DMTM topology reconfiguration and its associated hop-by-hop routing protocols, presented in Chapter 3, are used within the wired backbone network.

In order to achieve consistent routing and maintenance of the mobile connections, the following fields are added in the standard metasignalling packet format [Black 93].

- Mobile Connection Id.: This 6 octet (4 octet source address + 2 octet sequence number) identifier is used extensively during segment and handover management.
- Segment Id.: Currently, this is a single octet field and is used for identifying a segment within a mobile connection; again used for segment and handover management. While setting up a new connection this is allocated either by an end MU or by a mobile representative. Since a new segment, during a segment management, is always created by an end MU, the latter assigns the Segment Id. for all the newly created segments.
- Connection Stage Id.: This 4 bit information is used for representing the stage of the current segment within the mobile connection. This information is used by a WildListener thread to decide the type of *module* to be spawned (see Section 8.4.3).

8.4.5 Connection Maintenance

Signalling

Unlike the connection/segment setup and tear-down procedures, the handover operations rely on pre-established *signalling* connections. One such connection exists between a base station and a representative host entity, if they have at least one mobile connection segment passing through them. Soon after the first segment is established, the base station connects to an MSNL signalling port of the representative host entity. When all the segments are deleted, it is again the base station which closes down the signalling connection. A user level thread is employed in each base station and representative host unit, for handling the handover signalling messages. The periodic *keep alive* messages are used by these threads for checking the end consistency of a signalling connection.

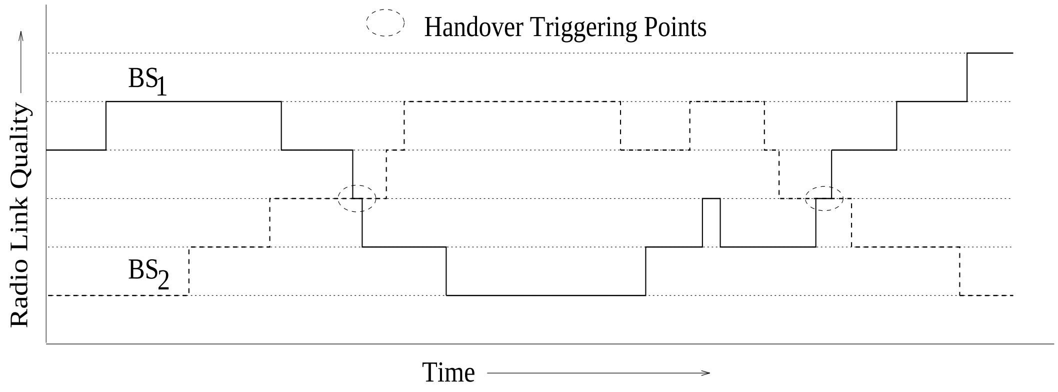


Figure 8.9: Radio link quality change and handover triggering

Segment and Handover Management

As mentioned in Section 8.2.4, the quality of an emulated radio link is varied according to a pre-loaded mobility model. Such a model is pictured in Figure 8.9, where the solid and the dashed lines represent the time varying quality of the radio links between the mobile unit MU_j and the base stations BS_1 and BS_2, of Figure 8.3. The monotonic increment of a link quality represents the movement of the mobile unit towards the corresponding base station. Whenever the link-quality lines cross each other, there is a possibility of a handover triggering by the mobile unit.

Any change of link status produces an event, which is sent to the Segment and Handover Manager (SHM) thread of the corresponding mobile module. Depending on the event type (link *addition*, *deletion* or *quality change*), the SHM thread performs the segment creation/deletion and handovers, if required. The actions taken by an SHM thread, on reception of a link status change event, are summarised in the flowcharts in Figure 8.10. In these flowcharts, the existence of a radio link is assumed only when the link quality is good enough for creation of a mobile connection segment. A handover decision is always taken on the basis of the comparative radio link qualities associated with the current active segment and the other available segments. More about the segment management and handover operations are already discussed in Chapter 7.

8.5 Experimental Programme

The following experiments were carried out for evaluating the performance of the implemented prototype architecture. A *soft* approach was taken where most of the code was written within the user space, primarily to maintain simplicity and portability but of course, at an expense of compromised performance. The experiments are divided into following three groups:

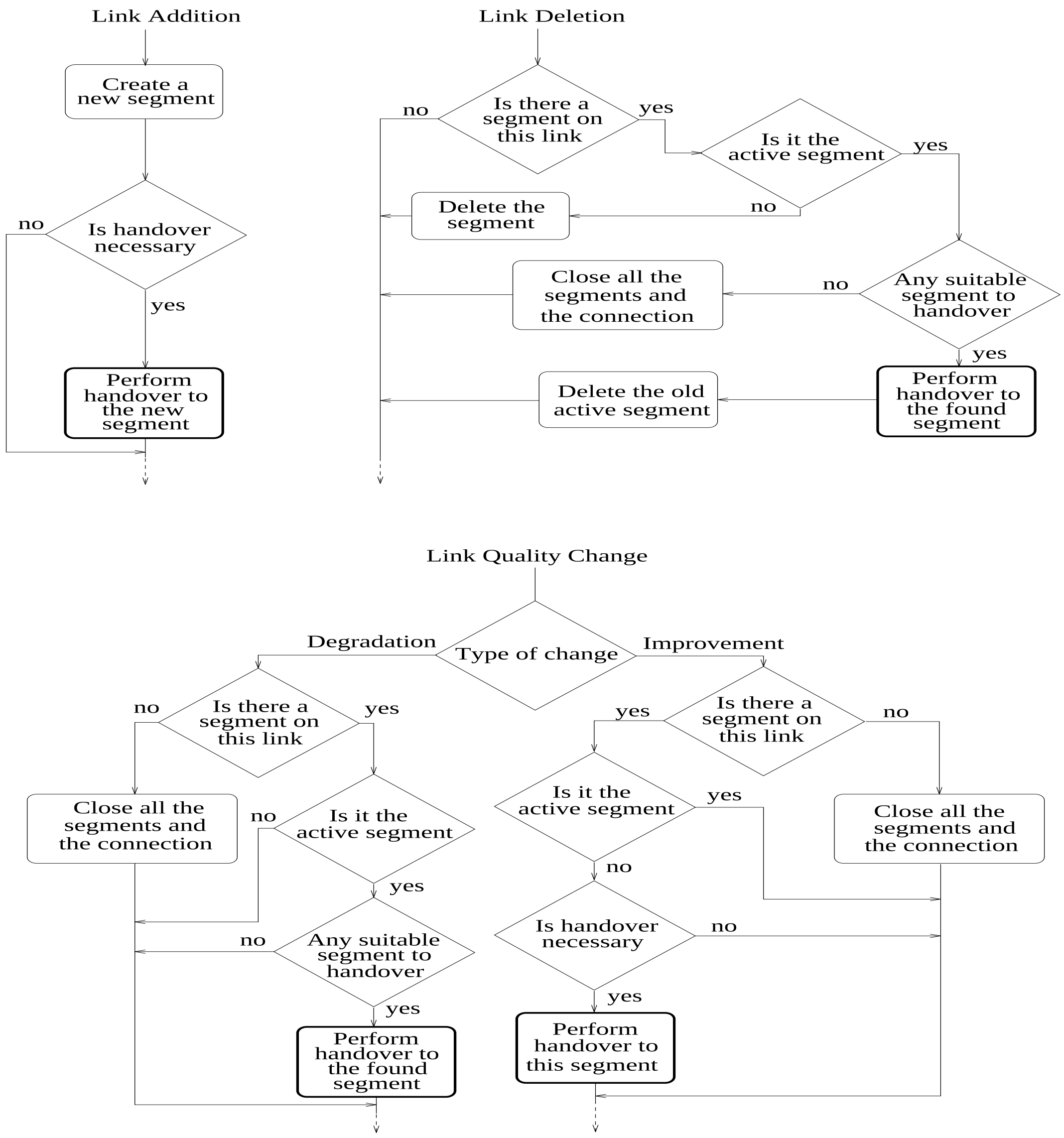


Figure 8.10: Segment and handover manager thread activities

1. Measurement of signalling and datapath performance within the backbone network. The results from these experiments are used for characterisation of the WANDA and MSNA, running on the switches; these also provide a reference level to compare the performance of the mobile connections.

2. Examination of the Mobile Communication Interface performance, under varying loading and system conditions.

3. Feasibility study of the segment management and the handover strategies.

Table 8.2 summarises the specifications of two constant bit rate (CBR) audio and video streams, used for the following experiments. Each audio block represents 4 ms of data sampled at 8 KHz and 8 bits per sample [Sreenan 93]; with additional protocol overhead the block size is rounded to the nearest multiple of FDL cell size, i.e. 48 bytes. The video entry corresponds to a full rate (25 frames per second) stream with block size similarly rounded to the nearest multiple of 48 bytes. In the present experiments, dedicated WANDA threads are used for generating streams which correspond to the described audio and video specifications.

Stream Type	Block Size (Octets)	Period (ms)	Datarate (Kb/s)
Audio	96	4	192
Video	1920	40	384

Table 8.2: Details of the streams used for experimentation

Unless otherwise stated, all the presented results correspond to MSSAR level performance, as seen by the end user applications. A default *best-effort* QoS approach was taken for the experiments, described throughout this chapter. A number of *background* threads were present during most of the experiments. These threads include a kernel one, checking sanity of all the four ports of the host switch once in every 4 seconds; a group of other background threads were actively involved in backbone network reconfiguration (see Section 3.7) and location management. Connection performance degradation, especially in terms of *jitter*, caused by these background threads, is investigated in Section 8.5.1. A remedy, based on asymmetric CPU scheduling priorities, is proposed in Section 8.5.3.

8.5.1 Backbone Network

This section details the performance of MSNA running on the version of WANDA, ported onto the Olivetti switches. The objective of the experiments is to tune the system parameters of the target environment, before the implemented mobile communication interface is investigated. To maintain a similar environment, mobile units were connected to the network during these experiments so that the location and the representative management threads were always present, unless otherwise stated explicitly.

Signalling

Table 8.3 summarises the connection setup latency under different kernel clock interrupt frequency. In each case, 1000 connections were setup with an interval of 1 second and the corresponding latencies were recorded. Note that the large difference between the best and the worst setup performance for each kernel clocking scenario, is primarily caused by the background threads, mentioned earlier. With frequent clock interrupts, the CPU spends

Kernel Clock Interrupt Interval (ms)	One Hop Setup Latency (ms)			Two Hop Setup Latency (ms)			Three Hop Setup Latency (ms)		
	Min	Max	Mean	Min	Max	Mean	Min	Max	Mean
0.5	3.72	7.48	5.28	10.20	15.27	12.88	15.85	20.10	17.96
1.0	3.22	6.17	4.57	9.02	16.35	11.08	14.31	19.71	17.19
2.0	3.15	6.17	4.16	8.37	13.62	9.43	12.38	17.11	14.41
4.0	2.40	2.88	3.37	5.89	11.25	7.60	9.85	13.92	11.98
8.0	2.30	6.21	2.80	5.50	9.53	7.25	9.02	13.70	9.99
16.0	1.68	7.03	2.14	5.26	10.68	5.99	8.06	12.16	9.21

Table 8.3: Connection establishment latency in backbone network

more time within the Interrupt Service Routine (ISR), causing increased setup latency. It is worth noting that for two hop connection setup, the latency is more than double of its one hop counterpart. The additional time (e.g. 1.11 ms for the 2 ms case) is spent within the router for constructing the *gateway module*, described in Section 8.4.3. The average connection closing latency was found to be 405.3 μsec.

Initially, it was not very clear that how frequently the kernel should be clock interrupted while implementing the Mobile Communication Interface; detailed study in Section 8.5.3 suggested that a 2 ms interrupt interval is required to attain an optimum performance for this particular implementation. More about the clocking frequency selection is discussed in the later sections.

Delay

The delay, experienced by the user level clients, are presented in Table 8.4. Because of the unavailability of any hardware or communication protocol for global time synchronisation, round trip latency is chosen as an index of the network delay. A dedicated *source* thread sends a block once in every 40 ms and it is collected by a *sink* thread, running in the same machine. A *bouncer* thread is arranged at the remote-end machine of the connection which receives and immediately sends the block back through the same connection. The round trip delay is calculated from the time-stamps, put by the source and the receiver threads, on the block header. This procedure was repeated for 50000 blocks in order to establish each set of the presented results.

The round trip delays, with 2 ms kernel clock interval, are tabulated for both single and multihop scenarios. The influence of varying thread scheduling priorities are also investigated; the *high priority* column represents the case when the source, sink, bouncer and all the data *pumping threads* within a gateway module run with a higher priority level than the default user thread priority. All the other background threads, comprising the location and

Block Size (Octets)	One Hop (ms) Data Priority		Two Hop (ms) Data Priority		Three Hop (ms) Data Priority	
	High	Default	High	Default	High	Default
48	0.613	0.722	1.187	1.501	1.907	3.249
240	0.802	1.107	1.526	2.236	2.467	3.891
480	1.005	1.486	2.014	2.982	3.254	4.792
720	1.204	1.911	2.459	3.751	3.982	5.542
960	1.386	2.302	2.922	4.512	4.658	6.387
1200	1.551	2.521	3.300	4.867	5.322	7.214
1440	1.723	2.751	3.523	5.257	6.038	8.102
1680	1.901	2.948	4.041	5.401	6.875	9.019
1920	1.987	3.046	4.312	5.424	7.551	9.687

Table 8.4: Round trip delay with different data thread priorities and hop-counts

the reconfiguration manager, execute with the default user priority. A *default priority* entry represents a situation where no preference is given to the data threads and all the threads are running with default user space priority. The benefits of assigning higher priority to the data handling threads are quite apparent from the numbers in Table 8.4. Inspired by the current results, a similar strategy was adopted while coding the mobile communication interface.

Calculation shows that for the minimum block size (an FDL cell) and with higher data priority, an insertion of a user space router incurs a routing delay of 287 μs. This delay includes the copying time of the buffer from the input network interface FIFO to the user space and again to the FIFO of the outgoing network interface. The setting up time of the intermediate buffers also accounts for a part of this routing delay. With the current approach of routing, the delay and jitters are relatively higher than the *cell forwarding* strategy [Dixon 91], where a quicker datapath is set within the kernel while setting up the connection through the router machine. This reduces the overhead of copying data to and from the user space for each block routing. Frequent context switching is also expected for the user space implementation where both the pumping threads are to be scheduled alternately, depending on the traffic flow in each direction. Even in the best possible situation where the pumping threads of a gateway module (see Section 8.4.3) are the only two user space threads, frequent context switching is necessary to route blocks in both the directions. Considering the implementation simplicity and acceptable performance level, a software routing approach is used for the rest of the experiments described in this chapter. Different datapath modules (mobile, base and representative) of a mobile connection also adopt a similar mechanism of using user space threads for data forwarding. Mapping all these functions into a *cell forwarding* implementation is one of the candidates for future extension.

As expected, the round trip delay increases almost linearly with the size of the transmitted blocks. Bigger blocks incur more delay in terms of transmission, reception and copying the data around, within the router. Additionally, increased segmentation and reassembly latencies, both within the end entities and the router, contribute to this increased latency. The ISR overhead, on the other hand, is relatively lower for the bigger blocks and this gives rise to higher sustainable throughput.

Jitter

The following set of experiments was carried out to measure the jitter performance of video and audio streams under varying system conditions. For each experiment, 50000 blocks were sampled at the receiver side and the corresponding video inter arrival time distributions are plotted with a group-interval of 40 μs; audio distribution needed a lower group-interval of 15 μs so that the relatively small variations in inter arrival times can be highlighted. In addition to the standard deviation, jitter is also expressed in terms of *centile* values where the nth centile is defined as the minimum time interval in which n% of the sample distribution occurs [Sreenan 93].

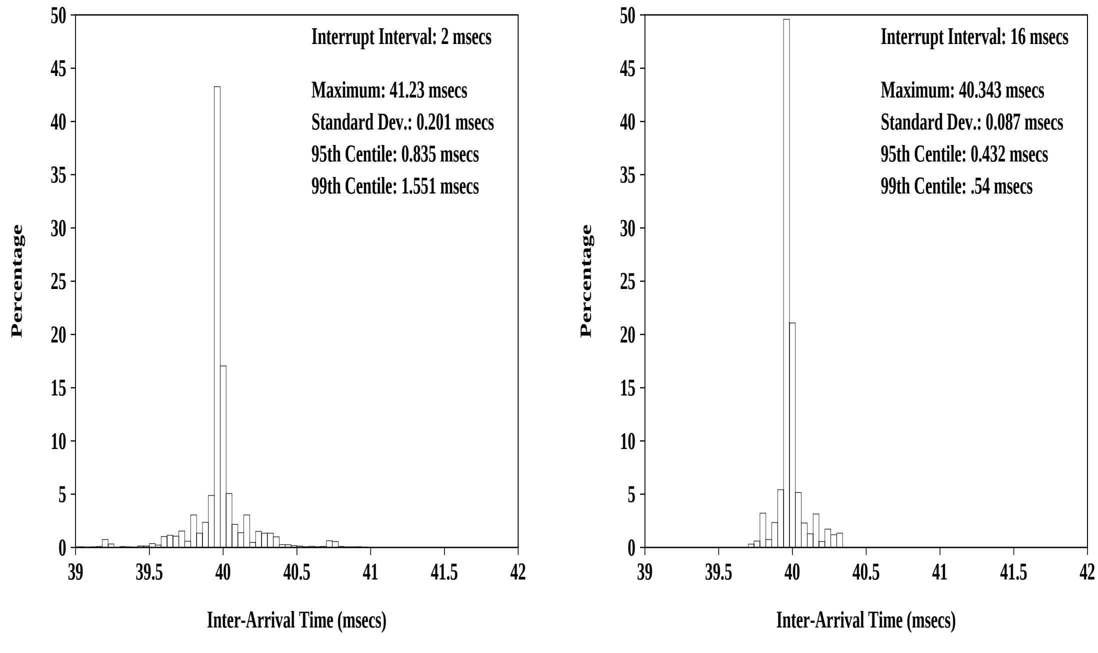


Figure 8.11: Distribution of video inter arrival times with varying kernel clock interrupt frequency

The first histogram in Figure 8.11 depicts the jitter suffered by a video stream passing through a single router. Within the router and the end entities, all the data handling threads were running with default user space priority. It is argued that the contention from the active background threads, performing location and topology reconfiguration management, contributes to this high jitter value.

Another expected reason for high jitter was the ISR contention due to the kernel clock interrupts. To verify this, the timer interrupt frequency was brought down to once in 16 ms from once in 2 ms. Improvement of jitter performance is apparent from the difference in 99th centile values of the histograms in Figure 8.11. In general, the timer interrupt frequency should be in the order of the period of the highest frequency thread to ensure accurate scheduling of these threads. This restriction, along with the required lower bound of the time related system call (e.g. $WandaTimeP$) resolution, prevents the clock interrupt interval from being too high in practice. Hardware timing mechanism may be used as an alternative to periodic timer interrupt to the kernel; experiments carried out in [Dixon 91] show a drastic reduction of jitter by complete removal of the periodic clock interrupts.

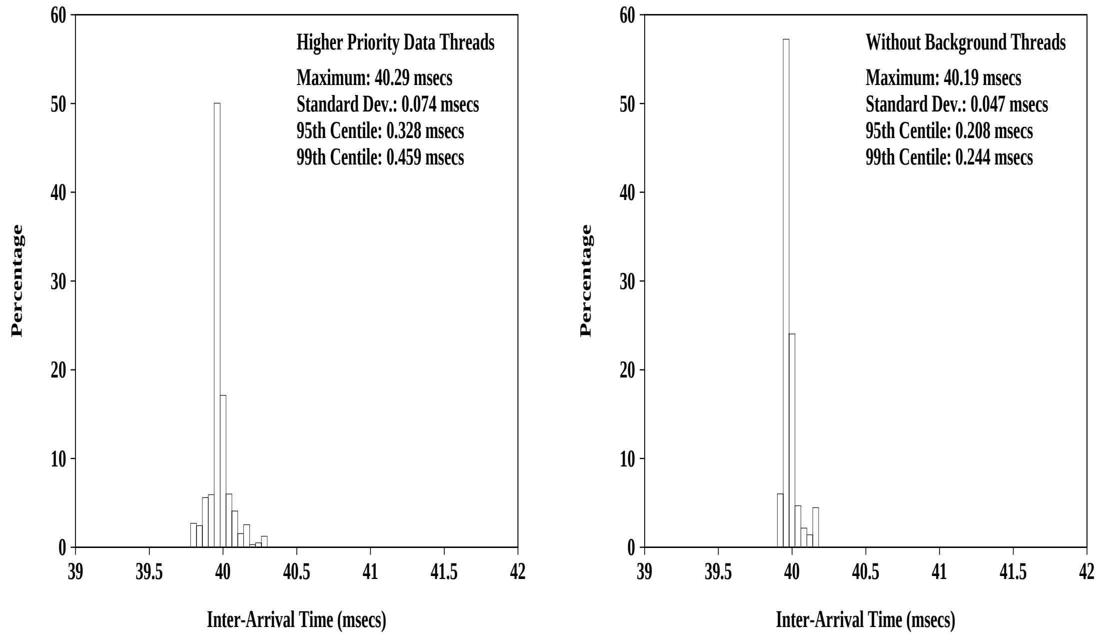


Figure 8.12: Reduction of jitter with improved data priority and reduced background activities

The effects of background threads on jitter were examined in two steps. Initially, the data handling threads (see the last section) were assigned higher scheduling priorities than all the other existing threads, including the background ones. Since the current WANDA scheduling policy allows preemption, high priority data threads experience less contention for the CPU and as a result the jitter is reduced. Figure 8.12 indicates that the user level jitter is brought down to as low as 459 μs of 99th centile value.

As the second measure, all the background threads were disabled[11] and the data handling threads (the source, sink and the data pumping threads within the router) were the only

[11] Note that although the location manager is not necessary at this point of experimentation, disabling the reconfiguration daemon required manual setting of the routing table entries, in order to keep the MSNL connection management going. Considering their low invoking frequency, the kernel background threads for interface sanity checking were not disabled.

user space entities contending for CPU time. The drastic reduction of jitter, shown in the second histogram of Figure 8.12, further verifies the performance degradation caused by the background threads. This also suggests the necessity of transferring few of the management functionalities (e.g. representative handling) to the relatively more dedicated machines.

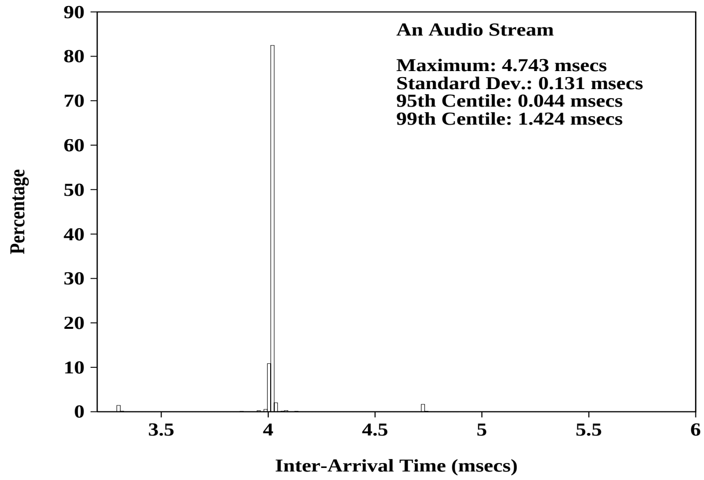


Figure 8.13: Distribution of audio inter arrival time

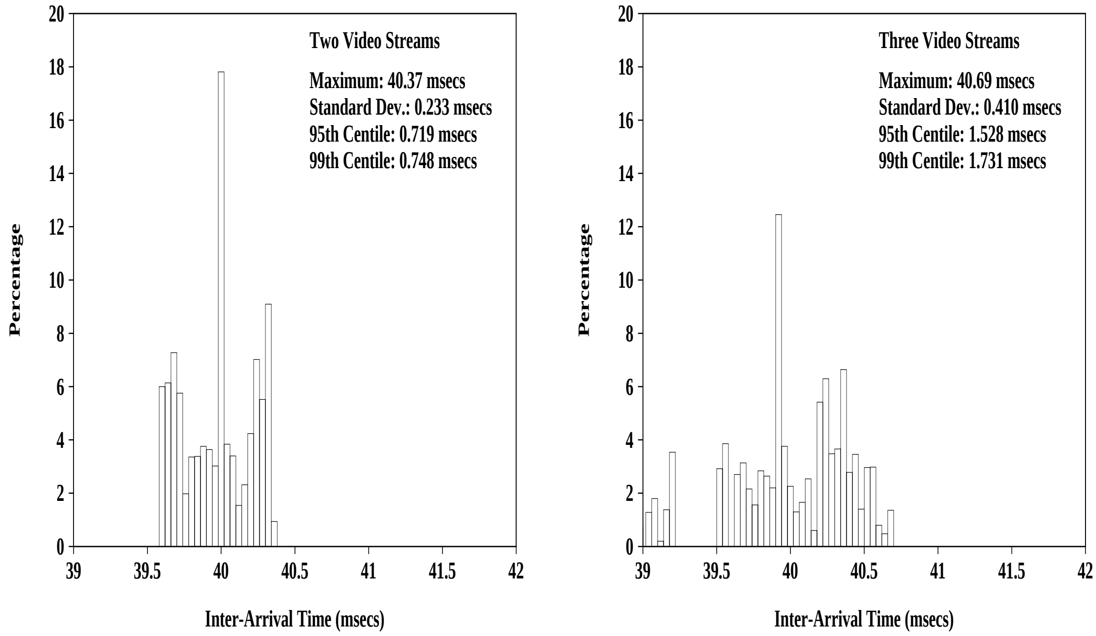


Figure 8.14: Inter arrival time distribution for multiple video streams

The same experiment was repeated with an audio stream and the corresponding inter arrival distribution is presented in Figure 8.13. Under the similar system conditions (infrequent clock interrupt and no background threads), audio showed a much worse jitter performance when compared to video. Especially, the maximum inter arrival time is quite large in comparison to the overall distribution. In the presence of both audio and video streams, large audio jitter is usually reduced by assigning higher scheduling priority to the audio threads, in comparison to video [Jardetzky 92].

Until now, in each of the described experiments, jitter was studied for a single stream. Influence of stream contention is depicted in Figure 8.14, where multiple video streams were generated from a single source machine and passed through the same router upto a common sink entity. As expected, high CPU contention is evident from the multiple stream inter arrival time distributions. Especially within a router, the context switching frequency for the data pumping threads increases almost linearly with the arrival rate of the video cell bursts. This accounts for a substantial part of the large jitter for multiple streams.

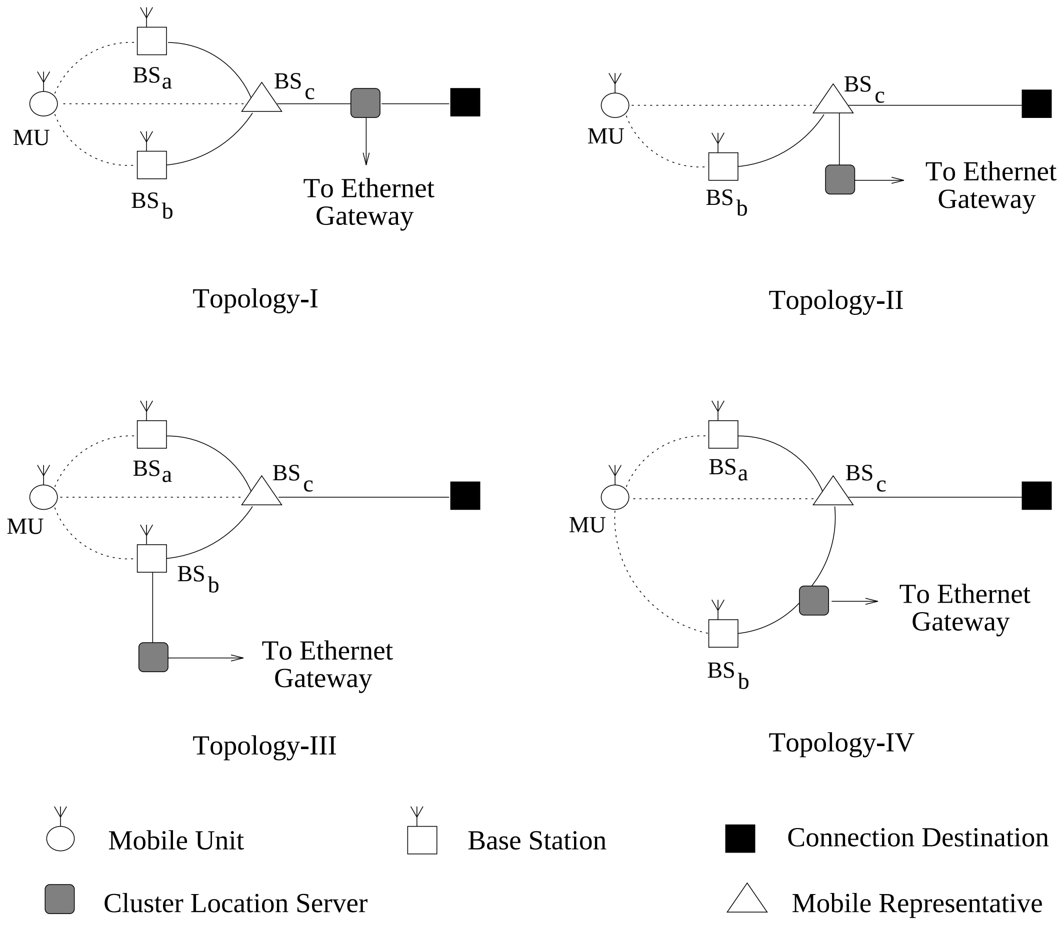


Figure 8.15: Mobile connections with various experimental topologies

8.5.2 Mobile Connections

Performance of the mobile connections, supported by the current prototype system, are presented in this section. Figure 8.15 outlines various topologies of the experimental setup. All the described experiments were carried out in a single cluster environment and a dedicated switch was used for hosting the corresponding Cluster Location Server (see Section 6.6). The CLS host-entity[12] is also used for connecting these experimental switches to the gateway (see Figure 8.3) between the WANDA and the UNIX world. In all the four cases, the mobile representative was arranged to be lodged within a base station; this arrangement allowed performance investigation of the co-located base and representative modules. During all these experiments, the background location and topology reconfiguration threads were active with the default user space priority and the kernel timer interrupt was set as once in every 2 ms.

Signalling

Mobile connection setup and the associated delays are presented in Table 8.5. These figures represent the time required for setting up the *initial path* (see Section 7.5) of a mobile connection. Segment setup and handovers are dealt with in the following sections. A connection, originated from the mobile unit, is routed either through the base station BS_b or through the co-located base and representative entity BS_c. For each table entry, 1000 such connections were setup with an interval of 1 second and the corresponding mean values were computed.

Topology		End-to-end Setup (ms)	BS to CLS Setup (ms)	Location Access (ms)	Signalling Conn. Setup (ms)
I	Non Co-located	22.53	11.12	1.97	8.10
	Co-located	24.14	5.23	1.41	11.73
II	Non Co-located	18.02	12.01	2.12	8.24
	Co-located	19.06	5.19	1.44	12.21
III	Non Co-located	18.16	5.37	1.40	9.29
	Co-located	19.47	10.90	1.95	11.42
IV	Non Co-located	23.63	5.29	1.49	14.02
	Co-located	18.81	5.21	1.44	11.48

Table 8.5: Components of mobile connection setup latency

While creating the first connection, an involved base station needs to setup a location query connection with the CLS to download the location information about the mobile representative. The corresponding setup and location information access delays are also included in Table 8.5. The last column presents the time required by the base station to setup a handover signalling connection with the representative host entity. It is to be noted that the presented figures for end-to-end mobile connection setup delay (the first column in Table 8.5) do not include any of the BS-CLS location connection setup time, location access

[12]Note that, each of the entities, shown in Figure 8.15, represent a suitably configured switch.

delay and the BS-MR signalling connection setup latency. While the first connection setup through BS_b (non co-located) in topology-I takes 43.72 ms[13], the successive connections in steady state need only 22.53 ms to setup, when the location information caching scheme is enabled (see Section 7.5.1) within the base station. Disabling this caching requires BS_b to access the location information of the representative during each connection setup and that, in turn, increases the end-to-end setup delay by 1.97 ms (location access time for BS_b). The BS-MR signalling connection setup latency contributes to the end-to-end setup delay only for the first time and so does not have much relevance in terms of the currently investigated signalling performance. It is worth mentioning that the reverse (i.e. when the MU is the connection destination) setup latency was found to be different from its forward counterpart. Different BS-CLS, BS-MR connection setup time and the location access latencies, depending on the relative placement of the BS, representative and the CLS, account for this discrepancy.

To gauge the connection setup overhead of the mobile communication interface code, it is necessary to compare the present figures with the backbone setup delays, presented in Table 8.3. While the standard IPC code takes 14.41 ms (entry for the 2 ms kernel clock interrupt interval is considered) to setup a three hop connection, an equivalent mobile connection in topology-III requires 18.16 ms to be setup. This additional 3.75 ms of signalling delay is inserted primarily by the mobile, base and the representative module management code within the mobile communication interface layer, illustrated in Figure 8.4. For each topology, performance is shown both for the co-located and the non co-located base and representative modules. Except for topology-IV, connection setup in a non co-located scenario always outperform its co-located counterpart. These effects can be explained by the lack of concurrency between the two WildListener threads (see Section 8.4.3), responsible for the base and the representative module creation. When the base and the representative are located separately, once the representative replies back to the base station, both the WildListener threads can execute parallelly in two different switches. Since in the co-located case both of them are to be scheduled on the same CPU, serialisation of their execution causes this additional delay for connection setup. The high BS-MR signalling connection setup latency, indicated for the co-located entries, can also be explained with similar arguments. In topology-IV however the connection through the base station BS_c is one hop-count shorter than the one through BS_b. The reduction of delay due to this reduced hop-count dominates the performance degradation caused by the co-located base and representative modules; this explains the smaller setup delay for the co-located entry.

A closer look at the numbers in Table 8.5 reveals that the mobile connections, in different topologies but with equal number of hops, do not necessarily have the same setup latency. Investigation suggests that these differences were caused due to the asymmetric loading by the location and the topology reconfiguration threads. While a connection through BS_b in topology-II takes 18.02 ms to setup, an equivalent three hop connection setup through BS_b in topology-III requires a delay of 18.16 ms. Additional reconfiguration activities within

[13] This includes the BS-CLS location connection setup time, location access latency and the BS-MR signalling connection setup delay.

BS_b in topology-III, caused by the physical neighbourhood of CLS, were found to be the reasons for this additional delay of 140 μs.

Unlike the standard WANDA IPC code for wired MSNL connections, resource creation and deletion operations are considerably slower within the mobile communication interface. Table 8.6 summarises such latencies for the mobile connection resources. A module creation involves acquiring the relevant system resources and forking the associated data pumping and management threads. To keep the end-to-end setup delay low, modules were arranged to be created concurrently with the setup procedure. A mobile unit, for instance, creates a mobile module while waiting for the connection reply from the relevant base station. The base and the representative units also create the modules in a similar way.

Latency	Mobile Module	Base Module		Rep. Module	
		Non Co-located	Co-located	Non Co-located	Co-located
Creation (msecs)	3.21	2.72	3.21	2.91	3.47
Deletion (msecs)	3.22	3.02	3.90	1.74	2.52

Table 8.6: Creation and deletion latencies for different data path modules

A module deletion takes place while closing the corresponding mobile connection or deleting an obsolete segment of the connection. These are relatively heavyweight procedures because a deletion involves sanity and consistency checking for all the components within the relevant module. Deletion was implemented in a synchronous fashion where the data pumping and management threads within the relevant module are killed sequentially and the last surviving thread exits after cleaning the module-occupied resources. Even though a closure procedure takes quite a long time, a user is insulated from its adverse effects by transferring the closing functionalities to a separate *mobile closure thread.* On invocation of the *MobileClose* function by an end-user, a mobile closure thread is created; this point onwards the closure thread is responsible for deleting all the present segments and for releasing the reserved resources, associated with the relevant mobile connection. While a complete closing operation at the mobile end takes an average of 3.22 ms, the user level application experiences a delay of only 516 μs to close the connection. The additional 111 μs, compared to the 405.3 μs closing delay for a normal IPC connection within the backbone network (see Section 8.5.1), is caused by the creation latency of the closure thread.

The reasons behind the module creation/deletion performance degradation, for the co-located situations, were again found to be the same as explained for the setup case. The competition between the WildListener threads, responsible for the base and the representative module creation, accounts for this degradation. Note that the presented creation and deletion related numbers are independent of the specific topology and the number of hops associated with a mobile connection.

Block Size (Octets)	Round Trip Delay (msecs)	
	High Priority Data Threads	Low Priority Data Threads
48	2.326	3.393
240	2.971	4.250
480	3.790	5.232
720	4.199	6.061
960	5.099	6.872
1200	6.004	7.742
1440	6.653	8.350
1680	7.348	9.278
1920	8.102	10.130

Table 8.7: Effects of block size and thread scheduling priority on mobile connection round trip delay

Delay

The variation of the round trip mobile connection delay, with different MSSAR block sizes, is outlined in Table 8.7. The *source*, *sink* and the *bouncer* threads were arranged as described in Section 8.5.1 for the delay measurement experiments. Each of the presented figures represents the average computed from the round trip delays, separately measured for 50000 blocks; the corresponding mobile connection is the one through BS_b in topology-II. Note that all the previously mentioned background activities were *on* during these measurements. Effectively, two streams (one forward and one return) were present with 40 ms inter block transmission time at the sender end.

Mobile Module Delay (μsecs)	Base Module Delay (μsecs)		Rep. Module Delay (μsecs)	
	Non Co-located	Co-located	Non Co-located	Co-located
240.23	393.84	647.23	368.99	557.87

Table 8.8: Delay offered by different data modules

As expected, the bigger size blocks experience larger delay which is partially caused by the high segmentation and reassembly overhead. The user space gateway, base and the representative modules also cause each block to be copied from the kernel to the user space and back - resulting higher delay. More about these were discussed in the delay measurement subsection of Section 8.5.1. The effects of data thread priority variation on the round trip delay are outlined in Table 8.7. The evident difference of performance strongly advocates for assigning the data threads a scheduling priority higher than the default value.

Block Size (Octets)	Round Trip Delay (msecs)	
	Non Co-located Base and Rep. Modules	Co-located Base and Rep. Modules
48	2.326	4.598
480	3.790	5.853
960	5.099	6.938
1440	6.653	8.166
1920	8.102	9.474

Table 8.9: Round trip delay variation with co-located and non co-located modules

The three-hop round trip delay over a wired connection, with high priority data threads and the minimum block size (48 octets), was found to be 1.907 ms in Table 8.4. An equivalent (in terms of number of hops) mobile connection offers a round trip delay of 2.326 ms to the same size blocks. Calculations reveal that each of the base and representative modules imposes an average of 105 μs additional delay compared to the gateway modules (which was computed as 287 μs during the wired connection delay measurement). A greater number of data pumping threads within these modules account for this additional delay. These results are further verified from the delay figures for different datapath modules, shown in Table 8.8. The effects of the relative location of the base and the representative modules are summarised both in Tables 8.8 and 8.9. Note that the delays, shown in Tables 8.8, correspond to 48 octet block size and both the Tables 8.8 and 8.9 contain results obtained in a high data thread priority situation.

Jitter

The inter block arrival time distributions for the mobile connections are presented in Table 8.10, Figure 8.16 and Figure 8.17. Performance was measured for the three hop connection through the base station BS_b in topology-II. The source mobile unit sends MSSAR blocks once in every 40 ms and the corresponding inter block arrival time is measured at the sink end. These inter arrival times were observed for 50000 blocks and the corresponding distributions are plotted as histograms with a group-interval of 40 μs.

As explained in Section 8.5.1, the jitter through a gateway module is caused largely because of the frequent context switching of the relevant data pumping threads. Because of the use of a greater number of forwarding threads on the datapath, the problem is worse within the base and the representative modules. This is one of the main causes of higher mobile stream jitter, compared to its non-mobile counterpart. The CPU contention, caused by the background threads, also contribute to this large variation of inter arrival time.

Block Size (Octets)	Minimum (msecs)	Maximum (msecs)	Std. Dev. (msecs)	95th Centile (msecs)	99th Centile (msecs)
48	39.061	40.902	0.075	0.179	0.732
480	38.931	41.137	0.083	0.228	0.805
960	38.622	40.823	0.095	0.336	0.879
1440	38.818	40.938	0.128	0.472	1.116
1920	38.896	40.652	0.266	1.020	1.296

Table 8.10: Inter arrival time distribution with varying block size

The influence of block size on stream jitter is summarised in Table 8.10. The distributions for the two extreme block-sizes are also plotted in Figure 8.16. Note that during these jitter experiments, the kernel timer interrupt interval was set as 2 ms and all the background threads were as usually *on*; the data handling threads were running with a priority higher than the default value.

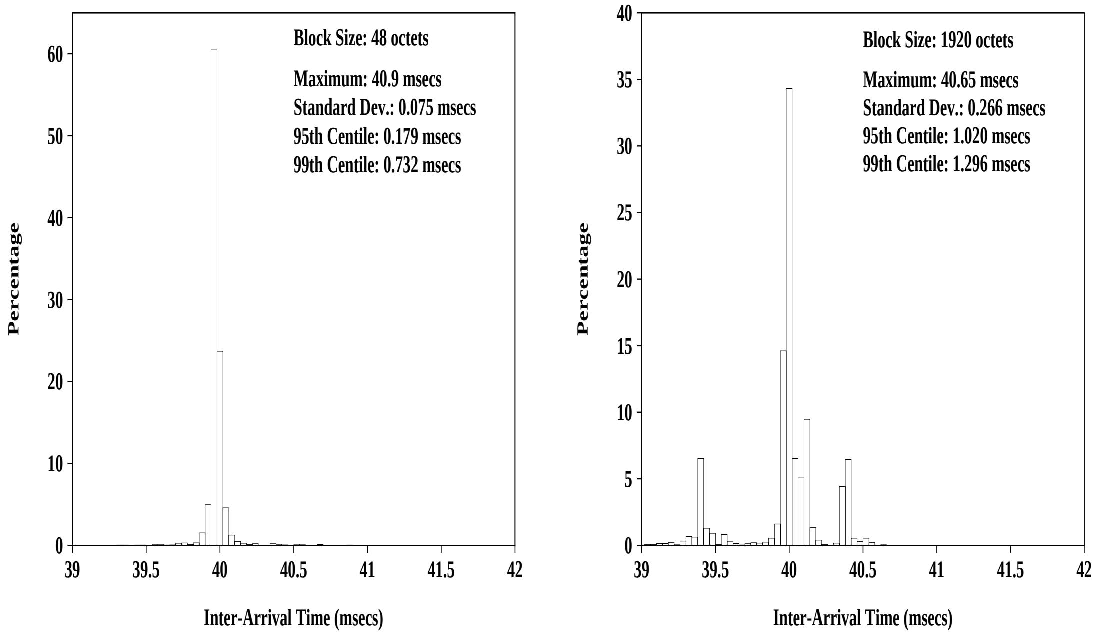


Figure 8.16: Effects of block size on stream jitter over a mobile connection

The results show that the increment of block size causes a steady rise of the 99th centile of the inter arrival time distribution. This can be understood as follows: within any datapath module (mobile, gateway, base or representative), once a pumping thread is scheduled for execution, all the other data handling threads have to wait till a block is completely shipped by the currently active thread. With larger blocks, this effect becomes more prominent and as a result the stream jitter increases substantially.

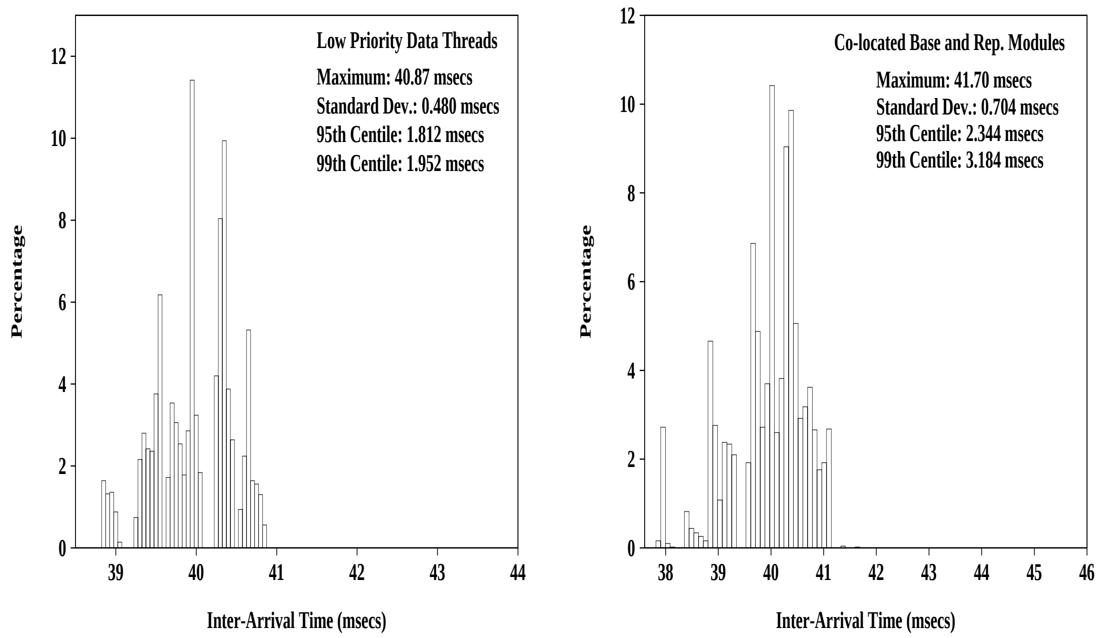


Figure 8.17: Jitter performance degradation with lower data thread priority and CPU contention by co-located base and representative modules

Note that the inter arrival distributions, shown in Figure 8.17, correspond to a video stream with 1920 octet block size and 40 ms inter block arrival. A reduction of the data thread priority requires the corresponding threads to compete with all the other existing threads, including the background ones. This contention is quite evident from the poor jitter performance shown in the first graph of Figure 8.17. The second histogram of the same figure shows the jitter performance degradation caused by the co-located base and representative modules.

8.5.3 Handover and Service Disruption

This section outlines the segment management and the handover performance of the implemented mobile connection management scheme. The presented performance was measured for the experimental mobile connections, with topologies as indicated in Figure 8.15.

Segment Management

Table 8.11 shows the off-line segment setup latency for variable length mobile connections; the corresponding management procedures are described in Section 7.6. In order to estimate the effects of base-to-representative hop-count variation on segment setp latency, two different connection topologies were studied. The experimental steps, involved in the segment setup delay measurement for topology-III, are as follows.

1. Create an *initial path* (see Section 7.5) through BS_c.

2. Connect the mobile unit (with segment management enabled) to the base station BS_b with an emulated radio link. This creates the segments MU-BS_b and BS_b-MR.

3. The segment and handover manager thread (see Figure 8.5) is programmed in such a way that it deletes these newly created segments after a second of their creation. These alternate creation and deletion procedures are allowed to continue for 1000 cycles and the relevant latencies for each segment creation are recorded; the computed average values are quoted here. Note that each of the segment setup delay entries in Table 8.11 represents the combined setup time for the MU-BS_b and BS_b-MR segments.

Table 8.11 also furnishes the BS-CLS connection setup and location information access delays. As explained in Section 8.5.2, the BS-CLS connection setup is required only when the relevant base station does not have a location query connection with the CLS; normally, this happens for the first time when a segment is created through the base station. Similarly, with location information caching enabled within the base stations, the location access overhead is present only after a migration of the mobile representative and during the first segment creation.

Topology	BS to MR Hop-Count	Segment Setup (ms)	BS to CLS Setup (ms)	Location Access (ms)
III	1	12.09	5.41	1.39
IV	2	18.13	5.21	1.48

Table 8.11: Off-line segment setup and the associated delays

A similar experiment was carried out for topology-IV where the base-to-representative distance is 2 hops. When compared to the *initial path* setup latencies, furnished in Table 8.5, the relatively smaller values of the segment setup delays indicate the advantages of the current segment caching scheme. When the mobile unit moves from base station BS_c to BS_b in topology-III, a new pair of segments are created within 12.09 ms; this should be contrasted with the corresponding end-to-end setup time of 18.16 ms. Since the segment setup time depends only on the base-to-representative hop-count, this difference of 6.07 ms is expected to be multiplied for the mobile connections, with larger representative-to-destination distances. The advantages of smaller segment setup latency will be more apparent while studying the handover performance in the following sections. In order to achieve low setup delay, the threads involved in segment management activity were assigned a scheduling priority which is higher than the default value; to maintain low end-to-end delay however the data handling threads were scheduled with priorities higher than those for the segment management threads. A more detailed and complete picture of the priority levels assigned to different classes of threads, present in the current prototype system, is illustrated in Figure 8.18.

Service Disruption

The major service disruptions, caused by a handover, are tabulated in Table 8.12. In order to study the asymmetric upstream and downstream disruptions, two opposite direction video streams were used for this set of experiments. It was explained in Section 7.7 that while the upstream data is held only within a mobile unit, the downstream traffic have to wait both within the corresponding representative and the base station queues during a *with hint* handover. These give rise to the requirements for separate downstream disruption measurement at the relevant base and representative host entities.

In order to measure the handover disruption within a mobile unit, two time-stamps are put on each block of data; once during its insertion within the outgoing I/O buffer queue and then while taking it out from the same queue. The difference in these stamp values indicates the time spent within the buffer. This latency is measured initially, in the absence of any handover, for 50000 blocks and the maximum value is recorded. After the handovers are activated, any such latency, which is larger than the previously recorded maximum, is interpreted as the upstream handover disruption. Measurements were taken for 5000 handovers to compute the average. The same procedure was used for measuring the downstream disruptions caused within the base and the mobile representative entities. The technique, described in Section 8.4.5, was used for emulating the handovers; the model parameters were chosen in such a way that on an average, handovers occured once in every second. The results presented in the forthcoming tables indicate the service disruption and required buffering during the *with hint* handovers, as described in Section 7.7. The performance indexes for the relatively less probable *without hint* handovers are summarised in Figure 8.21.

Disruption Type	Data: +4 Handover: +4	Data: +4 Handover: +5	Data: +5 Handover: +4
Upstream at MU (ms)	7.888	6.851	8.611
Downstream at MR (ms)	0.889	0.731	0.923
Downstream at New BS (ms)	2.963	2.852	3.152

Table 8.12: Handover service disruption with varying scheduling priorities

Table 8.12 presents the up and the downstream handover disruption figures for a mobile connection with topology-III. Measurements were taken only for the handovers with BS_b as the *new*[14] base station. The kernel clock interrupt interval was set at 2 ms during these measurements. In order to decide about an optimum thread priority hierarchy, the handover

[14]When a mobile connection is handed over from BS_c to BS_b, the base station BS_b is referred to as the *new* base station.

latencies were measured with varying priorities assigned to different types of threads. Initially, a set of signalling threads, responsible for handover operation, were identified. It was realised that in order to expedite a handover operation, those *handover signalling* threads require to preempt all the other background and data handling threads. Until this point all the experiments were performed with the maximum priority assigned to the data handling threads. Three different priority settings are illustrated in Table 8.12. Note that a positive priority identifier indicates a priority level with respect to the default user priority *zero*.

Priority	Thread Function
+5	Handover Signalling
+4	Data Forwarding
+3	Segment Management
+2	Mobile Connection Signalling Datapath Module Creation
+1	Mobile Connection Closure Datapath Module Deletion
0	Reconfiguration Service Location Service

Figure 8.18: Levels of thread scheduling priority

As expected, with higher priority handover signalling threads, the handovers perform better than both in the equal priority and the higher data priority situations. It is worth comparing these handover disruption figures with the connection and segment setup delays presented in Tables 8.5 and 8.11. An end-to-end setup of the same connection through BS_b of topology-III takes 18.16 ms in steady state. Following the mobile movement, a pair of segments need only 12.09 ms to be created by the off-line segment manager thread. The current segment caching scheme, with high priority handover signalling, offers a maximum upstream disruption of 6.851 ms for an equivalent mobile connection. This indicates a 62.27% reduction of latency when compared to the scheme where an end-to-end network layer connection has to be setup during each handover. Note that because of the unequal values of the upstream and the total downstream disruptions (3.583 ms for the high handover signalling priority case) bi-directional interactive traffic will suffer in an asymmetric fashion. One possible remedy is to use a sophisticated synchronisation scheme [Sreenan 93], based on the specific quality of service requirements. More about the reasons for these asymmetric disruptions was discussed in Section 7.7.

Guided by these and the previous results, the user threads in the current prototype system are assigned a priority structure as depicted in Figure 8.18. While the background activities, like topology reconfiguration and location services are performed with default user space priority, the segment mangement and setup threads are scheduled with a priority just lower than that of the data handling threads. A mobile connection setup involves datapath module creation and those operations need to be assigned priorities higher than the default value. Mobile connection closure and the datapath module deletion activities, on the other hand, are considered to be less urgent and that is why they are assigned a priority lower than the creation ones.

The next experiment was carried out for studying the effects of base-to-representative hop-count on both the up and the downstream disruptions. The second column of Table 8.13 represents the disruptions measured for topology-III with BS_b as the *new* base station. The first column, on the other hand, indicates the disruptions for the same topology but BS_c as the *new* base station. Similar measurements were taken for topology-IV (with BS_b as the *new* base station) and the corresponding disruption latencies are tabulated in column three.

It is interesting to note that the downstream latencies (both within a base and a representative) do not have any specific dependency on the BS-MR distance. This can be understood from the handover signalling sequence, illustrated in Figure 7.5. No signalling message transaction, which is responsible for the downstream disruptions, takes place between the base and the representative entities during a handover procedure. This makes the downstream handover latencies independent of the base-to-representative distance. The upstream traffic disruption however increases with the hop-count between the representative and the relevant base station. In the current setup, with a priority structure, as shown in Figure 8.18, a single hop increment (from 1 to 2) causes an additional upstream disruption of 191 μs. High handover signalling and data handling priorities account for this small delay when compared to the equivalent signalling latencies for the end-to-end connection and the segment setup procedures.

Disruption Type	BS-MR Hop-Count: 0	BS-MR Hop-Count: 1	BS-MR Hop-Count: 2
Upstream at MU (ms)	5.487	6.821	7.012
Downstream at MR (ms)	0.753	0.728	0.721
Downstream at New BS (ms)	2.257	2.874	2.773

Table 8.13: Effects of base-to-representative hop-count on handover disruptions

Unlike the mobile connection setup and the associated datapath module creation procedures (see Section 8.5.2), handover signalling does not have much room for concurrent execution within the base and the representative entities. An obvious advantage of this is the low upstream disruption when the *new* base station and the relevant representative host-entity are co-located within a single machine. This is evident from the low upstream disruption of 5.487 ms, as indicated in the first column of Table 8.13.

Variation of the upstream and the downstream handover disruptions with varying kernel clock interval frequency are shown in Table 8.14. The corresponding experiments were carried out on topology-I with a video stream handover, occurring between the base stations BS_a and BS_b. Although the results do not show any distinct effect of interrupt frequency on the downstream disruptions, the upstream counterpart of it changes considerably with varying kernel timer interrupt interval. At the lower end of the clocking interval (from 0.5 ms to 2.0 ms) disruption reduces with decreasing interrupt frequency. Lesser ISR overhead on the CPU accounts for this performance improvement. Any further reduction of interrupt frequency beyond this point (interval greater than 2 ms) causes the upstream disruption to deteriorate. Investigation of the handover signalling code revealed the following cause of this unexpected performance.

Kernel Clock Interrupt Interval (ms)	Upstream Disruption at MU (ms)	Downstream Disruption at MR (ms)	Downstream Disruption at BS (ms)
0.5	8.015	0.771	2.875
1.0	7.515	0.687	2.781
2.0	6.816	0.712	2.866
4.0	11.413	0.723	2.810
8.0	15.442	0.691	2.771
16.0	33.651	0.678	2.502

Table 8.14: Varying kernel clocking frequency and its effects on handover service disruptions

In order to reduce the system resource consumption, only one thread was employed in a representative unit to handle the handover messages coming through all the signalling connections with the relevant base stations. This required the thread to execute a polling loop in order to detect the incoming messages through all such signalling connections. Polling was implemented with a *WandaTimeP* system call, on execution of which the invoking thread sleeps for a specific time before doing its next task. The minimum possible sleeping period is dictated by the kernel timer interrupt interval. Therefore, with infrequent kernel timer interrupts the polling speed suffers considerably and, as a result of it, the processing of the incoming handover signalling messages from the base stations get deferred - causing high upstream disruption. Since this particular thread within the representative is not involved in the other parts of a handover procedure (see Figure 7.5), the downstream disruptions are insulated from this *polling degradation* effect. Upstream handover latencies in Table 8.14 show a distinct minimum of 6.816 ms with 2 ms of kernel clocking interval. This is why in most of the experiments, described in the present and the previous sections, this particular kernel clocking frequency was used.

Buffering

The next experiment measures handover buffering requirements at the different network entities. The length of the I/O buffer queues within the mobile, base and the representative modules were measured during 5000 separate handovers, occurring between the base stations BS_a and BS_b of topology-III. The average inter handover interval was set sufficiently long (1 second) in order to reduce the interference between successive handovers. Queue length (in Kbits) distributions, for a stream with a datarate of 384 Kbits/s[15], are illustrated in Figures 8.19 and 8.20. The chosen group-intervals are 100 and 120 bits for the mobile and the base module queue length distributions. A lower value (2 bits) is chosen for the representative case in order to highlight the relatively smaller buffer queue length variation within a representative module.

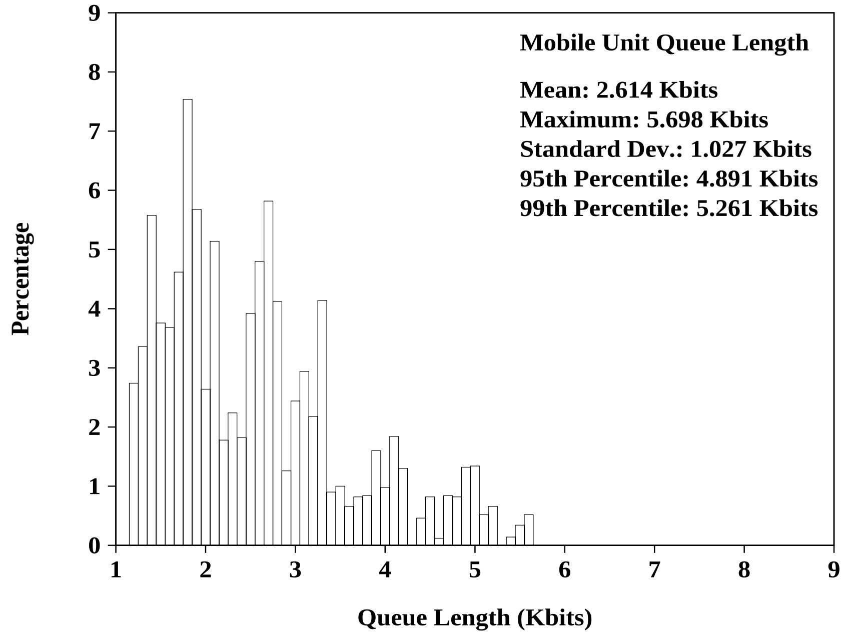


Figure 8.19: Upstream handover buffering requirement in an end mobile unit

As expected, the amount of buffering needed within the end mobile unit is much more than its base and representative counterpart. Although the observed maximum queue length is almost 5.7 Kbits, the mobile unit can handle 95%[16] of the handovers with a buffer space of 4.9 Kbits. This gives a rough estimate of the packet dropping probability for a maximum buffer size and specified datarate. Since the queue lengths depend primarily on the handover latency and the stream datarate, the relative positions of the base and the representative

[15] A block size of 48 octets and an inter block transmission time of 1 ms have been used. It is worth noting that the measured queue lengths are always in multiples of the used block sizes. This restriction was imposed by the block forwarding mechanism, used by all the user space datapath modules. To achieve the maximum resolution, the minimum possible block size (48 octets) is chosen for all the current experiments.

[16] Not to be confused with the *centile* figures used for jitter representation, in the earlier sections of this chapter.

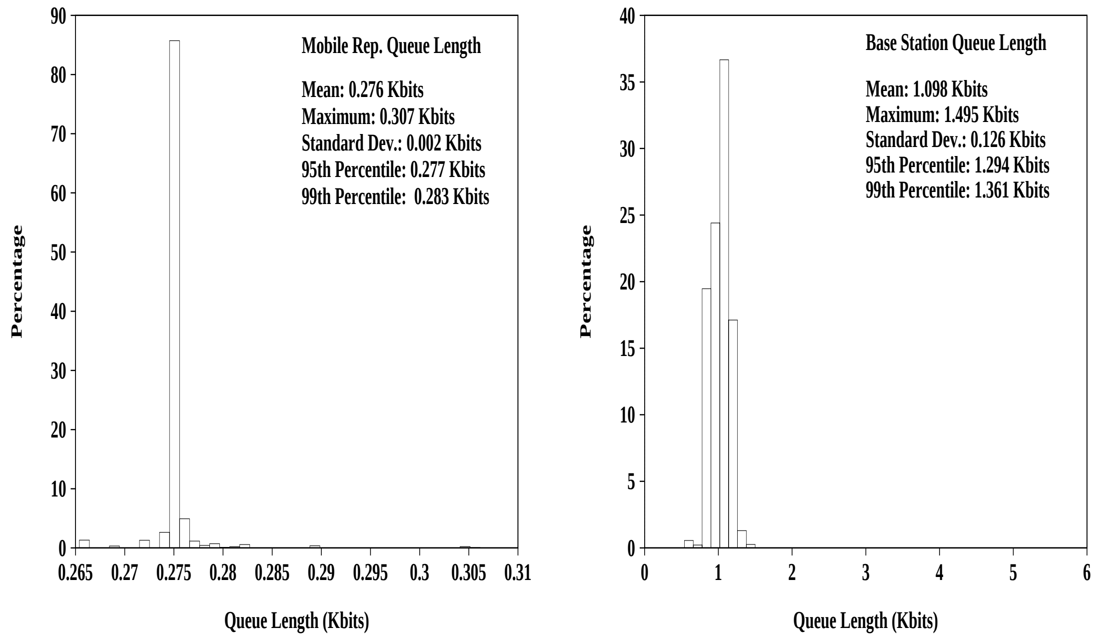


Figure 8.20: Downstream handover buffering in base station and mobile representative

host-entities affect only the mobile unit buffering requirements.

8.5.4 With and Without Hint Handovers

All the presented handover performance figures correspond to a *with hint* scenario where all the necessary segments are created before the actual handover procedure is started by the end mobile unit, possibly because of the deterioration of an active radio link quality. On the other hand, in a *without hint* case the mobile is forced to start a handover[17] followed by the sudden failure of a radio segment, which is a part of the *active path* of the relevant mobile connection. This requires the mobile unit to wait for the link failure detection before it can start the handover signalling procedure. The delay in link failure detection gives rise to the differences between the *with* and *without* hint handover performance figures, highlighted in Figure 8.21. Note that the radio link failure detection delay depends very much on the particular implementation. It is possible for the MAC layer to inform the connection manager code about the failure when the former detects an unusually large error rate on a radio link. In the current implementation however this information was extracted from the location server and fed into the mobile module as a *link status change* event (see Figure 8.5). Since these events are generated by the location server code, the resolution of the detection procedure is eventually determined by the broadcasting frequency of the radio location beacons (see Section 6.8).

Note that the mentioned packet loss figures in Figure 8.21 correspond to infinite buffer

[17]Detailed discussions about these procedures can be found in Section 7.7.

	With Hint Handover	Without Hint Handover
Upstream Disruption at MU	T_{mu} (Measured)	$T_{muw} = T_{mu}$ + Link failure detection delay at MU
Downstream Disruption at BS	T_{bs} (Measured)	Not Present
Downstream Disruption at MR	T_{mr} (Measured)	T_{mrw}: Starts: On Rx. of 'failure inform' Ends: On Tx. of 'reply'
Upstream Buffering at MU	T_{mu} x Datarate (Measured)	T_{muw} x Datarate
Downstream Buffering at BS	T_{bs} x Datarate (Measured)	Not Present
Downstream Buffering at MR	T_{mr} x Datarate (Measured)	T_{mrw} x Datarate
Upstream Loss at MU	Normally Not Present	Link failure detection delay at MU x Datarate
Downstream Loss at BS	Not Present	Not Present
Downstream Loss at MR	Normally Not Present	(Link failure detection delay at old BS + Transit time of message 'failure inform') x Datarate

Figure 8.21: Performance indexes for *with* and *without* hint handovers

situations and therefore do not include the losses caused by buffer overflow. In a *with hint* case, the upstream mobile and the downstream representative losses are present only when the quality of the radio link between the mobile unit and the old base station deteriorates soon after the handover initiation. A discussion, regarding these, is included in Section 7.7. In a *without hint* handover, no disruption is present within the *new* base station and that is why no base station handover buffering is required. These also eliminate the possibility of any packet loss within a base module.

With a poor quality radio link between the mobile unit and the *new* base station, a considerable number of handovers fail because of delayed signalling responses. The situation is worse when the handled traffic requires strict upperbounds of the handover disruption latencies to be maintained. The handover failure probability [Rappaport 91] in such situations would represent a quality of service measure which is suitable for the current mobile connection management scheme. The amount of excess resources, occupied by multiple segments [Keeton 93] of a single mobile connection, is also a useful index of resource utilisation efficiency of the connection manager. The schemes for measuring these handover performance figures could not be implemented because of the limited availability of time.

8.6 Summary

This chapter provided the implementation details of a prototype version of the proposed mobile location and connection management schemes. An overview of the target environment was followed by the experimental programme, designed for an evaluation of the prototype system.

While the present prototype implementation ensured the feasibility of the proposed location and connection management schemes, it also aided the identification of the performance-critical parts of the design. The implementation of the block forwarding datapath modules in user space was proved to be one of the principal bottlenecks of the system. In order to achieve better performance, all the mobile, base, gateway and the representative module functionalities are required to be transferred within the kernel. An optimum priority hierarchy, for scheduling the WANDA threads of different classes, has been established through experimentation. There exist some further scopes for QoS based thread priority assignment. Although not very obvious from the design, co-location of the base and the representative modules was found to be a major reason for mobile connection performance degradation. These results aid the design of a judicious location server heuristic for choosing the representative host-entities.

Although the current handover mechanism is able to provide satisfactory disruption and buffering performance, the scheme puts too much computing and resource (e.g. I/O buffer, CPU) burdens on the mobile units. A careful revision of the connection management policy is recommended in order to transfer the computation and resource intensive parts of the protocols from the mobile units to the wired network entities. Also, a remote procedure call (RPC) based implementation of the location and the connection management operations would be more efficient and portable compared to the current message passing based system.

Chapter 9

Conclusions and Future Work

9.1 Conclusions

This dissertation has addressed various issues regarding a mobile radio LAN architecture design. In order to support continuous media traffic in a mobile environment, a connection-oriented approach has been adopted. The motivations behind this were the requirements for maintaining performance compatibility between the existing ATM based BISDN systems and the third generation Personal Communication Networks.

In order to implement a fast topology reconfiguration facility for the point-to-point backbone of the proposed wireless LAN, an incremental dynamic reconfiguration algorithm has been proposed, where a network topology change initiates the *minimum tree* (Chapter 3) maintenance procedures while not interrupting the ongoing ATM connections. In addition to the low reconfiguration latency, it has been demonstrated that the spatial locality of the proposed protocol can be exploited for implementing simultaneous connection routing and topology reconfiguration operations at different parts of the backbone network. A prototype version of the protocol was implemented for demonstrating its feasibility in an ATM environment.

CSMA/AED, a single frequency and carrier sensing based protocol, was proposed for the radio medium access operations. Simulation results, in Chapter 4, have shown that this statistical and reliable access scheme can meet the necessary realtime performance requirements in the proposed radio network. The effectiveness of a per-connection window based flow control mechanism, for the proposed radio LAN, was also investigated in Chapter 5. A hybrid technique has been used, where the medium access and the radio data-link layers were modelled using the mentioned simulator. An upper layer end-to-end queueing model, involving the flow dependent radio MAC servers, was solved using an approximate Mean Value Analysis technique which was augmented for faster iterative convergence. The results, obtained from the model, showed that the impact of the mobile application service times, on the network performance, are quite significant. It was also observed that an increase of the application service time reduces the delivery delay over radio (by shifting the network bottleneck from the radio data-link layer to the application layer), at a cost of reduced overall throughput.

With an aim for managing location information efficiently, a distributed location server architecture has been proposed in Chapter 6. It has been identified that an elegant way of handling mobility is to put most of the mobility management functionalities at the network layer. By keeping the upper layers insulated from the effects of mobility, this approach allows most of the existing applications to be ported into the mobile world without any major changes. As an extension of this mobility isolation strategy, the concept of a per-mobile software agent, known as a *representative*, has been introduced. It was shown that with the assistance of these representatives, along with an appropriate location management strategy, it is possible for two network entities to communicate without knowing the location of their respective peer entities. The use of a per-mobile partitioning scheme has been proposed where the network is partitioned based on the *mobility profiles* of the individual mobile users. It has been argued that the caching schemes, based on this partitioning strategy, would greatly enhance the location server performance.

User mobility requires a different approach for handling the network layer connections in a mobile environment. This motivated the design of a mobile connection manager which aims to provide the standard ATM functionalities over the mobile connections. It has been shown in Chapter 7 that by adopting a suitable combination of *source* and *hop-by-hop* routing, it is possible to successfully route a mobile connection to a roaming destination entity. This, of course, requires assistance from the representative of the destination mobile entity and from the location server, in general. Connection *handover* was identified as one of the most important performance-critical functions of the connection manager. While maintaining the end-to-end network layer connectivity, a handover needs to meet the realtime performance requirements of the transported continuous media streams. This was achieved by using a novel *connection caching* scheme where the relevant parts of a mobile connection are setup in a predictive manner, even before the handover operation is initiated. This approach reduces the handover latency by avoiding the requirements for extensive online signalling during a handover. In addition to that, by isolating the handover signalling at both the ends of a connection, scalability is achieved. It was also shown that unlike the other existing approaches, this scheme allows one end of a mobile connection to be handed over while keeping it completely transparent to the other end of the connection; this makes the *both end* simultaneous handovers possible without any additional signalling overhead.

A prototype system was implemented for investigating the feasibility and the performance of the designed location and connection managers. The details of the implementation were presented in Chapter 8. It was built in a point-to-point ATM environment where the management functions were implemented as user space processes on a locally developed micro-kernel. The wireless data-link and user mobility were emulated on co-axial cables, using an experimentally obtained in-building mobility model. A message passing based location server implementation confirmed the effectiveness of the representatives for the proposed mobility management functions. The signalling and datapath performance of the mobile connections have been evaluated experimentally. The performance figures clearly displayed the advantages of the proposed connection caching approach for implementing the handovers which are capable of supporting performance-critical continuous media streams. This set of experiments has also been used for pinpointing the performance-critical parts of the proposed design.

9.2 Future Work

Quality of Service Guarantee: While not making detailed recommendations about the mobile QoS implementation techniques, at different stages, this dissertation tried to identify the crucial design decisions which affect the QoS guarantee in a mobile network. It has been realised that in the context of mobility it is necessary to add and modify the existing quality of service parameters which were originally defined for the wired networks. This is an important area for future work where thorough investigation is needed in order to provide per-mobile-connection quality of service guarantee. Predictive resource allocation, based on the per-user *mobility profile* information, is considered as one of the probable means for mobile QoS guarantee. In the framework of such a statistical reservation scheme, the problem of finding an optimum resource utilisation strategy can be pursued as an interesting modelling exercise.

Alternative Handover Protocols: The dissertation shows that there is scope for more work to be done in the area of handover algorithms. The alternative protocols need to transfer more functionalities to the wired network in order to improve the mobile power consumption and the radio resource utilisation efficiency. It is also envisaged that the design of an efficient handover protocol would largely depend on the new ATM adaptation layer protocols which are tailored for the mobile networks.

Supporting Multicast: Multicast in ATM networks requires to support connections with multiple end entities. The presented connection manager can be extended by adding the multicast facilities for the mobile ATM connections. It is realised that the present approach of representative mediated connection management is particularly suitable for supporting mobile multicast. Multiple mobile end entities can be dynamically joined with an existing mobile connection by adding their fixed representatives, much like in a wired multicast case. However, the implementation of QoS over multicast connections would require a complicated resource allocation strategy.

GPS Interface: With the rapid advancement of the Global Positioning System (GPS) technology, the size of a minimal GPS receiver is brought down into a level where it is quite realistic to attach a GPS receiver to a roaming mobile entity. This enables the mobile unit to be informed by the relevant satellites about its accurate geographical position. Location management, based on the GPS information, would require a different approach where the mobile units can directly provide their location information to the backbone network. This should be contrasted with the present cellular locationing approach where a roaming entity's location information is represented in terms of its neighbour base stations. In addition to the location information, the precise GPS timing facilities would also enhance the continuous media stream synchronisation services. There is scope for future research in the area of GPS assisted mobile location management for the personal communication networks.

Mobile AAL: Since in a wired ATM network it is always assumed that the cells in a stream are delivered in sequence, the ATM adaptation layer does not provide any protection from the cell out-of-ordering. Due to the asymmetric radio link delays, during a handover, cell out-of-ordering may occur quite frequently and according to the current AAL standards (e.g. AAL 5) each cell out-of-ordering causes the corresponding block to be dropped and retransmitted depending on the transported traffic type. The necessity of avoiding this excessive block retransmission gives rise to the requirements for further research in order to come up with a modified adaptation layer protocol, suitable for handling cell out-of-ordering in a mobile environment.

Appendix A

Reconfiguration Protocol

This appendix presents a formal specification of the *Dynamic Minimum Tree Maintenance* (DMTM) algorithm which was discussed in Chapter 3. The protocol, specified in a pseudo *C* language, has three principal components, namely the *Add Tree*, the *Time-Out* and the *Delete Tree* procedures.

On reception of any one of the *construct*, *expand* or *repair* messages, the receiver entity $Node_r$ executes the *Add Tree* procedure. The *Delete Tree*, on the other hand, is invoked when $Node_r$ receives a *prune* message. A time-out of a reconfiguration connection triggers a node to execute the *Time-Out* procedure.

Add Tree Procedure:

$Node_r$ has received a reconfiguration message $Msg(src, type, node_{rem}, nif_{rem}, ver, hops, nif_{local})$ where $Msg.type$ indicates to *construct*, *expand* or *repair*.

if an $Entry(Node_{rem}, Nif_{rem}, Version, Hops, Nif_{local})$ is found in the connectivity database of $Node_r$, so that -

$((Entry.Node_{rem} == Msg.node_{rem})$ &&
$(Entry.Nif_{rem} == Msg.nif_{rem})$){

switch $(Msg.type)${

case $repair$:
case $construct$:

if $((Msg.ver > Entry.Version)$ ||
$((Msg.ver == Entry.Version)$ && $(Msg.hops < Entry.Hops)))${

$Entry.Version = Msg.ver;$

$Entry.Hops = Msg.hops;$

}

break;

case *expand*:

if $(Msg.hops < Entry.Hops)\{$

$Entry.Version = Msg.ver;$
$Entry.Hops = Msg.hops;$

}

break;

}

}

else{ /* no such entry found */

Create a new *Entry* with -

$Entry.Node_{rem} = Msg.node_{rem}$
$Entry.Nif_{rem} = Msg.nif_{rem}$
$Entry.Version = Msg.ver$
$Entry.Hops = Msg.hops$
$Entry.Nif_{local} = Msg.nif_{local}$

and put it into the connectivity database.

}

/* It is necessary to check if the message needs to be forwarded. */

if $((Msg.type == repair)$ &&
($Msg.src$ is one of the immediate neighbours)){

Terminate the *Add Tree* procedure. Any further forwarding may cause looping or oscillation.

}

if ((A new *Entry* was created for the connectivity database) ||
((an *Entry* was updated) && $(Msg.hops > 1)$)){

/* suitable condition for forwarding the message */

for (all the available network interfaces nif_j, where

$(nif_j \neq Msg.nif_{local})$ &&
$(Msg.src \neq$ the node at the other end of the link
connected to $nif_j))\{$

Create a message $FwdMsg$ with -

$FwdMsg.src = Msg.src$
$FwdMsg.type = Msg.type$
$FwdMsg.node_{rem} = Msg.node_{rem}$
$FwdMsg.nif_{rem} = Msg.nif_{rem}$
$FwdMsg.ver = Msg.ver$
$FwdMsg.hops = Msg.hops + 1$
$FwdMsg.nif_{local} = NULL$ /* to be filled up by the receiver */

and forward the message through the reconfiguration connection on nif_j.
}
}
if $(Msg.hops == 1)\{$ /* message originator is a neighbour */

Reschedule the *timeout* event for the reconfiguration connection on $Msg.nif_{local}$.
if (Msg is the first message through the reconfiguration connection
on $Msg.nif_{local})\{$

/* the link on $Msg.nif_{local}$ is just created; it may be necessary to */
/* expand few trees. */

for (all the entries $Entry_k$ in the connectivity database, so that -
$(Entry_k.Nif_{local} \neq Msg.nif_{local})$ &&
$(Entry_k.Node_{rem} \neq$ the node at the other end of the
link connected to $Msg.nif_{local}))\{$

Create a message $ExpandMsg$ with -

$ExpandMsg.src = Node_r$
$ExpandMsg.type = Expand$
$ExpandMsg.node_{rem} = Entry_k.Node_{rem}$
$ExpandMsg.nif_{rem} = Entry_k.Nif_{rem}$
$ExpandMsg.ver = Entry_k.Version$
$ExpandMsg.hops = Entry_k.Hops + 1$
$ExpandMsg.nif_{local} = NULL$ /* to be filled up by the receiver */

and forward it through the newly created connection on $Msg.nif_{local}$.
}

}
}

Time-Out Procedure:

Time-out of the reconfiguration connection through nif_t of $Node_r$ is interpreted by the node as an indication of the deletion of the link on that interface.

for (all the entries $Entry_k$ in the connectivity database, so that
$(Entry_k.Nif_{local} == nif_t)$){

for (all the available reconfiguration connections on nif_j, where
$(nif_j \neq nif_t)$ &&
($Entry_k.Node_{rem} \neq$ the node at the other end of the
link connected to nif_t)){

Create a message $PruneMsg$ with -
$PruneMsg.src = Node_r$
$PruneMsg.type = Prune$
$PruneMsg.node_{rem} = Entry_k.Node_{rem}$
$PruneMsg.nif_{rem} = Entry_k.Nif_{rem}$
$PruneMsg.ver = Entry_k.Version$
$PruneMsg.hops = Entry_k.Hops$

$PruneMsg.nif_{local} = NULL$ /* to be filled up by the receiver */

and forward it through the reconfiguration connection on nif_j.
}

delete $Entry_k$ from the connectivity database of $Node_r$.
}

Delete Tree Procedure:

$Node_r$ has received a reconfiguration message $Msg(src, type, node_{rem}, nif_{rem}, ver, hops, nif_{local})$ with $Msg.type == prune$. It may be necessary to prune few trees which are affected by the *Link Deletion* event which caused the origination of the current message. Simultaneous repairs are also needed to maintain the minimum hop trees as described in Section 3.3.5.

if an $Entry(Node_{rem}, Nif_{rem}, Version, Hops, Nif_{local})$ is found in the connectivity database of $Node_r$, so that -

$((Entry.Node_{rem} == Msg.node_{rem})$ &&

```
    (Entry.Nif_rem == Msg.nif_rem) &&
    (Entry.Nif_local == Msg.nif_local)){

    /* pruning is necessary; check the conditions for forwarding */

    if (Msg.ver ≥ Entry.Version){

        for (all the available reconfiguration connections on nif_j, where
            (nif_j ≠ Entry.Nif_local) &&
            (Entry.Node_rem ≠ the node at the other end of the link
            connected to nif_j)){

            Create a message FwdMsg with -

            FwdMsg.src = Msg.src
            FwdMsg.type = Msg.type
            FwdMsg.node_rem = Msg.node_rem
            FwdMsg.nif_rem = Msg.nif_rem
            FwdMsg.ver = Msg.ver
            FwdMsg.hops = Msg.hops + 1
            FwdMsg.nif_local = NULL   /* to be filled up by the receiver */

            and forward it through the reconfiguration connection on nif_j.
        }

        delete Entry from the connectivity database of Node_r.
    }
}
else{       /* no pruning was necessary; check any necessity of repair */

    if  an Entry(Node_rem, Nif_rem, Version, Hops, Nif_local) is found
        in the connectivity database of Node_r, so that -

        ((Entry.Node_rem == Msg.node_rem) &&
        (Entry.Nif_rem == Msg.nif_rem) &&
        (Entry.Nif_local ≠ Msg.nif_local) &&
        (Entry.Node_rem ≠ the node at the other end of the link
        connected to Msg.nif_local)){

        /* it is necessary to start a repairing cycle */

        Create a message RepairMsg with -
```

$RepairMsg.src = Node_r$
$RepairMsg.type = Repair$
$RepairMsg.node_{rem} = Entry.Node_{rem}$
$RepairMsg.nif_{rem} = Entry.Nif_{rem}$
$RepairMsg.ver = Entry.Version$
$RepairMsg.hops = Entry.Hops + 1$
$RepairMsg.nif_{local} = NULL$ /* to be filled up by the receiver */

and forward the message through the reconfiguration connection on $Msg.nif_{local}$

}
}

Appendix B

Analytical Derivations

This appendix describes the derivation details of few of the equations, used in Chapter 5. A modified scheme, used for faster convergence of the MVA algorithm, is also presented along with the relevant convergence patterns for the different network bottleneck situations.

B.1 Mean Queueing Time

Equation 5.1, presented in Section 5.3.2 of Chapter 5, is derived in this section. All the notations, introduced in Section 5.3.2, are valid for the current derivations. The mean queueing time $t_i^j(W)$, for a chain j ($j = 1, \ldots, R$) packet at queue i ($i \in Q(j)$), has the following three components.

- The mean service time for the arriving packet.
- The mean service time for all the other packets waiting in the queue.
- The mean residual service time for the packet currently being serviced.

While τ_i^j represents the mean service time for an arriving packet, the service time for all the backlog packets in queue i can be written as

$$\sum_{k \in R(i)} \tau_i^k [n_i^k(W) - \rho_i^k(W)]$$

and the residual service time can be expressed as [Kleinrock 75]

$$\rho_i(W) \sum_{k \in R(i)} [f_i^k(W) \frac{\sigma_i^k(W)}{2\tau_i^k}].$$

Adding all the three components together,

$$t_i^j(W) = \tau_i^j + \sum_{k \in R(i)} \tau_i^k [n_i^k(W) - \rho_i^k(W)] + \rho_i(W) \sum_{k \in R(i)} [f_i^k(W) \frac{\sigma_i^k(W)}{2\tau_i^k}]. \qquad \text{(B.1)}$$

For a delay node, only the first component exists. This leads to the expression

$$t_i^j(W) = \tau_i^j.$$

For a queue with exponential service time, the probability distributions of the service time and the residual service time are the same. Taking this into account, the third component [Mitrani 87] can be written as

$$\rho_i(W) \sum_{k \in R(i)} f_i^k(W) \tau_i^k$$

and the expression for mean queueing time simplifies to

$$t_i^j(W) = \tau_i^j + \sum_{k \in R(i)} \tau_i^k n_i^k(W).$$

For a constant service time queue, the residual service time is

$$\rho_i(W) \sum_{k \in R(i)} f_i^k(W) \frac{\tau_i^k}{2}$$

and the total queueing time can be expressed as

$$t_i^j(W) = \tau_i^j + \sum_{k \in R(i)} \tau_i^k [n_i^k(W) - \frac{1}{2}\rho_i^k(W)].$$

Summarising these three cases, Equation B.1 simplifies to

$$t_i^j(W) = \begin{cases} \tau_i^j & \text{delay node,} \\ \tau_i^j + \sum_{k \in R(i)} \tau_i^k n_i^k(W) & \text{exponential service time,} \\ \tau_i^j + \sum_{k \in R(i)} \tau_i^k [n_i^k(W) - \frac{1}{2}\rho_i^k(W)] & \text{constant service time.} \end{cases} \tag{B.2}$$

B.2 Effects of Background Load

The derivation of Equation 5.8, used for representing the background load in Section 5.3.2, is presented in this section.

If the queueing entity i, in Figure B.1, represents a delay node then the background load does not have any effect on the service time τ_i^j for a chain j packet. The queue i utilisation for the chain j packets can be written as

$$\rho_i^j(W) = \lambda^j(W) \tau_i^j.$$

So the utilisation for queue i is

$$\rho_i(W) = \sum_{k \in R(i)} \rho_i^k(W) = \sum_{k \in R(i)} \lambda^k(W) \tau_i^k.$$

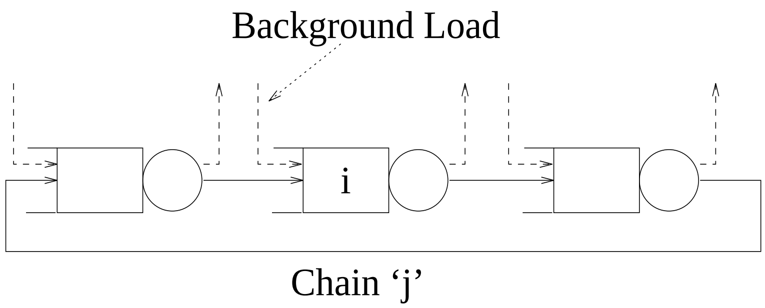


Figure B.1: Background load on chain j at queue i

The population of the chain j packets, in queue i, can be expressed as

$$n_i^j(W) = \frac{\rho_i^j(W)}{1-\rho_i(W)} = \frac{\lambda^j(W)\tau_i^j}{1-\sum_{k\in R(i)}\lambda^k(W)\tau_i^k}.$$

Reformatting the terms of this expression yields

$$n_i^j(W) = \frac{\lambda^j(W)\tau_i^j/[1+\lambda^j(W)\tau_i^j-\sum_{k\in R(i)}\lambda^k(W)\tau_i^k]}{1-\lambda^j(W)\tau_i^j/[1+\lambda^j(W)\tau_i^j-\sum_{k\in R(i)}\lambda^k(W)\tau_i^k]}.$$

Substituting

$$\widehat{\rho_i^j(W)} = \frac{\lambda^j(W)\tau_i^j}{1+\lambda^j(W)\tau_i^j-\sum_{k\in R(i)}\lambda^k(W)\tau_i^k}$$

the expression simplifies to

$$n_i^j(W) = \frac{\widehat{\rho_i^j(W)}}{1-\widehat{\rho_i^j(W)}},$$

where $\widehat{\rho_i^j(W)} = \lambda^j(W)\widehat{\tau_i^j}$. It shows that the effects of the background load from the other chains can be incorporated by modifying the queue i service time expression, for the chain j packets, as

$$\widehat{\tau_i^j} = \begin{cases} \tau_i^j & \text{delay node,} \\ \dfrac{\tau_i^j}{1+\lambda^j(W)\tau_i^j-\sum\limits_{k\in R(i)}\lambda^k(W)\tau_i^k} & \text{otherwise.} \end{cases} \tag{B.3}$$

B.3 Modified Scheme for Fast MVA Convergence

Figure B.2 illustrates the nature of convergence of the iterative solution scheme, presented in Section 5.3.2, in the two extreme bottleneck situations (see Section 5.4). When the network bottleneck is in the application layer, for a convergence error factor threshold (ϵ) of 10^{-6}, the scheme converges rapidly within 40 iterations. In case of radio bottleneck, the solution

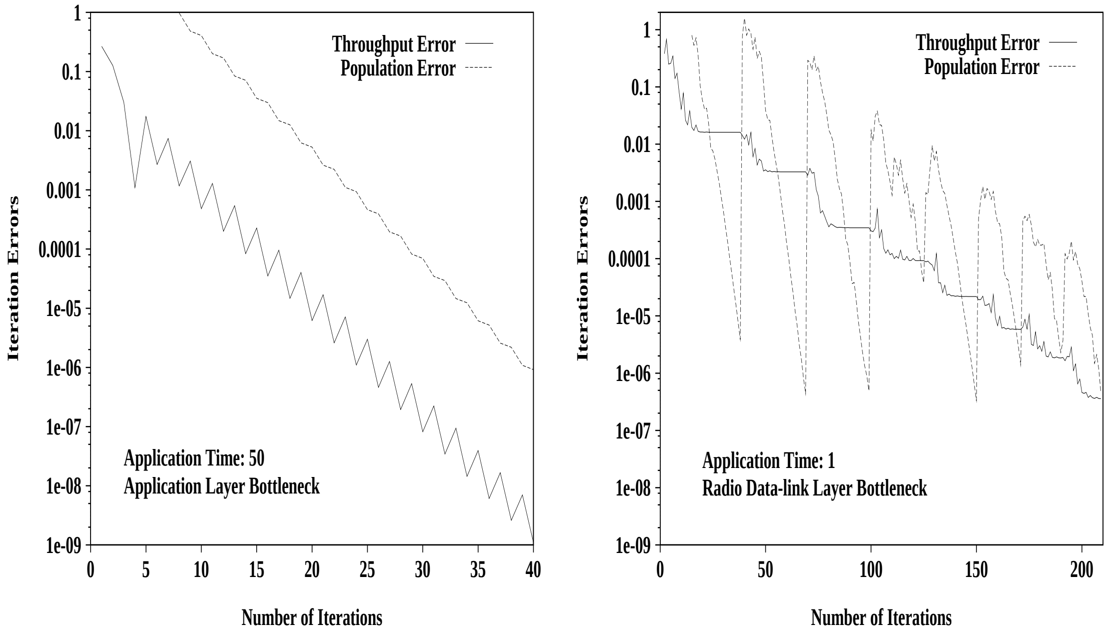


Figure B.2: MVA convergence patterns in different bottleneck situations

needs approximately 210 iterations for both the throughput and the population errors to be less than ϵ.

A closer examination of both the graphs reveals that the errors do not reduce monotonically with the number of iterations. In the radio bottleneck case, the throughput error shows a tendency of stabilising at different iteration values (e.g. error value of 0.019 after 19 iterations). In order to overcome the problems of delayed convergence, caused by this premature error stability, a modified iterative scheme is used. Once an undesired throughput error stability is detected with sufficient accuracy (e.g. after 39 iterations in the radio bottleneck case), a forced convergence [Murata 88] is assured by the following strategy.

Modified Solution Algorithm

1. For all j ($j = 1, \ldots, R$) initialise $\lambda^j(W) = \lambda^j_{base}(W) = \eta$ (e.g. $\eta = 10^{-2}$) and $n^j_i(W)$ for all $i \in Q(j)$. The new set of parameters $\lambda^j_{base}(W)$ are introduced for keeping track of the maximum values of $\lambda^j(W)$, in order to avoid any divergence of the iteration procedure. While initialising, the condition $\sum_{i \in Q(j)} n^j_i(W) = W^j$ for all $j = 1, \ldots, R$ should be satisfied.

2. Using Equation 5.12, compute the service times τ^j_i for all i ($i = 1, \ldots, N$). If a $\lambda^j(W)$ is out of range so that Equation 5.12 cannot be applied, then modify it as $\lambda^j(W) = (\lambda^j_{base}(W) + \lambda^j(W))/2$.

3. For all the chains j ($j = 1, \ldots, R$) run the single chain MVA, using Equations 5.8, 5.9, 5.10 and 5.11. Compute the error parameters $a^j_i(W)$ and $b^j_i(W)$ from the single chain

MVA results and Equation 5.7.

4. Compute the queueing times $t_i^j(W), (i = 1, \ldots, N; j \in R(i))$ from Equation 5.6.

5. The throughput for each chain j ($j = 1, \ldots, R$) is to be computed by using Equation 5.3. The new and the old throughput of chain j are represented by $\lambda_{new}^j(W)$ and $\lambda^j(W)$ respectively. If $\lambda_{new}^j(W) > \lambda^j(W)$ then assign $\lambda_{base}^j(W) = \lambda^j(W)$ and

$$\lambda^j(W) = (\lambda^j(W) + \lambda_{new}^j(W))/2.$$

Otherwise, if abs $(\lambda^j(W) - \lambda_{base}^j(W)) < \zeta$, (for a very small value of $\zeta = 10^{-9}$) then assign

$$\lambda_{base}^j(W) = (\lambda_{base}^j(W) + \lambda^j(W))/2,$$

else

$$\lambda^j(W) = (\lambda_{base}^j(W) + \lambda^j(W))/2.$$

6. Compute the packet population $n_i^j(W)$ $(i = 1, \ldots, N; j \in R(i))$, using Equation 5.4.

7. If the current iteration number is n, a convergence error factor is defined as

$$\pi = \sum_{j=1}^{R}[\lambda^j\{n\} - \lambda^j\{n-1\}] + \sum_{i=1}^{N} \sum_{j \in R(i)} [n_i^j(W)\{n\} - n_i^j(W)\{n-1\}]. \qquad \text{(B.4)}$$

8. If the condition $\pi < \epsilon$ (e.g. $\epsilon = 10^{-6}$) is satisfied then the iteration process has to be stopped; otherwise repeat from step 2.

Appendix C

Location Update and Representative Management

This appendix describes the details of the messages used by the location server for location update and representative management operations, described in Chapter 6. A brief description is also included to highlight the actions taken by the location server in order to handle the error situations caused by the failures of different network entities. Finally, the implementation details of an augmented location update and representative management strategy, incorporating the *mobility profile* based network partitioning schemes (described in Section 6.10), is presented.

C.1 Location Update

This section presents the format and the usage of the update messages which are used for the construction and maintenance of the MU location database, outlined in Section 6.6.3.

C.1.1 Messages

The MU location update scheme uses the following types of messages.

1. **BS-CLS-ADD-HRLI(MU, BS, Age, Link-Qual):** Whenever the link-quality field of a BNT entry changes significantly or a new entry is added to the BNT, this message is sent from the relevant BS to the local CLS.
2. **BS-CLS-DELETE-HRLI(MU, BS)**: On deletion of a BNT entry, a BS sends this message to the local CLS.

   Base stations use these (type 1 and 2) messages to keep their local cluster location servers informed about the HRLI of the locally resident mobile units.

3. **CLS-HOME-CLS-ADD-LRLI(MU, Age, Link-Qual):** When a visiting MU enters into a cluster, the local CLS sends this message to the MU's home CLS in order to inform the latter about the low resolution location information (see Section 6.4.1) of the arrived mobile unit. Depending on the required temporal granularity of the location information, any *significant* change of the wireless link-quality may also cause a CLS to send this message.

4. **CLS-HOME-CLS-DELETE-LRLI(MU):** Departure of a visiting mobile unit from a cluster prompts the local CLS to send this message to the home CLS of the relevant MU.

   Message types '3' and '4' are utilised by the cluster location servers to inform the home CLSs about the LRLI of their domestic mobile units.

5. **HOME-CLS-CLS-ADD-FORWARD-LINK(MU, CLS):** If an MU is simultaneously seen by more than one cluster, then each such cluster needs to know about the HRLI of the MU in all of its co-sighted clusters. When the mobile unit MU_j is first seen by CLS_a, it sends a CLS-HOME-CLS-ADD-LRLI(MU_j) message to the home cluster location server CLS_h. If CLS_h realises that the mobile unit is already being seen by some other clusters, then it sends the HOME-CLS-CLS-ADD-FORWARD-LINK(MU_j, CLS_a) messages to all such co-sighted cluster location servers. Upon reception of such a message, a CLS puts the address of CLS_a into the NCLSP list (see Section 6.6.3) in the location database. This list is used by a CLS to inform CLS_a about any change regarding the local HRLI of the mobile unit MU_j. Similar messages are also sent to CLS_a, one for each co-sighted cluster.

6. **HOME-CLS-CLS-DELETE-FORWARD-LINK(MU, CLS):** Used by a home CLS to inform other cluster location servers to delete entries from NCLSP list. This happens soon after the home CLS is informed about the departure of an MU from a remote cluster.

7. **CLS-CLS-ADD-HRLI(MU, BS, Age, Link-Qual):** Whenever a CLS receives a BS-CLS-ADD-HRLI message regarding the mobile unit MU_j, the former sends the CLS-CLS-ADD-HRLI messages to all the co-sighted clusters, pointed by the NCLSP list in the location database. This is to keep the co-sighted clusters informed of any change of local HRLI regarding MU_j.

8. **CLS-CLS-DELETE-HRLI(MU, BS):** This is used in a fashion similar to the previous one except that a CLS sends the message only when it receives a BS-CLS-DELETE-HRLI message from one of the local base stations.

Other than the specified fields, each message carries the identities of the source and the destination entities.

C.1.2 Update Policy

These eight messages are sufficient to disseminate and store the mobile unit location information, in a format as specified in Section 6.6.3. Messages '1' and '2' are used for keeping a CLS updated about the local HRLI of an MU, which is currently being seen by the local cluster. In the extreme cases when an MU enters or leaves a cluster, messages '3' and '4' are used for keeping the corresponding home CLS informed about the remote LRLI of the relevant MU. Any change in the remote low resolution location information of a mobile unit causes the corresponding home CLS to send messages '5' and '6' so that all the clusters, seeing the MU, can maintain an up-to-date NCLSP list of all the other co-sighted clusters. Messages '7' and '8' are used by a cluster location server for sending any change of the local HRLI, about an MU, to the other co-sighted clusters, pointed by the NCLSP list in the location database.

These ensure that a cluster location server always maintains the remote LRLI regarding all its domestic mobile units and the complete HRLI about all the locally residing MUs. If a mobile representative is residing in the local cluster, the corresponding CLS needs to maintain the complete HRLI of the client MU. This needs the following minor augmentation of the described scheme. It is shown in the next section that the management of a representative is principally governed by the home CLS of the client mobile unit. Whenever a new representative MR_j is created in $Cluster_k$, the home CLS of MU_j sends the HOME-CLS-CLS-ADD-FORWARD-LINK(MU_j, CLS_k) messages to all the clusters which are currently seeing the mobile unit MU_j. This keeps CLS_k informed about the complete HRLI of the mobile unit MU_j. When a mobile representative is removed from a cluster, the corresponding home CLS sends the HOME-CLS-CLS-DELETE-FORWARD-LINK messages to all the clusters which are currently seeing the client mobile unit. These are used for purging the relevant NCLSP list entries from the recipient CLS location databases.

It is found that both during the location update and the query search procedures, the information transaction is restricted within the co-sighted neighbour clusters and the home cluster of the relevant MU. This property of the location server is also maintained within the design of the representative management scheme, described in the next section. The spatial locality of the update and the search traffic makes this scheme scalable for the large local area networks.

C.2 Representative Management Messages

This section provides the message details for the representative management and migration procedures, introduced in Section 6.9.

C.2.1 Creation and Deletion

The following messages are used for the mobile representative management and the maintenance of the replica of an MU location information within its representative database.

9. **HOME-CLS-CLS-CREATE-MR(MU):** After an appropriate cluster (e.g. $Cluster_c$) is chosen for lodging a mobile representative, this message is sent by the home CLS of the client MU (CLS_h) to instruct CLS_c to create the MR into it.

10. **CLS-SU-CREATE-MR(MU):** On reception of the above message, CLS_c selects a stationary unit SU_p as the host-entity of the MR and sends this message to SU_p.

11. **SU-CLS-MR-CREATED(MU):** When the stationary unit SU_p receives a CLS-SU-CREATE-MR message, it attempts to create a representative process for the specified mobile unit. On successful creation of the MR an SU-CLS-MR-CREATED message is sent back to the local CLS.

12. **CLS-HOME-CLS-MR-CREATED(MU, MR):** Once CLS_c comes to know about the successful creation of the mobile representative in SU_p, the former sends this message to the home cluster location server CLS_h, confirming the successful spawning of the desired mobile representative process.

    These four messages are used for creating the mobile representatives, both during their initial incarnations and during their migration from one cluster to another.

The next six messages are utilised for killing and gracefully removing the mobile representatives, when needed. Although the decision to kill an MR is usually taken by the home CLS of its client MU, the representative's host-entity and the local CLS are also entitled to kill the MR in the error situations, mentioned in Section C.3.

13. **HOME-CLS-CLS-KILL-MR-REQUEST(MU):** When the home cluster location server CLS_h decides to terminate a mobile representative, residing within $Cluster_c$, CLS_h sends this message to CLS_c.

14. **CLS-SU-KILL-MR-REQUEST(MU):** On reception of the previous message, CLS_c sends this message to the host-entity of the relevant MR.

15. **SU-CLS-KILL-MR-REQUEST-ACCEPT(MU):** The host-entity of the MR acknowledges the above request by sending this message back to the local cluster location server CLS_c.

16. **CLS-HOME-CLS-KILL-MR-REQUEST-ACCEPT(MU):** This message is used as an extension of the previous acknowledge operation and sent by CLS_c to the home cluster location server CLS_h. Note that these two acknowledge messages do not ensure the removal of the representative; they only guarantee that the host-entity of the relevant MR is informed about the intention of the termination of the representative.

17. **SU-CLS-MR-REMOVED(MU):** Once the sentenced MR decides to commit suicide (this is when all the mobile connections passing through it are terminated by the end entities; see Chapter 7 for more details), it sends this message to the local cluster location server CLS_c before the self-termination.

18. **CLS-HOME-CLS-MR-REMOVED(MU):** On reception of the previous message, CLS_c sends a CLS-HOME-CLS-MR-REMOVED message to the home cluster location server CLS_h, informing about the successful termination of the relevant MR. Note that these two messages are not absolutely essential for the representative removal process; they are needed to maintain consistent information regarding the existence of the mobile representatives.

The following five messages are used for the creation and update of the representative databases within the cluster location servers. The replicated MU information, within the representative location databases, is also maintained by using these messages.

19. **HOME-CLS-CLS-ADD-MR-ADDR(MU, MR):** When the cluster location server CLS_h receives a confirmation regarding the successful creation of a representative for a domestic mobile unit MU_j, CLS_h sends the HOME-CLS-CLS-ADD-MR-ADDR messages to all the clusters which are currently being seen by MU_j. These messages are used by the recipient cluster location servers to create and update the representative address fields of their location databases.

20. **HOME-CLS-CLS-DELETE-MR-ADDR(MU):** On reception of a CLS-HOME-CLS-KILL-MR-REQUEST-ACCEPT message, the home cluster location server CLS_h sends the HOME-CLS-CLS-DELETE-MR-ADDR messages to all the clusters which are currently being seen by the mobile unit MU_j. These messages are used for the deletion update of the same database entries, as mentioned above.

21. **HOME-CLS-CLS-DELETE-UMRLC(MU):** Immediately after the completion of a representative removal cycle (after CLS_h receives a CLS-HOME-CLS-KILL-MR-REQUEST-ACCEPT message), CLS_h sends these messages to all the clusters whose address is found in the UMRLCP list of the DMR in the location database. On reception of such a message, to improve cache consistency, a cluster location server flushes the corresponding MR entry from its UMRLC in location database.

22. **CLS-MR-ADD-HRLI(MU, BS, Age, Link-Qual):** Soon after a mobile representative for MU_j is created in the local cluster, the CLS uses these messages to inform MR_j about the complete HRLI of the mobile unit MU_j. MR_j is notified about any further changes (except deletion) in the HRLI of MU_j.

23. **CLS-MR-DELETE-HRLI(MU, BS):** Also used for updating the replicated MU_j location information in the representative databases. Any deletion of MU_j HRLI in CLS location database, causes the local CLS to send this message to the representative MR_j.

C.2.2 Migration

On relocation of a mobile unit MU_j to a new cluster, the home CLS_h may decide to change the host-entity of the representative MR_j to the new cluster. CLS_h uses the heuristics, stated in Section 6.9, in order to choose a new residence cluster for MR_j. To avoid any disruptions of the mobile connections, passing through a representative during the migration process, a duplication scheme is adopted.

Soon after a migration decision is taken by the home cluster location server CLS_h, it uses message '13' to initiate a representative removal procedure which ends with the reception of an acknowledgement message '16' back from the CLS of the cluster, currently hosting MR_j. At this stage, representative MR_j does not terminate itself as long as at least one mobile connection through it remains active. This ensures that during the entire lifetime of a mobile connection, a representative does not change its host-entity; this feature makes handover procedure simple and consistent (see Chapter 7 for more detail). Messages '17' and '18' are used for informing CLS_h about the final termination of this copy of MR_j.

On reception of a CLS-HOME-CLS-KILL-MR-REQUEST-ACCEPT message from the old residence cluster of MR_j, CLS_h informs the relevant clusters about the removal of the representative so that those CLSs can update their MR databases accordingly. Although the old copy of MR_j stays alive for some more time, it remains transparent to the rest of the network and is used only by the already existing mobile connections. At this stage, CLS_h starts the procedure of creating a duplicate copy of the representative in the newly chosen cluster. Messages '9' to '12' are used for creating this new copy of MR_j and the corresponding MR databases are updated accordingly, using the messages '19', '20', '22' and '23'. From this point onwards, the connection manager and all the other applications use the new copy of MR_j.

C.3 Error Handling

All the location service and the representative management related operations heavily rely on the integrity of the connections between the base stations and the local CLSs and between the different cluster location servers. Whenever such a connection breaks, a special set of actions are required for maintaining consistency of both the MU and the MR location information.

C.3.1 BS-CLS Connection Failure

If the connection between the base station BS_i and the local cluster location server CLS_k breaks unexpectedly, the latter acts as if it has received the BS-CLS-DELETE-HRLI messages corresponding to all the HRLI entries (both in DMR and VMR) corresponding BS_i. When the connection is re-established, BS_i requires to send the BS-CLS-ADD-HRLI messages to CLS_k for each of its BNT entries.

C.3.2 CLS-CLS Connection Failure

In case of a failure of the connection between the cluster location servers CLS_p and CLS_q, CLS_p takes the following corrective actions for each of the mobile unit entries in its location database. The described actions correspond to the entry for the mobile unit MU_j. Depending on the relationship between MU_j, CLS_p and CLS_q the following possibilities are considered. In all the situations the described actions are taken by the cluster location server CLS_p.

CLS_p is in the home cluster of MU_j

- Delete the remote location information (both HRLI and LRLI) corresponding to CLS_q from the MU_j entry in DMR.
- Delete CLS_q entry from the NCLSP list of DMR.
- Send the HOME-CLS-CLS-DELETE-FORWARD-LINK(MU_j, CLS_q) messages to all the CLSs, whose addresses are found in the NCLSP list of DMR.
- If MR_j was residing in $Cluster_q$ then CLS_p assumes that CLS_q would kill and remove MR_j as a result of this connection termination. The home cluster location server CLS_p sends the HOME-CLS-CLS-DELETE-MR-ADDR messages to all the CLSs, pointed by the NCLSP list of DMR. Once these are done, CLS_p chooses a new cluster to lodge the representative MR_j and the necessary MR creation cycle is invoked.

From CLS_p's part no action is necessary after the connection is re-established.

CLS_q is in the home cluster of MU_j

- Delete the remote, the NCLSP and the MR_j address fields of the MU_j entry in VMR. Retaining the local high resolution location information speeds up the recovery process after the connection is re-established.
- If MR_j is residing into the local cluster then send a CLS-SU-KILL-MR-REQUEST message to the relevant host-entity to initiate the MR_j removal process.

Soon after the connection is re-established, CLS_p should send the CLS-HOME-CLS-ADD-LRLI messages to the home cluster location server CLS_q, for all the HRLI entries corresponding to MU_j.

Neither of CLS_p and CLS_q is in the home cluster of MU_j

- Delete the remote location information (both HRLI and LRLI) corresponding to CLS_q from the MU_j entry in VMR. The CLS_q entry in NCLSP list, however, is to be retained so that it can be used after the connection is re-established.

- If MR_j is residing locally, CLS_p should inform the representative about the change of location information of MU_j, as specified above.

- If MR_j is residing in CLS_q then CLS_p should not change the MR address entry in its VMR. This transient inconsistency of MR_j address field in CLS_p may mislead the applications temporarily, but this approach is useful once the broken connection is re-established.

  If both CLS_p and CLS_q have connectivity with the home CLS of MU_j then the broken connection is bound to be re-established. On the other hand, if either of CLS_p and CLS_q looses connection with the home CLS then the MR address field in the VMR of CLS_p disappears anyway. This proves the transient property of the mentioned inconsistency of MR address fields in the VMR.

Once the connection is re-established, CLS_p sends the CLS-CLS-ADD-HRLI messages to CLS_q (note that CLS_q is chosen from the NCLSP list in VMR) to inform the latter regarding the local HRLI of MU_j.

C.3.3 CLS-MR Connection Failure

When the connection between the mobile representative MR_j and the local cluster location server CLS_k breaks, CLS_k assumes that the host-entity of MR_j would kill the representative as a result of this connection termination. CLS_k also sends a CLS-HOME-CLS-KILL-MR-REQUEST-ACCEPT message to the home CLS of MU_j (if CLS_k itself is not the home CLS) to inform the latter about the termination of MR_j. This allows the home cluster location server to take necessary actions regarding the creation of a new MR_j and so on. On the other side of the terminated connection, the host-entity removes the representative MR_j.

An important point to be noted is that for any connection termination event, it is not advisable to delete the local HRLI entries. Keeping these information for the future references would expedite the recovery procedure.

C.4 Mobility Profile Augmentation

An augmented location update strategy which attempts to incorporate the *mobility profile* based partitioning schemes (see Section 6.10), is presented in the section. Since most of the location queries, about a mobile unit MU_j and its representative MR_j, originate at MU_j's query partition, it is useful to keep complete (both HRLI and LRLI) location information of MU_j into the CLSs of the clusters belonging to QP_j. If CLS_k is a member of the query partion QP_j then all the clusters, seen by MU_j, require to keep CLS_k's address in the NCLSP list of their location databases. This ensures that CLS_k would be informed of the complete location information regarding both MU_j and MR_j. When the home cluster location server CLS_h decides[1] that CLS_q belongs to one of the clusters within QP_j, it takes the following actions.

- Send a HOME-CLS-CLS-ADD-MR-ADDR message to CLS_q so that the latter can create an MR address entry in its location database.
- Also send HOME-CLS-CLS-ADD-FORWARD-LINK(MU_j, CLS_q) messages to all the clusters which are currently seen by the mobile unit MU_j.

These are sufficient for the cluster location server CLS_q to be up-to-date about the location information regarding MU_j and MR_j. When CLS_h decides that CLS_q should not belong to QP_j any more, the former needs to inform CLS_q to delete all the entries regarding MU_j and MR_j location information. A new message type **HOME-CLS-CLS-DELETE-QUERY-PARTITION-INFO(MU)** is required for this purpose. When CLS_q is removed from QP_j, CLS_h does the following.

- Send a HOME-CLS-CLS-DELETE-MR-ADDR message to CLS_q so that the latter can purge the MR_j entry from its location database.
- Send the HOME-CLS-CLS-DELETE-FORWARD-LINK(MU_j, CLS_q) messages to all the clusters which are currently seen by the mobile unit MU_j. This is just to let those clusters know that it is no more a requirement to forward the MU_j location information to CLS_q.
- Send a HOME-CLS-CLS-DELETE-QUERY-PARTITION-INFO(MU_j) message to CLS_q. On reception of this message, CLS_q deletes all the location information regarding MU_j.

[1] Any such decision is taken based on the mobility statistics of MU_j.

Bibliography

[Abu-Amara 93] H. Abu-Amara, "A Fast Topology Maintenance Algorithm For High-Bandwidth Networks," *IEEE/ACM Transactions on Networking,* vol. 1, no. 3, June 1993, pp. 386-394.

[Armbrüster 92] H. Armbrüster, B. G. Marchent and M. Reiβ, "Advanced Terminal and Personal Mobility - A Challenge for ATM Concepts and Switching?," *Proceedings of the XIV International Switching Symposium,* Yokohama, Japan, October 1992, article B1.1.

[Arunkumar 92] S. Arunkumar and R. S. Panwar, "Efficient Broadcast Using Selective Flooding," *Proceedings of the IEEE INFOCOM,* Florence, Italy, May 1992, pp. 8D.3.1-8D.3.8.

[ATMUNI 93] ATM Forum Technical Committee, "ATM User-Network Interface Specification Version 2.1," *Signalling Subworking Group Draft,* May 1993.

[Awerbuch 90] B. Awerbuch, I. Cidon, I. Gopal, M. Kaplan and S. Kutten, "Distributed Control for PARIS," *Proceedings of the Ninth Annual ACM Symposium on Principles of Distributed Computing,* 1990, pp. 145-149.

[Awerbuch 91] B. Awerbuch and D. Peleg, "Concurrent Online Tracking of the Mobile Users," *Proceedings of the ACM SIGCOMM Symposium on Communication, Architectures and Protocols,* October 1991, pp. 221-233.

[Aziz 94] A. Aziz, "A Scalable and Efficient Intra-Domain Tunneling Mobile-IP Scheme," *Computer Communications Review - ACM SIGCOMM,* vol. 24, no. 1, January 1994, pp. 12-21.

[Badrinath 92] B. R. Badrinath, T. Imielinski and A. Virmani, "Locating Strategies for Personal Communication Networks," *Proceedings of the Workshop on Networking of Personal Communication Applications,* Florida, December 1992, pp. II.1-II.9.

[Bae 91] J. Bae and T. Suda, "Survey of Traffic Control Schemes and Protocols in ATM Networks," *Proceedings of the IEEE,* February 1991, pp. 170-189.

[Berenbaum 93] A. Berenbaum, J. Dixon, A. Iyengar and S. Keshav, "A Flexible ATM-Host Interface for XUNET II," *IEEE Network Magazine,* July 1993, pp. 18-23.

[Bhagwat 93] P. Bhagwat and C. R. Perkins, "A Mobile Networking System Based on Internet Protocol (IP)," *Proceedings of the USENIX Symposium on Mobile and Location-Independent Computing,* Cambridge, Massachusetts, August 1993, pp. 69-82.

[Biswas 92] S. K. Biswas, J. D. Porter and A. Hopper, "Performance of a Multiple Access Protocol for an ATM Based Pico-Cellular Radio LAN," *Proceedings of the Third IEEE International Symposium on Personal, Indoor and Mobile Radio Communications,* Boston, Massachusetts, October 1992, pp. 5.2.1-5.2.6.

[Biswas 93] S. K. Biswas, "Performance Modelling of Window Based Flow Control in a Multihop Packet Radio LAN," *Proceedings of the IEEE INFOCOM,* vol. 3, San Francisco, March-April 1993, pp. 1222-1231.

[Biswas 94a] S. K. Biswas and A. Hopper, "A Connection Management Scheme for a Mobile Radio LAN," *Proceedings of the IEEE International Conference on Personal Wireless Communications,* Bangalore, India, August 18-19, 1994.

[Biswas 94b] S. K. Biswas, "An Agent Based Location Manager Design for a Mobile Radio LAN," *Under Preparation.*

[Black 93] R. Black, "FDL Cell Formats and Meta Signalling," *University of Cambridge Systems Research Group, ATM Document Collection,* February 1993, pp. 5.1-5.4.

[Black 94] R. Black and S. Crosby, "Experience and Results from the Implementation of an ATM Socket Family," *Proceedings of the USENIX Winter Technical Conference,* San Francisco, January 1994.

[Brody 92] G. Bordy, J. Parker and J. Wasserman, "Subscriber Tracking and Locating in Personal Communications Networks," *Proceedings of the XIV International Switching Symposium,* Yokohama, Japan, October 1992, article B3.2.

[Buchholz 91] D. Buchholz, P. Odlyzko, M. Taylor and R. White, "Wireless In-Building Network Architecture and Protocols," *IEEE Network Magazine,* November 1991, pp. 31-38.

[CCITT 90a] CCITT, "Broadband Aspects of ISDN," *CCITT Draft Recommendation I.121,* May 1990.

[CCITT 90b] CCITT, "Video Codec for Audiovisual Services," *CCITT Draft Recommendation H.261,* December 1990.

[Chia 91] S. T. S. Chia, "A Handover Protocol for a Mixed Cell System," *Proceedings of the Sixth International Conference on Mobile Radio and Personal Communication,* United Kingdom, December 1991, pp. 225-232.

[Cidon 85] I. Cidon and M. Sidi, "The Effect of Capture on Collision-Resolution Algorithms," *IEEE Transactions on Communications,* vol. com-33, April 1985, pp. 317-324.

[Cohen 92] D. Cohen, J. B. Postel and R. Rom, "IP Addressing and Routing in a Local Wireless Network," *Proceedings of the IEEE INFOCOM,* vol. 2, Florence, Italy, May 1992, pp. 626-632.

[Comer 91] D. E. Comer and D. L. Stevens, "Internetworking with TCP/IP, volume I," *Prentice-Hall, Inc. Englewood Cliffs,* New Jersey, 1991.

[Cooper 83] G. R. Cooper and R. W. Nettleton, "Cellular Mobile Technology: The Great Multiplier," *IEEE Spectrum,* vol. 20, no. 6, June 1993, pp. 30-37.

[Crosby 91] S. A. Crosby, A. E. Krzesinski and J-Y Le Boudec, "An MSCCC Model of Multilayer Window Flow Control," *Proceedings of the Fifth International Conference on Modelling Techniques and Tools for Computer Performance Evaluation,* Italy, February 1991, pp. 71-85.

[Crosby 93] S. Crosby, "In-Call Renegotiation of Traffic Parameters," *Proceedings of the IEEE INFOCOM,* vol. 2, San Francisco, March-April 1993, pp. 638-646.

[DeSimone 92] A. DeSimone, S. Nanda and O-C. Yue, "Cellular Networking Issues in Wireless LANs," *Proceedings of the Workshop on Networking of Personal Communication Applications,* Florida, December 1992, pp. III.1-III.19.

[Dixon 91] M. J. Dixon, "System Support for Multi-Service Traffic," *Ph.D. Thesis, Cambridge University Computer Laboratory,* September 1991.

[Doar 93] J. M. S. Doar, "Multicast in the Asynchronous Transfer Mode Environment," *Ph.D. Thesis, Cambridge University Computer Laboratory,* April 1993.

[Earnshaw 92] R. Earnshaw, "Simulation of Packet and Cell-based Communication Networks," *Ph.D. Thesis, University of Durham,* April 1992.

[Earnshaw 93] R. Earnshaw, "The Topology Discovery Service," *University of Cambridge Systems Research Group, ATM Document Collection,* February 1993, pp. 11.1-11.8.

[Fairbairns 93] R. Fairbairns, "Esprit BRA 6865 - Pegasus: Summary Report, Kernel Work Package," *Pegasus Project Report,* September 1993.

[Forum 93] ATM Forum, *ATM User-Network Interface Specification,* Version 2.1, 1993.

[Goodman 91a] D. J. Goodman and S. X. Wei, "Efficiency of Packet Reservation Multiple Access," *IEEE Transactions on Vehicular Technology,* vol. 40, no. 1, February 1991, pp. 170-176.

[Goodman 91b] D. J. Goodman, "Trends in Cellular and Cordless Communications," *IEEE Communication Magazine,* June 1991, pp. 31-40.

[Greaves 90] D. J. Greaves, D. Lioupis and A. Hopper, "The Cambridge Backbone Ring," *Proceedings of the IEEE INFOCOM,* San Francisco, March-April 1990, vol. 1, pp. 8-14.

[Harter 93] A. Harter and F. Bennett, "Low Bandwidth Infra-Red Networks and Protocols for Mobile Communication Devices," *Olivetti Research Ltd. Technical Report,* no. 93-5, 1993.

[Harter 94] A. Harter and A. Hopper, "A Distributed Location System for the Active Office," *IEEE Network Magazine,* January/February 1994, pp. 62-70.

[Hayter 93] M. D. Hayter, "A Workstation Architecture to Support Multimedia," *Ph.D. Thesis, Cambridge University Computer Laboratory,* November 1993.

[Hellstern 93] K. S. Meier-Hellstern, G. P. Pollini and D. J. Goodman, "Network Protocols for the Cellular Packet Switch," *To be appeared in the IEEE Transactions on Communications.*

[Hopper 88] A. Hopper and R. Needham, "The Cambridge Fast Ring Networking System," *IEEE Transactions on Computers,* vol. 37, October 1988, pp. 1214-1223.

[Hopper 90] A. Hopper, "Pandora - An Experimental System for Multimedia Applications," *ACM Operating System Review,* April 1990.

[Hyden 94] E. A. Hyden, "Operating System Support for Quality of Service," *Ph.D. Thesis, Cambridge University Computer Laboratory,* February 1994.

[Imielinski 92a] T. Imielinski and B. R. Badrinath, "Querying in Highly Mobile Distributed Environment," *Proceedings of the 18th VLDB Conference,* Vancouver, British Columbia, Canada, 1992.

[Imielinski 92b] T. Imielinski and B. R. Badrinath, "Mobile Wireless Computing: Solutions and Challenges in Data Management," *Technical report,* DataMan project, Rutgers University, April 1990.

[Ioannidis 91] J. Ioannidis, D. Duchamp and G. Maquire, "IP-based Protocols for Mobile Internetworking," *Proceedings of the SIGCOMM,* ACM, September 1991, pp. 235-245.

[Ioannidis 92] J. Ioannidis, D. Duchamp and G. Maquire, "Protocols for Supporting Mobile IP Hosts," *Mobile IP Working Group Internet Draft,* June 1992.

[IS54 89] Electronic Industries Association, "Dual-Mode Subscriber Equipment - Network Equipment Compatibility Specifications," *Interim Standard 54,* December 1989.

[ISO/IEC 91] ISO/IEC, "Coded Representation of Picture, Audio and Multimedia/Hypermedia Information," *Committee Draft ISO/IEC CD 11172,* December 1991.

[Jardetzky 92] P. W. Jardetzky, "Network File Server Design for Continuous Media," *Ph.D. Thesis, Cambridge University Computer Laboratory,* October 1992.

[Jones 93] A. Jones and A. Hopper, "Handling Audio and Video Streams in a Distributed Environment," *Proceedings of the Fourteenth ACM Symposium on Operating System Principles,* Asheville, North Carolina, December 1993, pp. 231-243.

[Jung 93] J. Jung and D. Seret, "Translation of QoS Parameters into ATM Performance Parameters in B-ISDN," *Proceedings of the IEEE INFOCOM,* vol. 2, San Francisco, March-April 1993, pp. 748-755.

[Keeton 93] K. Keeton, B. A. Mah, S. Seshan, R. H. Katz and D. Ferrari, "Providing Connection-Oriented Network Services to Mobile Hosts," *Proceedings of the USENIX Symposium on Mobile and Location-Independent Computing,* Cambridge, Massachusetts, August 1993, pp. 83-102.

[Khan 91] J. Y. Khan, Y. Kriaras and J. Dunlop, "A Mixed Voice and Data Protocol for Digital Cellular Radio," *Proceedings of the Sixth International Conference on Mobile Radio and Personal Communication,* United Kingdom, December 1991, pp. 108-114.

[Kleinrock 75] L. Kleinrock, "Queueing Systems, Theory, vol. 1," *John Wiley and Sons,* New York, 1975.

[Kurose 93] J. Kurose, "Open Issues and Challenges in Providing Quality of Service Guarantees in High-Speed Networks," *Computer Communications Review - ACM SIGCOMM,* vol. 23, no. 1, January 1993, pp. 6-14.

[Lam 80] S. S. Lam, "Packet Broadcast Networks - A Performance Analysis of the R-ALOHA Protocol," *IEEE Transactions on Computers,* vol. C-29, no. 7, July 1980, pp. 596-603.

[Lawton 91] M. C. Lawton, R. L. Davies and J. P. McGeehan, "An Analytical Model for Indoor Multipath Propagation in the Picocellular Environment," *Proceedings of the Sixth International Conference on Mobile Radio and Personal Communication,* United Kingdom, December 1991, pp. 1-8.

[Leopold 93] H. Leopold, A. Campbell and N. Singer, "Will B-ISDN Services Meet the Needs of Distributed Multimedia Communications," *Proceedings of the Fourth IEE Conference on Telecommunications,* Manchester, UK, 18-21 April 1993, pp. 139-145.

[Leslie 83] I. M. Leslie, "Extending the Local Area Network," *Ph.D. Thesis, Cambridge University Computer Laboratory,* February 1983.

[Leslie 91] I. Leslie and D. McAuley, "Fairisle: An ATM Network for the Local Area," *Proceedings of the SIGCOMM 1991,* pp. 327-336.

[Leslie 92] I. Leslie, D. McAuley and S. J. Mullender, "Pegasus - Operating System Support for Distributed Multimedia Systems," *Cambridge University Computer Laboratory Technical Report,* no. 282, 1992.

[Lo 94] S. L. Lo, "A Modular and Extensible Network Storage Architecture," *Ph.D. Thesis, Cambridge University Computer Laboratory,* January 1994.

[Mallinder 88] B. J. T. Mallinder, "An Overview of the GSM System," *Proceedings of the Digital Cellular Radio Conference,* Hagen, Germany, October 1988, pp. 1a/1-1a/13.

[McAuley 90] D. R. McAuley, "Protocol Design for High Speed Networks," *Ph.D. Thesis, Cambridge University Computer Laboratory,* January 1990.

[McQuillan 80] J. M. McQuillan, I. Richer and E. C. Rosen, "The New Routing Algorithm for the ARPANET," *IEEE Transactions on Communications,* vol. com-28, May 1980, pp. 711-719.

[Metcalfe 76] R. M. Metcalfe and D. R. Boggs, "Ethernet: Distributed Packet Switching for Local Computer Networks," *Communications of the ACM,* vol. 19, no. 7, July 1976, pp. 395-404.

[Milway 94] D. R. Milway, "Personal Communication," April 1994.

[Minzer 89] S. E. Minzer, "Broadband ISDN and Asynchronous Transfer Mode (ATM)," *IEEE Communication Magazine,* September 1989, pp. 17-24.

[Mitrani 87] I. Mitrani, "Modelling of Computer and Communication Systems," *Cambridge University Press,* Cambridge, 1987.

[Murata 88] M. Murata and H. Takagi, "Two-Layer Modelling for Local Area Networks," *IEEE Transactions on Communications,* vol. com-36, September 1988, pp. 1022-1034.

[Namislo 84] C. Namislo, "An Analysis of Mobile Radio Slotted ALOHA Networks," *IEEE Journal on Selected Areas in Communications,* July 1984, pp. 583-588.

[Nanda 91] S. Nanda, D. J. Goodman and U. Timor, "Performance of PRMA: A Packet Voice Protocol for Cellular Systems," *IEEE Transactions on Vehicular Technology,* vol. 40, no. 3, August 1991, pp. 584-598.

[NCLATM 92] "Network Compatible ATM for Local Network Applications," *Discussion Document Circulated by Apple Computer, Bellcore, Sun Microsystem and Xerox,* April 1992.

[Ott 94] M. Ott, V. Bansal, R. J. Siracusa, J. P. Hearn, R. Dighe, D. Raychaudhuri, I. Iwata and H. Suzuki, "A Prototype ATM Network Based System for Multimedia-on-Demand," *Proceedings of the IEEE COMSOC Workshop Multimedia'94,* Kyoto, Japan, May 17-19, 1994.

[Panwar 88] S. Panwar, D. Towsley and J. Wolf, "Optimal Scheduling Policies for a Class of Queues with Customer Deadlines to the Beginning of Service," *Journal of ACM,* October 1988, pp. 832-844.

[Perkins 92a] C. Perkins and Y. Rekhter, "Short-cut Routing for Mobile Hosts," *Internet Draft,* July 1992.

[Perkins 92b] C. Perkins and Y. Rekhter, "Support for Mobility with Connectionless Network Layer Protocols," *Internet Draft,* November 1992.

[Perkins 93] C. Perkins, "Providing Continuous Network Access to Mobile Hosts Using TCP/IP," *Proceedings of the Joint European Networking Conference,* May 1993.

[Perlman 85] R. Perlman, "An Algorithm for Distributed Computation of a Spanning Tree in an Extended LAN," *Proceedings of the Ninth ACM Data Communication Symposium,* 1985, pp. 44-53.

[Pollini 92] G. P. Pollini, K. S. Meier-Hellstern and D. J. Goodman, "Handover Protocols between Metropolitan Area Networks," *Proceedings of the IEEE Global Telecommunication Conference,* Orlando, Florida, December 1992.

[Porter 94] J. Porter and A. Hopper, "An ATM Based Protocol for Wireless LANs," *Olivetti Research Ltd. Technical Report,* no. 94-2, April 1994.

[Postel 81] J. Postel, "Internet Protocol," *RFC 792,* DDN Network Information Centre, September 1981.

[Pratt 94] I. Pratt and P. Barham, "The ATM Camera V2 (AVA2000)" *University of Cambridge System Research Group, ATM Document Collection,* March 1994, pp. 36.1-36.29.

[Rappaport 89] T. S. Rappaport, "Indoor Radio Communications for Factories of the Future," *IEEE Communication Magazine,* May 1989, pp. 15-24.

[Rappaport 91] S. S. Rappaport, "Modeling the Hand-off Problem in Personal Communication Networks," *Proceedings of the 41st IEEE Vehicular Technology Conference,* 1991, pp. 517-523.

[Raychaudhuri 93] D. Raychaudhuri and N. Wilson, "Multi-media Transport in Next-Generation Personal Communication Networks," *Proceedings of the International Conference on Communication,* May 1993, pp. 858-862.

[Raychaudhuri 94] D. Raychaudhuri and N. Wilson, "ATM Based Transport Architecture for Multiservices Wireless Personal Communication Networks," *IEEE Journal of Selected Areas of Communication,* Submitted for publication.

[Reiser 79] M. Reiser, "A Queueing Network Analysis of Computer Communication Networks with Window Flow Control," *IEEE Transactions on Communications,* vol. com-27, August 1979, pp. 1199-1209.

[Reiser 80] M. Reiser and S. S. Lavenberg, "Mean-Value Analysis of Closed Multichain Queueing Networks," *Journal of ACM,* vol. 27, no. 2, April 1980, pp. 313-322.

[Richardson 94] T. Richardson, G. Mapp, F. Bennett and A. Hopper, "Teleporting in an X Window System Environment," *IEEE Personal Communication Magazine,* Submitted.

[Roe 93] M. Roe, "IP over MSNL," *University of Cambridge System Research Group, ATM Document Collection,* February 1993, pp. 8.1-8.7.

[Schroeder 91] M. D. Schroeder, A. D. Birrel, M. Burrows, H. Murry, R. M. Needham, T. L. Rodeheffer, E. H. Satterthwaite and C. P. Thacker, "Autonet: A High-Speed, Self-Configuring Local Area Network Using Point-to-Point Links," *IEEE Journal on Selected Areas in Communications,* vol. 9, no. 8, October 1991, pp. 1319-1335.

[Simmons 92] P. Simmons and M. Mouly, "Switching Handovers in Microcellular Mobile Networks: An Architectural Evolution," *Proceedings of the XIV International Switching Symposium,* Yokohama, Japan, October 1992, article B2.1.

[Sreenan 93] C. J. Sreenan, "Synchronisation Services for Digital Continuous Media," *Ph.D. Thesis, Cambridge University Computer Laboratory,* March 1993.

[Steele 85] R. Steele and V. K. Prabhu, "Mobile Radio Cell Structure for High User Density and Large Data Rates," *Proceedings of the IEE, Part F,* no. 5, August 1985, pp. 396-404.

[Steele 89] R. Steele, "The Cellular Environment of Lightweight Handheld Portables," *IEEE Communication Magazine,* vol. 27, no. 7, June 1989, pp. 20-29.

[Steele 92] R. Steele, "Mobile Radio Communications," *Pentech Press Ltd.,* London, 1992.

[Swain 90] R. S. Swain and D. W. J. Holmes, "The Digital Cordless Communications Common Air Interface," *British Telecom Technical Journal,* vol. 8, no. 1, January 1990, pp. 12-18.

[Tajibnapis 77] W. D. Tajibnapis, "A Correctness Proof of a Topology Information Maintenance Protocol for a Distributed Computer Network," *Communications of the ACM,* vol. 20, no. 7, July 1977, pp. 477-485.

[Targett 88] D. J. Targett and H. R. Rast, "Handover-enhanced Capabilities of the GSM System," *Proceedings of the Digital Cellular Radio Conference,* FRG, October 12-14, 1988, pp. 3C/1-3C/11.

[Tasaka 83] S. Tasaka, "Stability and Performance of the R-ALOHA Protocol," *IEEE Transactions on Computers,* vol. C-32, no. 8, August 1983, pp. 717-726.

[Taxi 92] "Data Sheets for Am7968/Am7969-175 TAXIchip Integrated Circuits," *Technical Manual, AMD Ltd.,* 1992.

[Tekinay 92] S. Tekinay and B. Jabbari, "A Measurement Based Prioritization Scheme for Handovers in Mobile Cellular Networks," *IEEE Journal on Selected Areas in Communications,* vol. 10, no. 8, October 1992, pp. 1343-1350.

[Teraoka 93] F. Teraoka, Y. Yokote and M. Tokoro, "A Network Architecture Providing Host Migration Transparency," *Proceedings of the SIGCOMM 1993,* pp. 209-220.

[Tobagi 75] F. A. Tobagi and L. Kleinrock, "Packet switching in Radio Channels: Part II - The Hidden Terminal Problem in Carrier Sense Multiple-Access and the Busy-Tone Solution," *IEEE Transactions on Communications,* vol. com-23, December 1975, pp. 1417-1433.

[Viterbi 91] A. J. Viterbi, "Wireless Digital Communication: A View Based on Three Lessons Learned," *IEEE Communication Magazine,* vol. 29, no. 9, September 1991, pp. 33-36.

[Want 92] R. Want and A. Hopper, "Active Badges and Personal Interactive Computing Objects," *Olivetti Research Ltd. Technical Report,* no. 92-2, February 1992.

[Wilson 93] N. Wilson, R. Ganesh, K. Joseph and D. Raychaudhuri, "Packet CDMA Versus Dynamic TDMA for Multiple Access in an Integrated Voice/Data PCN," *IEEE Journal of Selected Areas of Communication,* vol. 11, no. 6, August 1993, pp. 870-884.

[Wray 94] S. Wray, T. Glauert and A. Hopper, "The Medusa Applications Environment," *Proceedings of the IEEE International Conference of Multimedia Computing and Systems,* Boston, May 14-19 1994, pp. 265-274.

[Young 79] W. R. Young, "Advanced Mobile Phone Service: Introduction Background and Objectives," *Bell System Technical Journal,* vol. 58, no. 1, part 3, January 1979, pp. 1-14.