WIRELESS ATM NETWORK MANAGEMENT MODELING

By

WENBO SHENG

A Thesis
Submitted to the Faculty of Graduate Studies
in Partial Fulfillment of the Requirements
for the Degree of

MASTER OF SCIENCE

Department of
Electrical and Computer Engineering
University of Manitoba
Winnipeg, Manitoba

0-612-32246-7

THE UNIVERSITY OF MANITOBA

FACULTY OF GRADUATE STUDIES

*****

COPYRIGHT PERMISSION PAGE

WIRELESS ATM NETWORK MANAGEMENT MODELING

BY

WENBO SHENG

A Thesis/Practicum submitted to the Faculty of Graduate Studies of The University

of Manitoba in partial fulfillment of the requirements of the degree

of

MASTER OF SCIENCE

I hereby declare that I am the sole author of this thesis.

I authorize the University of Manitoba to lend this thesis to other institutions or individuals for the purpose of scholarly research.

I further authorize the University of Manitoba to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

The University of Manitoba requires the signatures of all persons using or photocopying this thesis. Please sign below, and give address and date.

ACKNOWLEDGMENTS

I would like to take this opportunity to thank all the people who have contributed towards this thesis. Actually, this thesis is a result of collaboration and interaction with many people.

I would first like to thank my advisor, Dr. David C. Blight, for his guidance and encouragement through out my academic years and in particular his contributions and help with this thesis. Also, I would like to express my appreciation for his kindness and generosity. He deserves credit for this thesis.

I would also like to thank Dr. Robert D. McLeod and Dr. Randal J. Peters for taking the time to be on my thesis committee to read my thesis, and for all their effort during this thesis. My thanks also go to Dr. Robert D. McLeod for recommending me as a graduate student.

In addition, I would like to express my thanks to following friends, graduate students and staff at TRLabs: Jose Rueda, Dan Erickson, Peter Czezowski, Haiqing Ma, George Ortega, and Len Dacombe, for their help and discussions with my research. Thanks also go to my current manager, Yair Bourlas and Vice President, James Schellenberg, of Nortel Broadband Wireless Access Unit for their kindness and understanding.

Furthermore, special thanks to my wife, Yiqing Ma, and my baby boy, Jeff Ma Sheng, for their constant encouragement and support. I also give my sincere appreciation to my parents-in-law, Zengrong Ma and Yinfo Kang, for their help and care. Thanks also go to my parents, Guangji Sheng and Minzhi Li, and sister's family, for their remote support from Shanghai, China. Additionally, thanks to my sister-in-law, Liping Cheng, for her encouragement and help. Without their understanding, encouragement and support, it were have been impossible to finish this thesis.

Finally I would like to acknowledge the Telecommunications Research Laboratories (TR*Labs*) for their financial support and for providing facilities for undertaking my research. Finally I would also like to acknowledge the support of Nortel Broadband Wireless Access Unit (BNI).

ABSTRACT

Wireless personal communication services (PCS) and broadband networking for delivery of multimedia information represent two well-established trends in telecommunications. Given that ATM is now viewed as a universal base technology for broadband networks, it is reasonable to consider extension of standard ATM services into next-generation microcellular wireless and PCS scenarios.

In this M. Sc. thesis, I focus on the mobility management and quality of service (QoS) management in such a wireless ATM system. Several algorithms are investigated for the wireless ATM model, which were developed within OPNET. One is an error-free handoff algorithm that allows data to follow the mobile user. When a mobile user moves from one micro-cell to another, the error-free handoff algorithm is implemented to guarantee continuous communication without loss of packets. A replication algorithm was implemented when the user transmits a handoff request to make the data available at the next location of the mobile user. Dynamic-power-control and automatic-frequency-hopping algorithms were investigated to guarantee the quality of service. When at least one channel is idle, the automatic-frequency-hopping algorithm is implemented to avoid interference. If there are not enough channels to hop to, a dynamic-power-control algorithm is used to ensure the quality of service. The decision control algorithm is neural network based. Simulation studies indicate that the intelligence of a neural network provides for more efficient management than alternative techniques. Finally, a variance fractal dimension method was investigated to estimate the traffic and provide control descriptors, which can be used to make network management more efficient.

ABBREVIATIONS

AAL	ATM Adaptation Layer
ABR	Available Bit Rate
ACID	Atomicity, Consistency, Isolation and Durability
AFH	Automatic Frequency Hop
ASAP	As Soon As Possible
ATM	Asynchronous Transfer Mode
BER	Bit Error Rate
B-ISDN	Broadband ISDN
Bm	Brownian motion
BS	Base Station
BSC	Base Station Controller
CBR	Constant Bit Rate
CDMA	Code Division Multiple Access
CDV	Cell Delay Variation
CLP	Cell Loss Priority
CLR	Cell Loss Rate
CTD	Cell Transfer Delay
DLC	Data Link Control
DPC	Dynamic Power Control
FDMA	Frequency Division Multiple Access
FSM	Finite State Machine
GSM	Global System for Mobile communications
HLR	Home Location Register
IDC	Index of Dispersion for Counts
IP	Internet Protocol
ISDN	Integrated Service Digital Network
LAN	Local Area Network
LVQ	Learning Vector Quantization

MAC	Medium Access Control
MSC	Mobility Service Control
MU	Mobile User
OPNET	OPtimized Network Engineering Tool
PCS	Personal Communication Services
PHS	Personal Handy-phone System
PHY	PHysical Layer
PVC	Permanent Virtual Connection
QoS	Quality of Service
RX	Receive
SM	Self-organizing Map
SMDS	Switched MultiMegabit Data Service
SONET	Synchronous Optical NETwork
STD	State Transition Diagrams
STM	Synchronous Transfer Mode
SVC	Switched Virtual Connection
TCP	Transmission Control Protocol
TDMA	Time Division Multiple Access
TDD	Time Division Duplex
TX	Transmit
UBR	Unspecified Bit Rate
UHF	Ultra High Frequency
UMTS	Universal Mobile Telecommunication System
VBR	Variable Bit Rate
VC	Virtual Connection
VLR	Visitor Location Register
WATM	Wireless ATM

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1 INTRODUCTION

Wireless personal communication services (PCS) and broadband networking for delivery of multimedia information represent two important areas in telecommunications. While technologies for PCS and broadband communications developed independently, harmonization into a single architecture framework is motivated by an emerging need to extend multimedia services to portable terminals, as well as by service integration and operational efficiency considerations. Given that ATM is now viewed as a universal base technology for broadband networks, it is reasonable to consider extension of standard ATM services into next-generation microcellular wireless and PCS scenarios [YUA96]. In this thesis, I focus on mobility management and quality of service (QoS) management in such a wireless ATM system.

The mobility of nodes in mobile computing networks is an ongoing problem. These nodes move in and out of the communication range of the fixed distributed network, requiring continuous updating of network topology charts by all the nodes. The problem is in switching from one transmitter to another as the mobile node moves out of range of the former and into the range of the latter. An error-free handoff algorithm is investigated here for improved mobility management by allowing data to follow the mobile user.

In considering the basic scenario of wireless networks connected to ATM networks, we note that one of the primary difficulties is meeting ATM end-to-end quality of service (QoS) requirements. In this wireless ATM model, a QoS guarantee algorithm is developed for the QoS management. There are two parts to this algorithm: a dynamic-power-control algorithm and an automatic-frequency-hop algorithm. Both are demonstrated to work efficiently in the wireless ATM model.

In order to make the network management more intelligent, a neural network is investigated in the network management model. Neural networks can solve certain problems better than conventional digital computers and provide fast and feasible solutions as they can be trained with a relatively smaller number of data samples. The potential benefits expected from using neural networks for traffic control are: adaptive learning; potentially high computation rates due to the massive parallelism of the hardware implementation; generalization from learning and a high degree of robustness or fault tolerance due to the nature of distributed processing. In this study, a neural network is developed for QoS management within the dynamic power control algorithm.

Fractal variance dimension is used to estimate the traffic in the model. Although the spectral dimension may reveal the multifractal nature of the underlying process by estimating β for different short-term (windowed) Fourier analysis, the choice of the window size is difficult, especially since small numerical errors may greatly affect the solution and introduce artifacts. The variance fractal dimension avoids these problems as it does not require a window in the Fourier sense, and therefore does not introduce the window artifact.

Although the proposed algorithms could be tested and assessed in a real network directly, this is very expensive and impractical. Instead, simulation allows for the testing and assessment of many models before actual implementation. In other words, a model developed in the simulation is an abstraction of something for the purpose of understanding it before building it. A simulation tool can be used to demonstrate the feasibility of an approach or algorithm and validate the design of the proposed protocol. Because a model omits nonessential details, it is easier to manipulate than the real entity. Models reduce complexity by separating out a small number of important things to deal with at a time. For the simulation and modeling studies conducted here, OPtimized Network Engineering Tool (OPNET) was used. OPNET is a comprehensive software environment for modeling, simulating, and analyzing the performance of communications networks, computer systems and applications, and distributed systems. Both behavior and performance of a modeled system can be analyzed by performing discrete event simulation. The OPNET/Radio module provides an opportunity to examine the higher level and more complex behavior of Mobile Computing Networks.

CHAPTER 2 LITERATURE SURVEY

2.1 Wireless ATM Network

This section provides a brief overview of ATM networks and wireless communications outlining the basic technologies and difficulties associated with mobility. Wireless ATM networks are then overviewed combining aspects of ATM and wireless communication.

2.1.1 ATM Networks

It will be possible in the future, using broadband ISDN (B-ISDN) networks, to transmit the four basic services in telecommunications - speech, pictures, data and video - quickly and cheaply via a single network infrastructure. In 1988, ATM was selected by CCITT as the transport mechanism for B-ISDN networks, the basis on which the universal network concept is built [KYA95].

Asynchronous Transfer Mode (ATM) is a cell-based switching and multiplexing technology designed to be a general-purpose, connection-oriented transfer mode for a wide range of services. ATM is also being applied to the LAN and private network technologies as specified by the ATM Forum [McD95].

ATM handles connection-oriented traffic either directly or through adaptation layers, and connectionless traffic through adaptation layers. ATM virtual connections may operate at either a Constant Bit Rate (CBR) or a Variable Bit Rate (VBR). Each ATM cell sent into the network contains addressing information that establishes a virtual connection from its origin to a destination. All cells are then transferred, in sequence, over this virtual connection. ATM provides either Permanent or Switched Virtual Connections (PVCs or SVCs) [McD95].

ATM offers the potential to standardize network architecture, defining the multiplexing and switching method, with SONET/STM providing the underlying physical transmission standard for very high-speed rates. ATM also supports multiple Quality of Service (QoS) classes for differing application requirements on delay and loss performance. Thus, the vision of ATM is of an entire network constructed from ATM and ATM Adaptation Layers (AALs) switching and multiplexing principles to support a wide range of services, such as voice, packet data (SMDS, IP, FR), video, imaging, and circuit emulation [McD95].

A major benefit of the layered structuring of communications protocols is that intermediate elements within the network do not necessarily need to cope with the functions represented by each of the layers [CLA96]. Figure 1 illustrates the peer-to-peer communications which take place when two user end devices communicate with one another by means of an ATM transport switch. The ATM switch supports only the lowest three protocol layers, and relays information peer-to-peer with each of the ends. Meanwhile, at the AAL and higher layers, the two end devices communicate peer-to-peer directly over the connection established by the lower three layers [CLA96].

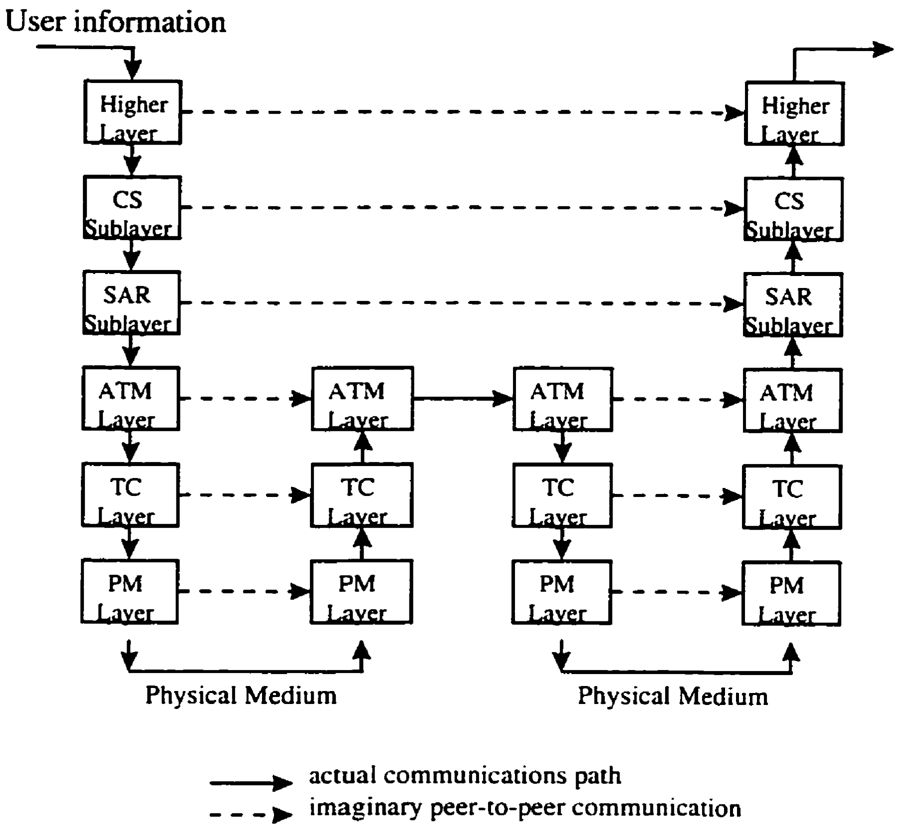


Figure 1. Protocol Layer representation of two end devices communicating via ATM Layer switch

2.1.2 Wireless Communications

Wireless systems use free space as the transmission media for radio signals sent via an electromagnetic signal. Electromagnetic radiation is created by inducing a current of sufficient amplitude and frequency into an antenna whose dimensions are approximately the same as the wavelength of the generated signal. The signal can be radiated uniformly (as in a light bulb) or can be directed as a beam of energy (as in a spotlight) [BLA96(a)].

Radio signals are radiant waves of energy transmitted into space and are similar to waves of heat or light. The radio frequency spectrum is divided and classified by frequency bands as shown in Table 1 [BLA96(a)].

Classification Band	Initials	Frequency Range
Extremely low	ELF	Below 300 Hz
Infra low	ILF	300 Hz - 3 kHz
Very Low	VLF	3 kHz - 30 kHz
Low	LF	30 kHz - 300 kHz
Medium	MF	300 kHz - 3MHz
High	HF	3 MHz - 30 MHz
Very high	VHF	30 MHz - 300 MHz
Ultra high	UHF	300 MHz - 3 GHz
Super high	SHF	3 GHz - 30 GHz
Extremely high	EHF	30 GHz - 300 GHz
Tremendously high	THF	300 GHz - 3000 GHz

Table 1. Radio frequency bands

The wireless network is an emerging communication technology that is focused on the local area network industry (i.e. wireless LANs). Presently, most LANs are configured with coaxial cable, twisted pair, or optical fiber. While these media have proven extremely beneficial, they have considerable installation and maintenance costs and lack the mobility offered by a wireless media. The wireless LAN industry has its origins in a FCC ruling [BLA96(a)] stipulating that certain frequency bands need not be licensed if they use low power transmitters and spectrum techniques (such as the use of a number of frequencies to transmit a signal). The products emerging in the wireless network marketplace transmit within a relatively limited area (a few thousand feet). Originally these products were used to connect to Ethernet LANs where a physical media was impractical [BLA96(a)].

Many mobile radio standards have been developed for wireless systems throughout the world, and more are likely to emerge. Table 2 lists the most common paging, cordless, cellular, and personal communications standards presently used in North America [RAP96].

Standard	Type	Year of Introduction	Multiple Access	Frequency Band	Modul-tion	Channel Bandwidth
AMPS	Cellular	1983	FDMA	824-894 MHz	FM	30 kHz
NAMPS	Cellular	1992	FDMA	824-894 MHz	FM	10 kHz
USDC	Cellular	1991	TDMA	824-894 MHz	Pi/4-DQPSK	30 kHz
CDPD	Cellular	1993	FH/ Packet	824-894 MHz	GMSK	30 kHz
IS-95	Cellular/ PCS	1993	CDMA	824-894 MHz 1.8-2.0 GHz	QPSK/ BPSK	1.25 MHz
GSC	Paging	1970's	Simplex	Several	FSK	1.25 MHz
POCSAG	Paging	1970's	Simplex	Several	FSK	1.25 MHz
FLEX	Paging	1993	Simplex	Several	4-FSK	15 kHz
DCS-1900 (GSM)	PCS	1994	TDMA	1.85-1.99 GHz	GMSK	200 kHz
PACS	Cordless/ PCS	1994	TDMA/ FDMA	1.85-1.99 GHz	Pi/4-DQPSK	300 kHz
MIRS	SMR/PCS	1994	TDMA	Several	16-QAM	25 kHz

Table 2. Major mobile radio standards in North America

2.1.3 Wireless ATM Networks

Tens of billions of dollars are expected to be invested in both broadband and wireless networks over the next decade. Current broadband network investments are predominantly in the area of ATM-based platforms, while demand for wireless communications are expected to grow rapidly. However, at present these two communications network activities are "disjoint". Although wireless technology could provide nomadic access capability to ATM-based networks, technical and service-related issues have prevented integration. Indeed, until recently the integration of wireless access and mobility with ATM has received little attention [HSI97].

There are several factors favoring the use of ATM cell transport for personal communication networks (PCN), including: flexible bandwidth allocation and service type selection for a range of applications (some of which are yet to be defined); efficient multiplexing of traffic from bursty data/multimedia sources; end-to-end provisioning of broadband services over wireless and wireline networks; suitability of available ATM switching equipment for inter-cell switching; improved service reliability with packet switching techniques and ease of interfacing with the wired B-ISDN systems that will form the telecommunications backbone [RAY94].

The strength of wireless ATM derives from the superior cost/performance characteristics of ATM switching technology, when used as a core infrastructure network with large traffic volumes, while also serving as a generic platform for both synchronous and packet services. Since ATM provides a superset of transport capabilities required by various wireless systems, such an infrastructure can provide plug-and-play support for various radio access technologies being deployed today (e.g., CDMA (IS-95), TDMA (IS-136), GSM/DCS-1900, PHS, GPRS, IEEE802.11, etc.). The same mobile ATM network can later be used to support high-speed, multimedia capable radio access technologies (e.g., Hiperlan or UMTS in Europe and Supernet/NII in the US) as they become available [RAY97].

The general concept of a mobile ATM network is shown in Figure 2. The basic idea is to integrate mobility support (typically via a switch software upgrade) into the ATM framework, providing facilities for both location management (finding the user) and handoff control (dynamically tracking the user). This is in contrast to conventional mobile networks available today, in which standard switches are used in conjunction with external mobility-related components such as a base station controller (BSC), mobility service controller (MSC), home location register (HLR) or visitor location register (VLR). The integrated ATM approach has the advantages of improving mobile switching performance, reducing the amount of equipment required (and hence cost), and does not require the network operator to make a-priori distinctions between mobile and static users.

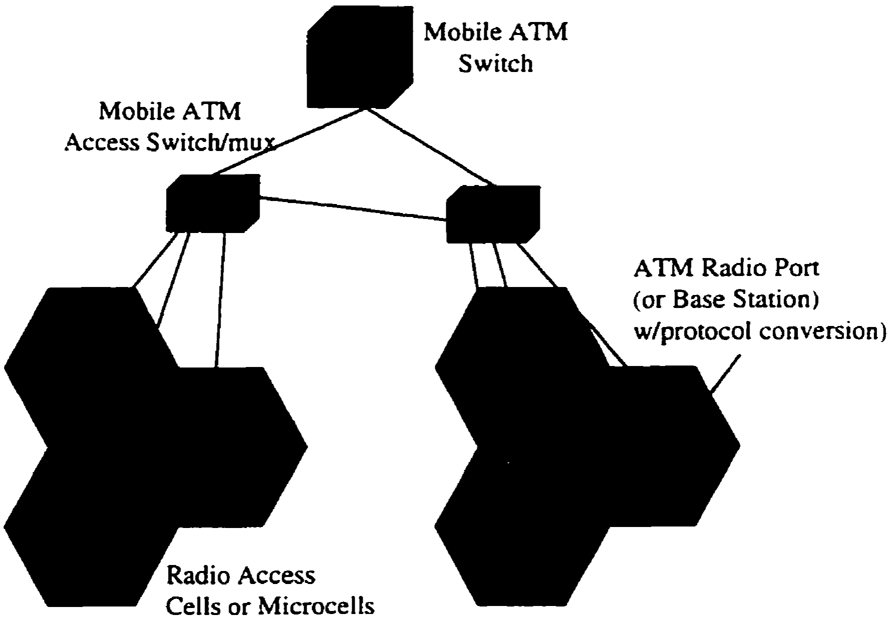


Figure 2. Mobile ATM network concept

The wireless ATM architecture is based on the concept of a seamless wired + wireless ATM network with end-to-end ATM transport, signaling and QoS control. This can be achieved by incorporating new wireless channel specific medium access control (MAC), data link control (DLC) and wireless control layers into the standard ATM protocol stack, while augmenting existing ATM control protocols to support mobility. The medium access protocol for wireless ATM access is required to support a mix of ABR (& UBR), VBR and CBR services with QoS control. One approach is to use the dynamic TDMA/TDD protocol with frame-by-frame allocation of ABR and VBR slots and periodic assignment of isochronous CBR slots. A protocol of this type has been shown to provide reasonable performance at throughput levels up to approximately 0.6-0.7 after accounting for control and physical layer overheads. The justification and more details are described in [RAY95].

A wireless ATM system consists of a radio access layer and mobile ATM network. These two major subsystems can be further divided into the following key design components:

(1) "Radio access layer" protocols

a) High-speed radio physical layer (PHY)

b) Medium access control (MAC)

c) Data link control (DLC)

d) Wireless control

(2) "Mobile ATM" protocol extensions

a) Handoff control

b) Location management

c) Routing and QoS control

The radio access layer consists of several new protocol sublayers necessary to extend ATM services over a wireless link. The major functions of this layer include high-speed physical level transmission/reception, medium access control for channel sharing by multiple terminals, data link control for amelioration of radio channel impairments, and wireless control for radio resource management and meta-signaling.

Mobile ATM is the term used to denote the set of enhancements needed to support terminal mobility within a fixed ATM network. The major functions of mobile ATM are location management for mapping of user names to their current locations, and handoff control for dynamic rerouting of VCs during terminal migration. Note that mobile ATM is intended to be independent of the specific radio access technology used. This means that in addition to supporting end-to-end wireless ATM services via the WATM radio access layer outlined above, mobile ATM capabilities can be used to provide an interconnection infrastructure for existing PCS, cellular and wireless LAN applications [RAY96].

2.2 Wireless ATM Network Management

This section discusses wireless ATM management issues. Specifically the issue of handoff for both communication and database applications is presented. This is followed by a brief discussion of QoS management and the potential utilization of artificial neural networks for improving or maintaining QoS requirements.

2.2.1 Mobility Management

2.2.1.1 Handoff Algorithm in Distributed Broadband Mobile Computing Networks

The mobility of nodes in mobile computing networks introduces a major new complication to be dealt with [SAL96]. These nodes move into and out of communication range of the fixed distributed network requiring that network topology charts be continuously updated by all the nodes. So the problem is focused on switching from one transmitter to another as the mobile node moves out of range of the former and into the range of the latter. McTiffin et al. [McT94] identify five techniques:

- *Parallel Transmission* - The mobile node communicates through a number of base stations in parallel. These paths are combined at the mobile node and at a point within the network. Additional network transmission capacity is required in proportion to the number of mobile nodes using this technique.
- *Selection Diversity* - The mobile node may communicate through any number of base stations on a frame-by-frame basis. There is no overall increase in traffic but the route taken by any one frame is indeterminate.
- *Hard Handoff* - The mobile unit communicates through a single base station until out of range. This results in a relatively long handoff to a new base station, most likely leading to a break in communications.
- *Soft Handoff* - The mobile unit communicates predominantly with one base station, but may be affiliated with several base stations in order to determine the best reception and route for data transfer.
- *Hybrid Schemes* - This involves using a combination of the above methods. It may be advantageous to use one method, such as selection diversity for transmissions to the mobile node (uplink) and another method (such as parallel transmission) for communications to the base node (downlink).

Salsbury et al. [SAL96] describes that the actual diversity technique selected depends on the type of service required and the network topology. Wireless mobile networks can take many forms such as communication via satellite link, communication locally within a building (LAN), or communication among a collection of shore stations, ships, and aircraft. The speed of the mobile units is an important factor in determining the diversity method [SAL96].

Elmagarmid et al. define call handoff for mobile hosts as the physical connection transfer between the old and new support station and service or connection transfer for mobile clients between the old and new information servers [ELM95]. Service handoffs are useful because they help reduce transmission delays caused by mobility. Note that the movement of mobile hosts in fixed networks may result in a long path of communications because the physical distance does not necessarily reflect network distance in mobile environments. Service handoffs may also help balance server workloads. In handoff decision making in a wireless network, the mobile client monitors the radio signal strength it receives from neighboring base stations while a call is in progress [ELM95].

According to Comer et al., the handoff algorithm is a transfer of mobile ownership (i.e., a base station hands off a mobile host to another nearby base station) [COR95]. Radio signal strength determines when a handoff should occur: if a new base station can maintain a better radio contact with a mobile host the base station that currently owns the mobile transfers ownership to the new base station. However, such a scheme requires hardware support, and one can not assume such support is available.

The design presented here uses software. In this design each base station includes a module that uses the following two criteria to determine when a handoff to a mobile host occurs :

1. A base station that receives a packet from a new mobile host consults the current owner before capturing the mobile.

2. The base station that currently owns a mobile is given priority in maintaining the ownership.

Without hardware support to measure the radio link quality between a base station and a mobile host, the protocol software uses frame reception to approximate link quality [COR95]. For example, if a mobile client who is making banking transactions emits a frame that is received by both the owning base station and a nearby base station, the protocol deduces two equally good links regardless of the actual quality of each individual link. Similarly, if the nearby base station receives the frame but the owning base misses it, the protocol deduces that the nearby base station has a better link than the owner.

Like the initial capture protocol, the handoff protocol uses broadband network control circuits to exchange control messages. When a base station receives a frame from a mobile that it does not own, the base station sends a message across a control circuit to the mobile's owner. The protocol requires each base station maintain a timestamp for each mobile host. Whenever an owning base station receives a frame transmitted by a mobile, the base station updates its timestamp for the mobile. When a message arrives from another base station that has received a frame from the mobile, the owning base station uses the timestamp information to determine whether to hand off the mobile host to the other base station or retain ownership.

In [COR95] the handoff algorithm is explained by way of example, which is developed here as a main reference. Assume that: Base station B owns a mobile host, M; M emits a frame, and the frame is received by nearby base stations. If B receives the frame, B updates the timestamp that corresponds to M and forwards the frame. Other base stations that receive the frame, buffer the frame, send a HANDOFF message to B, and start a handoff timer that expires after $T_{handoff}$ seconds. When B receives a HANDOFF message, B computes Δt, the difference between the time at which the HANDOFF arrives and B's timestamp for M. If Δt is greater than $T_{hthresh}$, B answers with a HANDOFF_ACK message, allowing the sending station to capture M; otherwise, B answers with a HANDOFF_NACK message, denying the sending station's handoff request. After B sends a HANDOFF_ACK, B denies subsequent handoff requests (i.e., requests from other base stations that have received a frame from M). The value $T_{hthresh}$ is normally a fraction of a second (e.g., 50 ms). Because the HANDOFF message traverses the distributed broadband network with little delay, B deduces that M is still in its area when Δt is less than $T_{hthresh}$. If Δt is greater than $T_{hthresh}$, B deduces that it missed the packet that caused a base station to send the HANDOFF message and allows the station to capture M.

In addition, the base station that receives a HANDOFF_ACK cancels the handoff timer, forwards the buffered frames and broadcasts a control message to declare the ownership of the captured mobile. A HANDOFF_NACK causes a base station to cancel the handoff timer and discard the buffered frames. When a mobile is situated in an overlapping area, the handoff timer allows a non-owner base station to capture the mobile in case the current owner station fails. Expiration of the handoff timer triggers a base station to use the initial capture protocol to capture a mobile [COR95].

When a mobile stays in an overlapping area, multiple base stations may receive frames from the mobile. If each frame from the mobile causes multiple base stations to send HANDOFF messages to the owner base station, the owner station may be overwhelmed with HANDOFF requests. To avoid this problem, base stations bound the rate at which they send HANDOFF messages. Specifically, a base station imposes a delay of at least T_{hdelay} seconds between successive HANDOFF messages.

2.2.1.2 Replication Algorithm in Distributed Broadband Mobile Computing Networks

Replication is simply maintaining several copies of an object while providing the illusion of a single, fault tolerant object [FAE96]. In a distributed system replication often makes sense for the purpose of increased locality or even disconnected operation. System designers have several motivations for using a replication strategy. The most common reasons for using a replication strategy are:

- *Fault Tolerance.* Distributed systems feature partial failures, which make them ideal candidates for replication. If one object replica fails, the other replicas are likely to be working, and service to the user can continue.
- *Autonomy.* Large scale systems are often administrated by multiple administration bodies (i.e., they are autonomous). These bodies often benefit from multiple copies of some objects, giving them direct access to their 'own' object. A replication scheme is needed to maintain integrity among the objects.

- *Performance.* In large scale distributed systems, communication delays can have serious effects on performance. Using a replication scheme, it is possible to achieve better performance by using a replica that is close to the processor. In a sense, replication is like caching, except it is concerned with a more heterogeneous communication infrastructure and hence non-uniformity of invocation delays and semantics.

Replication faces many challenges in distributed object systems. Most notably replication can incur severe overheads. Clearly, it is more costly to manage multiple objects than a single one [FAE96]. Some of the factors contributing to these high costs are :

- *Isolation.* One of the goals for a replication scheme is to provide transactional semantics for the replicated object. Users can then share the object without worrying about potential conflicts with other concurrent users of the object. Transactional semantics are strictly necessary to guarantee correct execution in the presence of multiple users. However, enforcing the ACID properties known from transaction processing theory is expensive, and is a research area in its own. In particular, locking strategies, the techniques used to enforce isolation, are complex.
- *Redundant system resources.* Secondly, the redundancy requires allocation of extra resources in the system, e.g. in the form of memory and CPU cycles.
- *Mutual consistency.* Maintaining mutual consistency in the replica group also requires communication, and this is expensive (and error prone), especially in large, heterogeneous distributed systems.

The following section identifies some of the potential problems encountered in the use of replication schemes :

- *Inconsistency*. Maintaining the illusion of only one copy of some object while using several may lead to inconsistency among the replicas.
- *Scalability*. In a large scale system, a preferred replication scheme is one that does not enforce any limits on the number of replicas. In particular, in a setting where replication is employed primarily for performance, locality increases with the number of replicas.
- *Performance*. Maintaining consistency among several object replicas requires communication and cooperation. This may severely limit concurrency in access to the replicas. The goal is to reduce amount of communication while maintaining consistency among the replicas.
- *System Support*. A replication scheme would be perfect if it allowed any object to be replicated completely and transparently. However, the semantics of the application often dictate which replication scheme is appropriate.

Many of these problems are related (sometimes in subtle ways). A replication scheme or replication algorithm is employed to provide abstractions to the user of a replicated system.

Faiz et al. introduce some replica control algorithm variations, which are based on the primary copy method [FAI95]. They include:

1) ASAP (as soon as possible). This protocol performs in such a way that "write" operations are executed on a primary copy. Committed "writes" are collected and sent to all other copies as independent transactions.

2) Quasi copies method. The control of information is carried out at a single central site, but the method is also applicable in case of multiple central sites. Coherency conditions are defined which allow some deviations between an object and its copy.

3) Differential file algorithm. A differential file is used to record the changes made on the primary copy.

4) Differential refresh algorithm. The algorithm is initiated by sending the snaptime to the base relation and checking it against the time associated with the base tuples.

5) Copy token method. The copy that holds the token is regarded as the primary copy. If the copy holding the token becomes unavailable, a new token is handed out.

Sandhu and Zhou describe the algorithms used for replication in distributed systems. Replicas are created and maintained on the servers on demand (i.e., accesses to a file from another cluster causes a replica of that file to be created in the accessing cluster). Synchronization of replicas is performed at file close time. Choice of file close consistency for synchronization of replicas is motivated by performance goals. The performance penalty of synchronizing replicas after every write operation will be high, and the benefits comparatively negligible given that most applications do not require such a high degree of consistency [SAN96].

There are two basic techniques for the synchronization of a replicated set of data : immediate update and invalidation [SAN96]. With immediate update, modification of a file causes all replicas to be updated as soon as the file is closed. The drawback with this technique is the potential number of wasted updates. That is, replicas may be updated many times before they are actually used by another site and the cost of sending updates after every modification may be prohibitive. With invalidation, on the other hand, a modification to a file causes all other replicas to be invalidated at file close time. Updates are then requested on demand (i.e., at file open time) by a site with an invalid replica, thus eliminating the problem of wasted updates. However, invalidation also has drawbacks. First, requiring replica sites to acquire updates on demand leads to higher file access latency, whereas, with immediate update, updates can be performed asynchronously. Second, although we are not explicitly concerned with fault tolerance, it is obvious that the chances of losing data are higher with invalidation, since, using this technique, it is possible for only a single replica of the file to exist at one time.

A third technique, denoted partial update, represents a compromise between invalidation and immediate update [SAN96]. Using partial update, a site other than the current owner is sent updates to the file, but all other sites are invalidated. If this "alternate site" is one that frequently accesses the file, then average latency is reduced, thereby lowering overhead cost. Moreover, partial update has the additional advantage of reduced risk of data loss (compared to invalidation) because at least two sites will always have valid replicas.

The immediate replication algorithm described in [SAN96] is similar to the eager replication algorithm, described in [GRA96]. Eager replication keeps all replicas exactly synchronized at all nodes by updating them as part of one atomic transaction. Eager replication gives serializable execution – there are no concurrency anomalies. However, eager replication reduces update performance and increases transaction response times because extra updates and messages are added to the transaction.

Two-tier replication is advised in [GRA96] as a near-ideal replication scheme. Replicated data items have two versions at mobile nodes: Master Version and Tentative

Version. The most recent value is received from the object master. The version at the object master is the master version, but disconnected or lazy replica nodes may have older versions. The local object may be updated by tentative transactions. The most recent value is maintained as a tentative value because of local updates. Similarly, there are two kinds of transactions: Base Transaction and Tentative Transaction. Base transactions work only on master data, and produce new master data. They involve at most one connected-mobile node and may involve several base nodes. Tentative transactions work on local tentative data and produce new tentative versions. They also produce a base transaction to be run at a later time on the base nodes. The key properties of the two-tier replication scheme are:

- mobile nodes may make tentative database updates.
- base transactions execute with single-copy serializability so the master base system state is the result of a serializable execution.
- a transaction becomes durable when the base transaction completes.
- replicas at all connected nodes convert to the base system state.
- if all transactions commute, there are no reconciliations.

Replication timing is important and is described in [SCH96]. An application need not worry about the timing of the asynchronous distribution of data to target sites. Getting this functionality from the replication server means less programming. More details are described in [SCH96]. The replication server should also provide several alternatives for timing. Some examples are:

1) ASAP. In this case the data is moved through the queues and replication server as fast as possible.

2) Scheduled. (as determined by the system administrator). In this case, data remains in the replication server until it is scheduled for distribution.

3) Triggered. In this case timing is determined by user defined criteria such as an event happening. The number of records exceeds a limit or time of day. When that trigger is fired, the server moves the data to the distribution queue for remote processing.

4) Under manual control. The user determines when the replicas are made consistent.

In the work presented here, the replication algorithm is implemented with the handoff scheme simultaneously. Although the environment presented does not represent a full featured database system (e.g. no concurrency), the replication algorithm illustrates a technique for having data follow a mobile user access base stations.

2.2.2 QoS Management

Quality of Service (QoS) provides an unifying theme on which the functions and facilities of the new integrated standards can be constructed. For future applications, especially highly interactive applications and those relying on the transfer of multimedia information, it is essential that QoS is guaranteed system-wide, including the distributed system platform, the transport protocol and the multiservice network. Enhanced communications protocol support such as end-to-end QoS negotiation, renegotiation, indication of QoS degradations and co-ordination over multiple related connections, is required [HUT96].

2.2.2.1 Wireless ATM QoS Management

If we consider the basic scenario of wireless networks connected to ATM networks, we find that one of the primary difficulties is in meeting ATM end-to-end quality of service (QoS) requirements [CHE95]. Even if we ignore the physical transmission error (which is usually much more frequent in wireless links than in wired links), wireless access is more difficult than wired access due to mobility, interference, and imperfect carrier sensing. It is noted that an effective and reliable access in wireless generally transforms a distributed contention into a centralized operation such as polling [CHE95][GOO89].

In existing wired QoS architectures the approach is, where possible, to mask out change. This presents a stable service to the application (effectively providing a level of network transparency). This is achieved by combining QoS functions such as admission control, resource reservation and maintenance. However such an approach is not sustainable in a mobile environment. Consequently, in such an environment emphasis should not be on transparency but rather on making information available to applications and empowering applications to make the necessary changes. Such an approach relies on the provision of QoS management functions supporting monitoring and adaptation. To complement this, the underlying system must be amenable to adaptation, including the ability to adapt to periods of disconnection. This is not to say that existing approaches to QoS management are invalidated. Rather, strategies such as admission control and resource reservation can be used for component networks and provide guarantees while the end system remains connected to the network [BLA96(b)].

2.2.2.2 Neural Network based ATM QoS Estimation

In spite of the fact that modern digital computers have made great leaps in both speed and processing power, certain tasks remain difficult to perform optionally because of the complexities of the following tasks: optimization, generalization, and classification. Neural networks are better suited to performing certain complex tasks quickly because they can be trained with relatively little data in applications where a more traditional or deterministic algorithm may not be known [KEU95].

The potential benefits expected from the use of neural networks for traffic control include:

1) Adaptive learning. Neural networks learn from observations or examples during the course of network operation and therefore do not require detailed mathematical understanding of the underlying network being controlled.
2) High computation rates. The massive parallelism of the hardware implementation allows neural networks to achieve high computation rates.
3) Generalization from learning. Neural networks have the ability to generalize learning to problems yet to be encountered. This ability is particularly important or relevant to congestion control in ATM networks where observations may be incomplete, delayed or only partially available.
4) High degree of robustness. Neural networks are robust and have fault tolerance due to the nature of distributed processing.

A neural network is a massively parallel distributed processing system that is suitable for storing knowledge and making it available for use. It is generally a multiple-input and multiple-output nonlinear mapping, which can learn an unknown nonlinear input-output relation from a set of examples. A neural network-based QoS estimation method has been shown to enhance the performance in ATM management [SHE97].

Neural networks run on algorithms that automatically adapt to new network situations. The use of neural networks investigated here are for use as tools that measures network traffic, classifying it for use in meeting QoS requirements, and hence predicting its future behavior over a certain monitoring period [TAR95].

Neural networks have been applied to the cell level control functions such as QoS management, traffic measurements, traffic enforcement (policing) and rate-based feedback congestion control at the access point to a network. Neural networks have also been applied to network control functions such as optimal link capacity assignment and dynamic routing because of their ability to solve optimization problems [TAR95].

In ATM networks, congestion occurs when too many cells are destined for one node. Some cells will be queued (cell delay) and some may even be discarded (cell loss). This affects the user's communication services. Thus, ATM traffic management or congestion control is needed to guarantee the quality of service (QoS) specified by parameters such as Cell Loss Rate (CLR) and Cell Delay Variation (CDV).

Figure 3 shows the CDV, CLR and other QoS parameters of one channel observed by an ATM switch with the help of a neural network. The parameters are regarded as inputs to the neural network which, using the trained weights and biases, produces an output telling the switch the current QoS level of the channel (excellent, good, fair, bad, or in between). The switch then can act accordingly [SHE97].

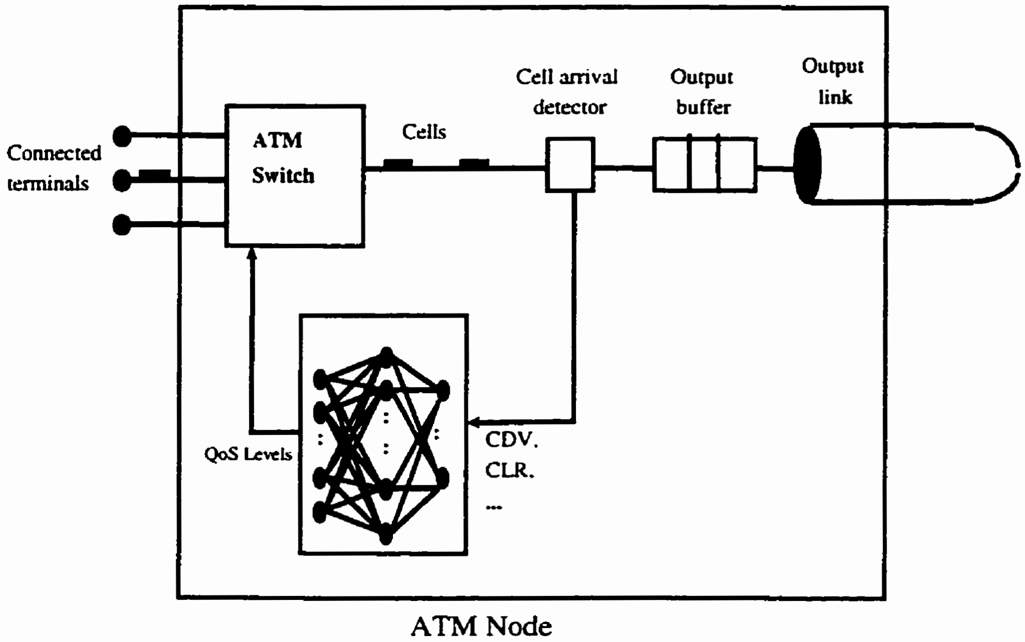


Figure 3. Neural network based QoS estimation

2.3 Variance Fractal Dimension Based Traffic Estimation

This section introduces the notion of the variance fractal dimension as a descriptor for estimating traffic for use within the network management scheme described in section 2.2.

2.3.1 Introduction

Observations of both Ethernet traffic and variable bit rate (VBR) video traffic revealed that they both exhibit "self-similarity" and /or infinite asymptotic index of dispersion for counts (IDC) [ADD95].

Fractal analysis is one of the tools used to analyze the traffic in the network. A series of measurements are conducted with the testbed capturing samples corresponding to cells transmitted in a real wireless ATM LAN. The variance fractal dimension ([Kin95], [PEI92]) has been suggested as a suitable metric and was used to analyze batch or real-time received traffic parameters, such as bit errors, in the wireless ATM LAN. The variance fractal dimension is calculated on small windows within the complete signal or received traffic parameters, based on the variance of increments for selected time periods. The temporal sequence of the variance fractal dimension values from sequential, evenly spaced windows form a variance fractal dimension trajectory. In this study, the received traffic parameter, bit errors per packet (bers_per_pk), are analyzed and the trajectory obtained. Changes in level at the boundaries between serious noise interference and normal transmission condition can be identified and represent a means to enhance the performance of the network management system.

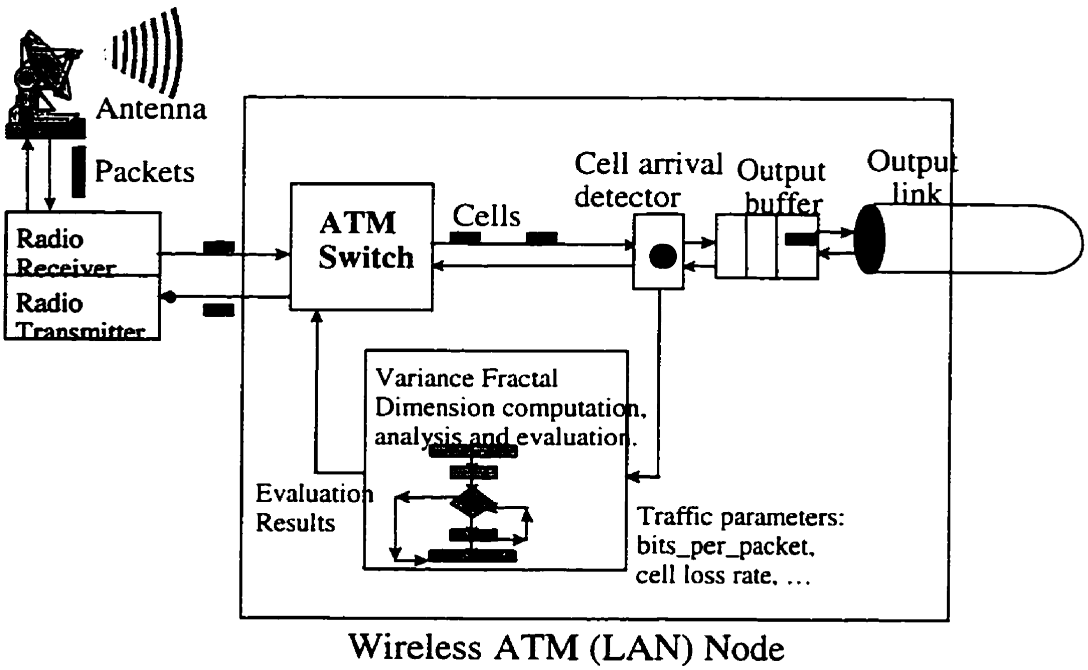


Figure 4. Variance fractal dimension based wireless ATM LAN evaluation

As shown in Figure 4, the bers_per_pk and other traffic parameters of radio channels that need to be monitored can be captured by an ATM switch. They are regarded as the input data of the variance fractal dimension computation. After the analysis and evaluation by the programmed software, the switch will obtain the results of the evaluation. By analyzing the results, the traffic can be estimated. Depending on the evaluation results as well as other results analyzed by other mechanisms, the network management can act accordingly, such as by sending handoff or re-transmission requests [SHE97].

2.3.2 Advantages of Variance Fractal Dimension

The technique for computing the variance and, therefore, the Hurst exponent which leads to the variance dimension, $D\sigma$, is simple enough to lend itself to real-time fractal analysis of dynamic communication traffic [PEI92] [GRI96].

Although the spectral dimension may reveal the multifractal nature of the underlying process for different short-term (windowed) Fourier analysis, the choice of the window size is difficult, especially, since small numerical errors may greatly affect the solution and introduce artifacts. Variance fractal dimension avoids these problems as it does not require a window in the Fourier sense, and therefore does not introduce the window artifact. Variance Fractal dimension provides a new approach to studying signals because rather than analyzing signals (strongly correlated events) contaminated by noise (with a weaker correlation), it analyzes noise contaminated by signal [Kin95] [GRI96].

2.4 OPNET

This section overviews the simulation environment used throughout the studies presented here.

2.4.1 OPNET Overview

OPtimized Network Engineering Tool (OPNET) is a comprehensive software environment capable of simulating large communications networks with detailed protocol modeling and performance analysis. OPNET features include: graphical specification of models; a dynamic, event-scheduled simulation kernel; integrated data analysis tools; and hierarchical, object-based modeling. OPNET's hierarchical modeling structure accommodates special problems such as distributed algorithm development. OPNET delivers open systems methodology and an advanced graphical user interface known as the MIL 3 User Interface [MIL96].

There are eight OPNET tools which fall under three broad categories:

- Model Development: Networks, Node, Process, Parameter.
- Simulation Execution: Probe, Simulation.
- Results Analysis: Analysis, Filter.

The Network, Node, and Process Editors are the primary development tools, responsible for defining models at each level of the modeling hierarchy. The Network Editor is used to construct network models. Depending on the type of network being modeled, a network model may consist of communicating nodes connected by point-to-point links, bus links, or radio links. Nodes and links can be placed within sub-networks, which can then be treated as single objects in the network model. This is useful for separating the network diagram into manageable pieces and provides a quick way of duplicating groups of nodes and links. Typically, initial design efforts in a project will focus on developing node models for each distinct type of node.

The Node Editor is used to create models of nodes. The node models are then used to create node instances in networks using the Network Editor. Internally, OPNET node models have a modular structure. A node is defined by connecting various modules with packet streams and "statistic wires". The connections between modules allow packets and status information to be exchanged between modules. Each module placed in a node serves a specific purpose, such as generating packets, queuing packets, processing packets, or transmitting and receiving packets.

Process models will initially be place-holders, but they will be successively refined during model development. The Process Editor is used to define models for the processes that run in processor and queue modules. At the core of most OPNET simulations are user-defined process models. Process models can represent the logic of communications hardware, network protocols, distributed algorithms, or high-level client-server processes. OPNET simplifies the construction of process models by offering graphical representation of extended finite state machines. The network model, consisting of nodes and links, must be defined before the network simulation can be built.

The Parameter Editor is used in conjunction with process and node model development to define special model structures. The Parameter Editor has six modes, each used to define a particular construct including: Packet Formats, ICI Formats, Probability Density Functions, Link Models, Modulation Functions, and Antenna Patterns.

Simulation Execution typically involves the preparatory step of attaching probes to the points of interest in the model using the Probe Editor. Probes monitor statistics computed during a simulation. Once a simulation executable has been produced, the Probe Editor is used to specify statistics to be monitored, or modules to be animated during the simulation. Users may "probe" built-in or user-created statistics; or they may create probes which activate animation data collection. The Simulation Tool allows specification of a simulation sequence which uses particular input and output options. The Simulation Tool is used to set up and execute simulations. Each simulation set object graphically represents one or more simulations. Simulation sets are defined by selecting the network model to be executed and then specifying values for input parameters such as a random number seed, duration, and output file names for probed data.

The Results Analysis tool includes an Analysis Tool which is used to evaluate the results of a simulation that presents statistics gathered during simulations in the form of two dimensional graphs or text listings. The information collected by a simulation can be viewed directly, or processed by the Filter Editor to generate new sets of data. The Filter Editor supports user definition of mathematical processing of simulation results [MIL96].

2.4.2 Proto-C and Kernel Procedures

Processes are expressed in a language called Proto-C which is a combination of state transition diagrams (STDs), Kernel Procedures (KPs) and the C programming language. A process model's STD defines a set of primary states that the process can enter. For each state, the STD defines the conditions for the process to move to another state. Transitions in STD describe the possible movement of a process from state to state and the conditions under which such changes may take place. Proto-C models allow actions to be specified at various points in the Finite State Machine.

KP is a library of high level commands (over 300 procedures) that can be called from within process models, Transceiver Pipeline stages, C functions which have been scheduled as interrupts, or C functions which are directly or indirectly invoked from one of these contexts. For the most part, the simulation services are accessed through KPs. KPs are categorized by nineteen primary functions, based on the types of objects they attempt to manipulate. The collection of KPs within a category is called a package, and KPs within the same package share a common package keyword in their procedure names. Table 3 lists the capabilities provided by the KP libraries [YAO96].

Package	Applications
Anim	Support for custom animaton development.
Dist	Probability destribution and random number generation.
Ev	Event list and event property query; event cancellation.
Ici	Formal interfaces between processes; association of information with interrupts.
Id	Identification of objects
Ima	In-simulation query and modification of object attributes.
Intrpt	Query of interrupt properties; control of interrupt handling.
Pk	Creation, destruction, modification, analysis, and transmission of packets.
Prg	Programming support: linked lists, memory, sting manipulation, debugging.
Pro	Creation, invocation of processes; process group shared memory.
Q	Queuing statistics; high-level queuing operations.
Rte	Basic routing support for static routing inplementations.
Sim	Simulation control: Customized messaging, simulation executing control.
Stat	Custom statistic generation; Intermodule signalling via statistic wires.
Strm	Communication between modules via packet streams, packet delivery.
Subq	Low-level queuing operations: antenna patterns, modulations.
Tbl	Accessing of tabular data: antenna patterns, modulations.
Td	Setting and getting of transmission data for custom link models.
Topo	Query of model topology (e.g., for automatic discovery and configuration).

Table 3. Major simulation Kernel Procedure packages

2.5 Summary

This chapter provided a background survey of the thesis. First of all, a wireless ATM network was introduced. Then wireless ATM network management was described, including mobility management and QoS management. Several algorithms were overviewed such as handoff schemes and replication algorithms. Two tools or facilities were introduced for use in network management, these were a neural network and the variance fractal dimension. The chapter concluded with an overview of the OPNET simulation environment used throughout the thesis.

CHAPTER 3 WIRELESS ATM MOBILITY MANAGEMENT MODELING

Chapter 3 presents details concerning the wireless ATM model used in the study and presents results from the handoff algorithm implemented. The final section of the chapter deals with a data replication scheme and how it is modeled within OPNET.

3.1 Model Description

The broadband wireless ATM model developed here contains a network-level model, which is composed of several nodes. Each node in turn contains its own node-level model, which is composed of network components such as radio transmitters, radio receivers and processes in which the algorithm code is written. These components are described following the description of the network-level model.

3.1.1 Network-level Model

Figure 5 shows the network-level of the broadband wireless ATM model. This network model consists of five kinds of nodes.

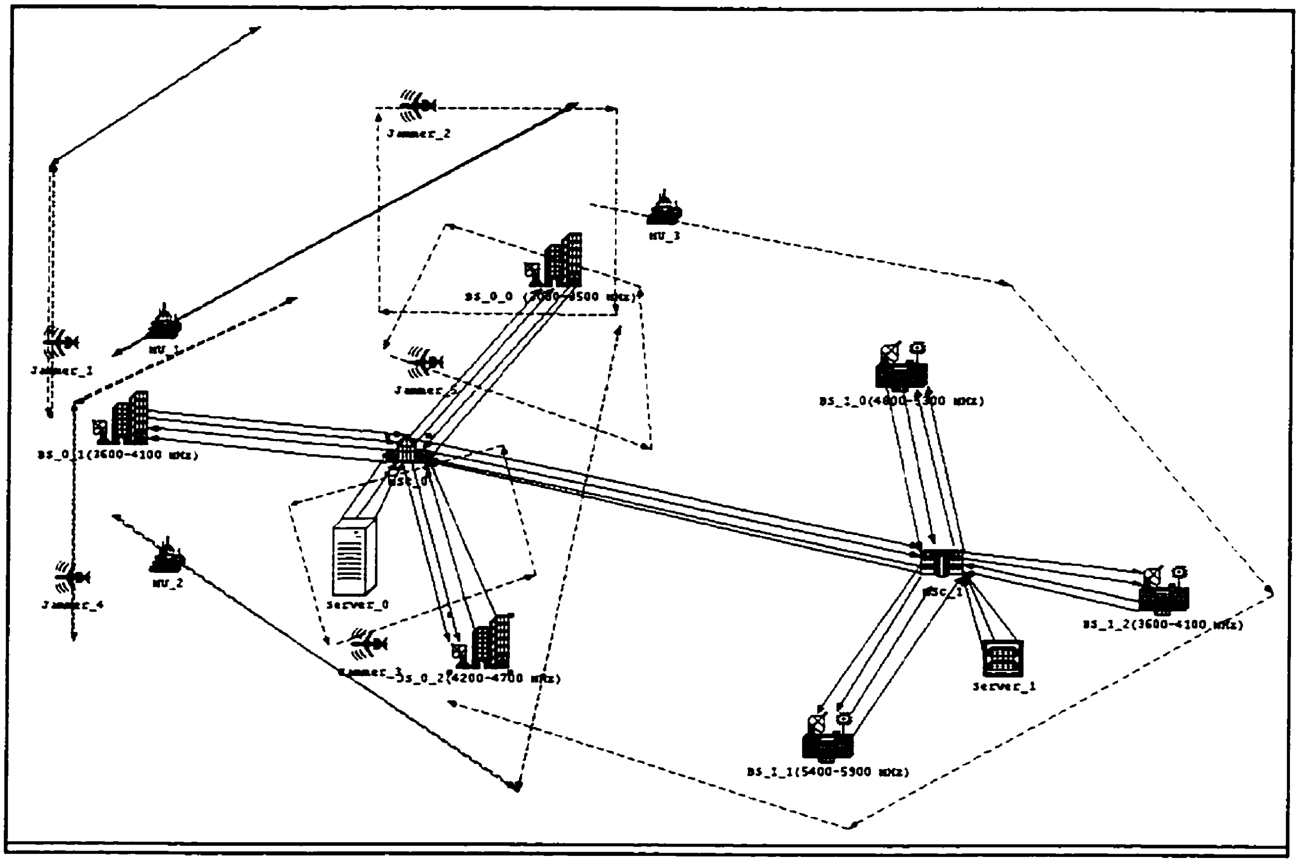


Figure 5. Wireless ATM network-level model

1) *Mobile Switch Center (MSC).* There are two MSCs in the model, MSC_0 and MSC_1. They are fixed nodes that switch the data stream between mobile units and servers, and control the mobility and QoS guarantee schemes. These two MSCs are connected with a broadband transmission optical network, such as SONET. In this model, each MSC connects with three base stations.

2) *Base Station (BS).* Three base stations (BS_0_0, BS_0_1, and BS_0_2) are connected to MSC_0 and three (BS_1_0, BS_1_1, and BS_1_2) are connected to MSC_1, using a broadband transmission optical network. The main function of the BS is transferring the packet stream between mobile units and a MSC. Recall that each BS

has its own frequency range (Table 1). For example, the frequency range of BS_0_0 is from 3000-3500 MHz. Because of limited frequency resources, frequency is reused under some rules as in this model, where BS_0_1 and BS_1_2 are allocated the same frequency range (3600-4100 MHz). Each BS controls a geographical region or cell according to its transmitter's power. In this model, the transmitting power of each BS is 10 mW, and the bandwidth of each channel is allocated to 1MHz.

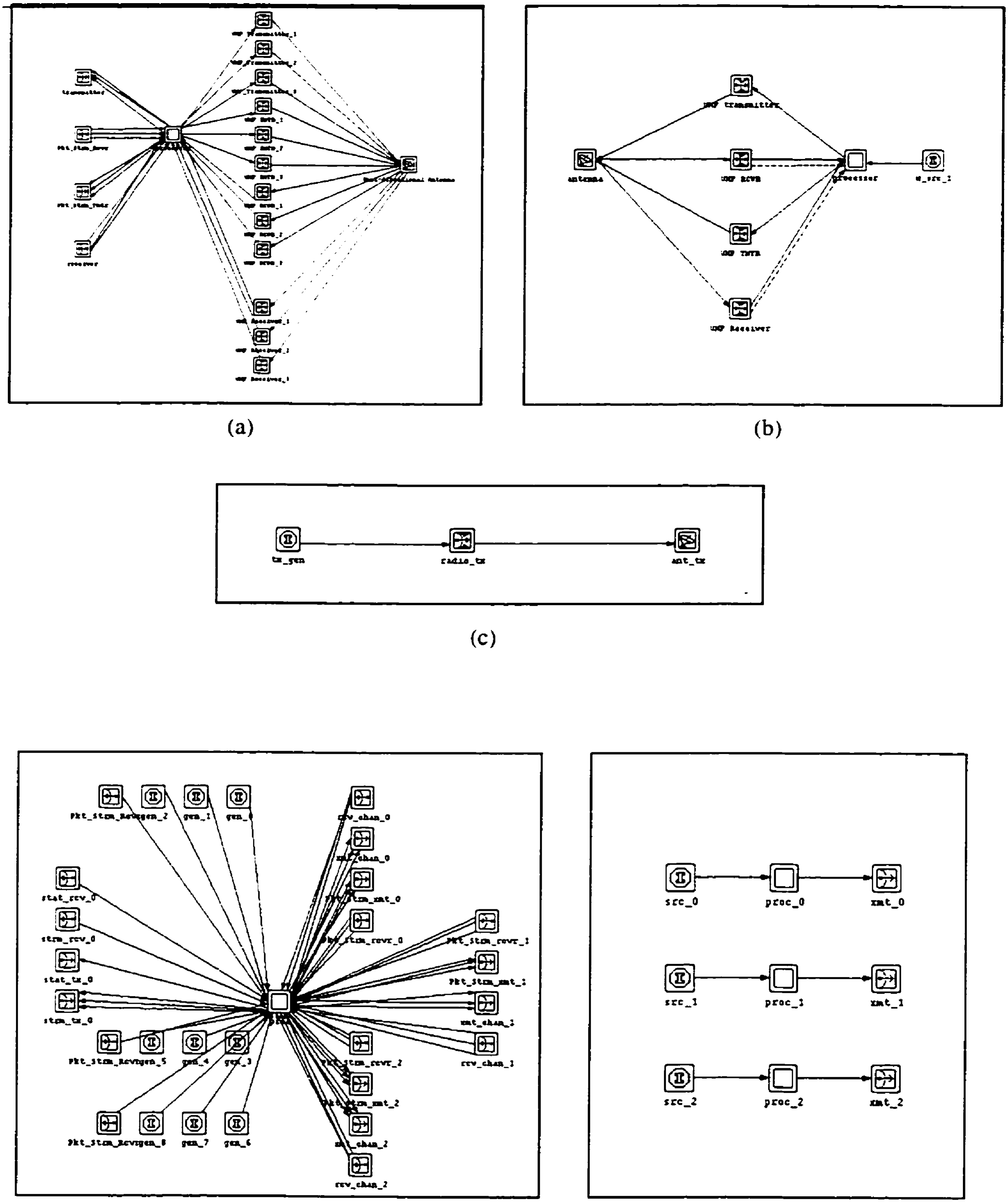
3) *Mobile Unit (MU).* There are three MUs in this model -- MU_1, MU_2 and MU_3. In figure 5 the dotted line attached to each MU indicates its roaming trajectory. It takes 50 seconds (simulation time) for the MU to move from one point to the next along the straight dotted line. For example, in the first 50 seconds, MU_1 moves from BS_0_1 to BS_0_0, MU_2 moves from BS_0_1 to BS_0_2, and MU_3 moves from BS_0_0 to BS_1_0. In the second 50 seconds, according to their trajectories, MU_1 moves back to BS_0_1, MU_2 moves to BS_0_0, and MU_3 moves to BS_1_2 and so on. In order to communicate with the BS, the frequency of the MU in that cell should match the frequency of the BS. For example, if the transmitting and receiving frequency of one channel in BS_0_0 is 3000 MHz and 3100MHz respectively, the MU's receiving and transmitting channel should be adjusted to 3000MHz and 3100MHz. In this model, the function of the MU is just like a mobile client, which is able to access the fixed server using a client-server protocol. The transmitting power of each MU is 1 mW, and the bandwidth is allocated to 1MHz.

4) *Server.* Server_0 and Server_1 in this model are connected to MSC_0 and MSC_1 respectively. Each server has its own databases which can be accessed by any mobile unit. Communication between the client-server is bi-directional.

5) *Jammer.* Jammer is an interference source that affects the receiving quality of the MU. The transmitting frequency of the jammer is the same as the receiving frequency of the MU. The jammer causes the quality of service (QoS) parameters, such as BER, to not conform to the traffic contract. In this model, Jammer_1 and Jammer_2 interfere with MU_1 when it moves in the BS_0_1 cell and the BS_0_0 cell. At the same time, Jammer_4, Jammer_3 and Jammer_5 interfere with MU_2 when it roams into the BS_0_1 cell, the BS_0_2 cell and the BS_0_0 cell. The jammers are also mobile, their trajectory represented by dotted lines. The mobility is identical to that of the MU. The transmitting power of each jammer is 0.01 mW.

3.1.2 Node-level Models

There are 5 types of Node-level Models in the wireless ATM model; they are illustrated in Figure 6 (a) – (e), and discussed below.



(a) (b) (c) (d) (e)

Figure 6. Node-level models

a) BS node. Logically there are two transmitters and two receivers connected to the MSC node. One pair is used for communication packet stream and the other pair is used for network management data stream. In addition, there are 6 radio transmitters and 6 radio receivers connected to an omni-directional antenna with 2 pairs each used for the communication packet stream and the network management data stream between mobile units (MU_1, MU_2 and MU_3). Each radio transmitter or receiver has initial settings for parameters such as power, data rate and bandwidth. The core of this node is the user-defined process model, which represents the logic of communication hardware, network protocols, distributed algorithms and data statistics. This process model is described in more detail in 3.1.3 (1).

b) Mobile Unit (MU) node. Each MU logically has 2 pairs of radio transmitters and receivers which are connected to an antenna, one for the communication packet stream and the other for the network management data stream. In this node model, an ideal generator generates the user's application data packets. Characteristic of these packets, such as packet length, data rate, etc., can be set initially or changed dynamically during the simulation time. The core of this node model is the user-defined process model, which represents the logic of communication hardware, network protocols, distributed algorithms and data statistics. This process model is described in more detail in 3.1.3 (2).

c) Jammer node. Each jammer node consists of an ideal generator, a radio transmitter and an antenna. Because of the jammer's interference function, it only transmits special packets designed to interfere with the mobile users. The transmitting frequency should be set as same as the receiving frequency of the mobile users with whom the jammer interferes.

d) Mobile Switch Center (MSC) node. Transmitters and receivers are each connected to their own base stations, servers, and the other MSCs. Again, one group is for communication packet stream and the other for network management stream. There are also several ideal generators, which generate network management data stream for all base stations. Again, this process in this node is a core one. The network protocols, switch functions, distributed algorithms and data statistics are all represented in this user-defined process model, which are discussed in detail in the following subsection 3.1.3 (c).

e) Server node. The server node is also simple. The function of the server is to generate communication data for each MU. Since there are 3 MUs in the network model, there are 3 generators in the server node. The parameters of each generator can be set initially or changed dynamically during the simulation time.

3.1.3 Process Models

The core process model represents the logic of communication hardware, network protocols, distributed algorithms and data statistics. Process models are instantiated as processes in the node domain and exist within processor or queue modules. Processes can be independently executing threads of control that perform general communications and data processing functions. They can represent functionality that would be implemented both in hardware and in software. The process editor specifies the behavior of process models and defines the model's interfaces (i.e., the aspects of the process model which are visible to the user). The interface includes the attributes and statistics of the process model.

Process models use a finite state machine (FSM) paradigm to express behavior that depends on current state and new stimuli. FSMs are represented using a state-transition-diagram (STD) notation. The states of the process and the transitions between them are depicted as graphical objects.

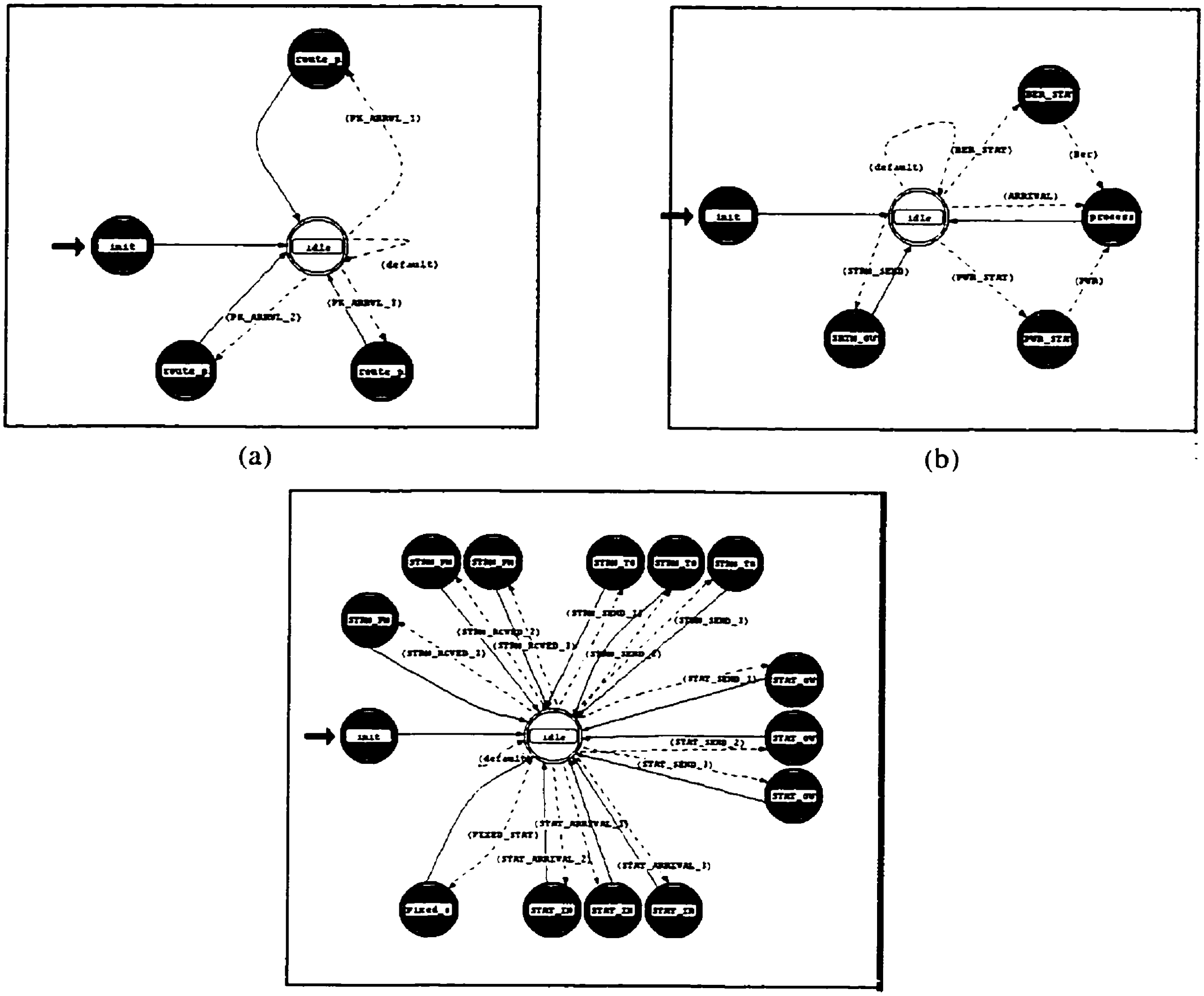
The state object represents the attained "mode" of the process, based on previous stimuli and corresponding decisions. States contain code expressing processing that is executed immediately after they are entered, or immediately before they are exited. A state can be forced or unforced. A process blocks immediately after executing the "enter" code of an unforced state, at which point it waits for a new interrupt before continuing. More details are described in [MIL96].

The transition object indicates the possible paths a process may take from a source state to a destination state (each state can be the source and destination of any number of transitions). A transition contains a condition statement which specifies the requirements for the process to follow the transition. An executive statement specifies actions that are to be taken when the process follows the transition.

The process editor provides operations to support the creation and editing of process models. Operations related to major capabilities are: create object, edit object, set initial state, edit model attributes, edit simulation attributes, edit model attribute interfaces, edit statistics, declare external object files and compile process model.

The components of a process model include a FSM diagram with embedded C statements, and various blocks containing code for variable declarations, macros, constants, and function definitions. These components are collectively termed Proto-C, since they define a variant of the C language specialized for protocols and distributed algorithms. The FSM diagram represents the functional flow of the process with an easily interpreted diagram of states and transitions. States typically contain C-syntax statements, called state executives, which perform a specific task. Events and conditions determine the migration from state to state through transitions.

For the process models developed here the initial state starts the state of the FSM where the packet count is initialized. The idle state is the one where the FSM waits for other states. Once the initial state is executed, the FSM will transition between the idle and other states.



(a) (b)

(c)

Figure 7. Process-level models

Figure 7 illustrates the 3 main process models for the : BS node, MU node and MSC node respectively.

a) BS process. In the base station process model, there are 3 "route" states (other than the initial and idle states), each one for the 3 MUs. The route states and receive transmit packets to and from MSCs to MUs. Probes are inserted in these states for data statistics.

b) MU process. Besides the initial and idle states there are 4 others: STRM_OUT, BER_STAT, PWR_STAT and process. The stream-out (STRM_OUT) state controls the data transmission from ideal generator to radio transmitter. The function of Bit-Error-Rate statistics (BER_STAT) state is to test the received QoS parameters (such as BER) and transfer them to the process state. The power statistics (PWR_STAT) state has a similar function, except that the transfer parameter is received power from the base station. The process state includes several algorithms and protocols, such as QoS guaranteed, error-free handoff, etc.

c) MSC process. In addition to initial and idle states there are 5 other types of states: STRM_FM, STRM_TO, STRM_IN, STRM_OUT and Fixed_s. Fixed_s state controls the data communications between two MSCs. STRM_FM, STRM_TO states are defined for network management data stream while the other two states are for communication data stream. Because there are 3 MUs in this model and each MSC owns 3 base stations, each one of these states has 3 duplications.

CHAPTER 4 WIRELESS ATM QoS MANAGEMENT MODELING

Chapter 4 extends the network model presented in Chapter 3 to include provisions for mobile data and QoS management. This is accomplished through the use of a replicated data handoff algorithm, dynamic power control of the base stations, and frequency hopping guided by the use of a neural network based controller.

4.1 Model Description

The model used for wireless ATM QoS Management is the same as that used for wireless ATM mobility management. Actually, both work together during the simulation time. The model description is the same pattern as in section 3.1.

4.2 Neural Network Models

Different from the traditional mathematical calculations, neural networks are thought to provide fast, flexible, adaptive and intelligent control in network management. A neural network is a massively parallel distributed processing system that is suitable for storing knowledge and makes it available for use. It is generally a multiple-input and multiple-output nonlinear mapping, which can learn an unknown nonlinear input-output relation from a set of examples. A neural network-based QoS estimation method is investigated to enhance the performance in ATM management.

For this study, the "Matlab Neural Network Toolbox" software from Math Works Inc., was used on a Sun Sparc 20 at TR*Labs*. Two main methods of training were studied:

a) Learning Vector Quantization (LVQ), a supervised method.

b) Self-Organizing Maps (SM), an unsupervised method.

4.2.1 Learning Vector Quantization (LVQ)

Learning Vector Quantization (LVQ) is a method for training competitive layers in a supervised manner. LVQ networks learn to classify input vectors into target classes chosen by the user.

The LVQ Architecture is shown in Figure 8.

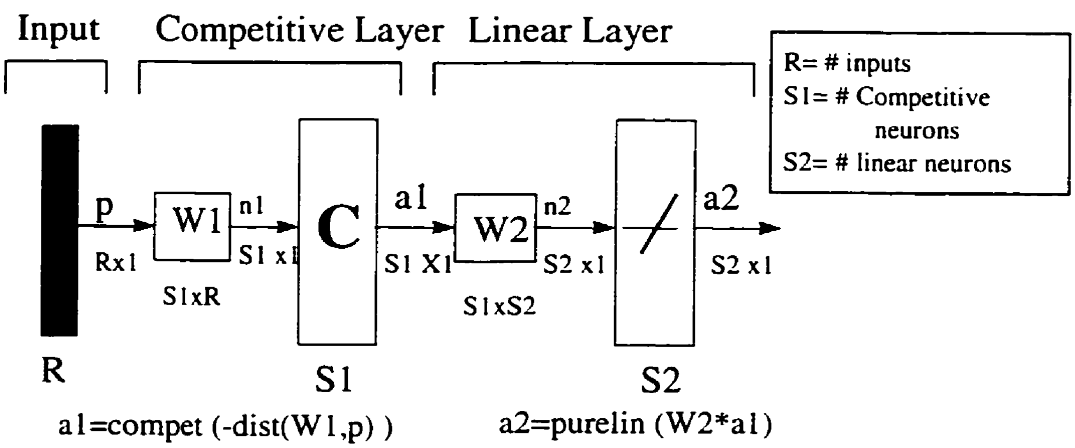


Figure 8. LVQ network architecture

An LVQ network consists of two layers. The first layer is a competitive layer, which learns to classify input vectors. The second layer (linear layer) transforms the competitive layer's classes into target classifications defined by the user. R, S1 and S2 represent the number of input, competitive neurons and linear neurons respectively. Each layer has its weight (W1 or W2). a1 is the output of competitive layer while a2 is the output of linear layer. "compet( )" is competitive transfer function while "dist( )" calculates distances between vectors. This means that if "a1" is the winning neuron then the distance between a1's weight vector and input vector is the shortest. "purelin( )" means the linear activation transfer function.

Figure 9 shows the relationship between epochs and error. An epoch means each pass through all of the training input and target vectors. The error represents the difference value between each two epochs. There are two curves in the figure: one represents the training time and the other represents the testing result. This figure shows that if the epochs increase or decrease too much, the result will not be ideal. If the learning rate is too small, it will take too long to train, if it is too large, the neural network will be unstable.

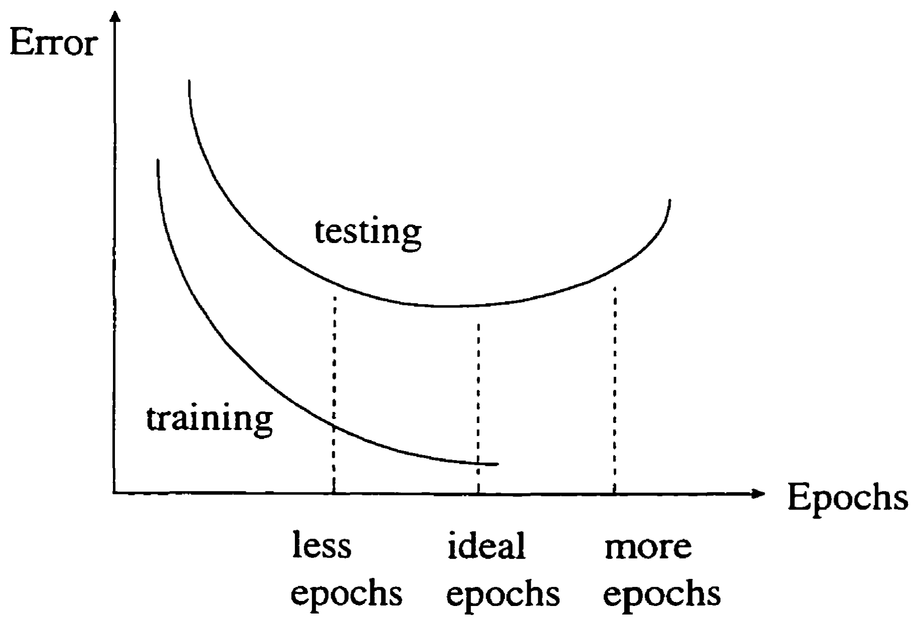


Figure 9. Epochs vs. Error

4.2.2 Self-organizing Map (SM)

A self-organizing map learns to detect regularities and correlations in the inputs and adapts accordingly. A SM also learns to recognize groups of similar input vectors in such a way that neurons physically close together in the neuron layer respond to similar input vectors.

Figure 10 shows the SM architecture. It is a competitive network with some modifications. The competitive neurons are ordered physically in one or more dimensions.

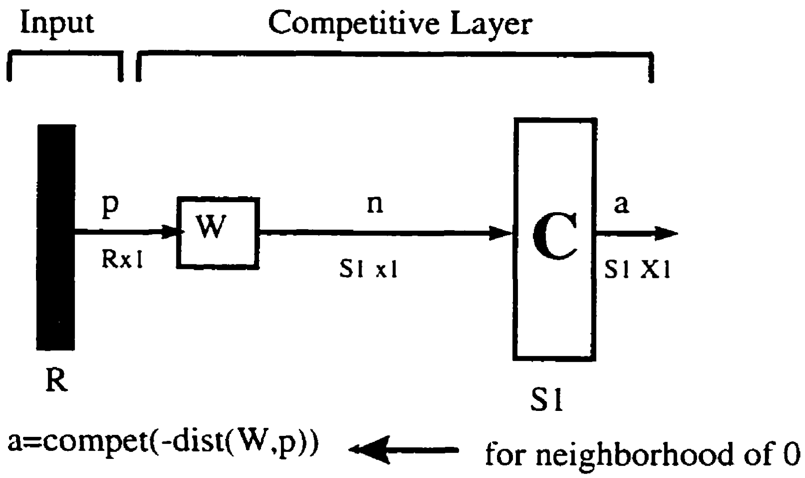


Figure 10. SM network

4.3 Simulation and Analysis

This section presents the experimental description and simulation results.

4.3.1Experiment Description

For the work reported here the total simulation time was 200 seconds.

Recall that in a wireless mobile network, when a mobile unit moves from one cell to another, a handoff occurs. In order to ensure continuous communication and a smooth data stream shift, an error-free handoff algorithm was implemented. In this model the error-free handoff algorithm was based on the received power measured by the mobile user as it transits the trajectory. Each BS counts the number of packets sent from the MSC as they pass through en route to each MU.

Since we are modeling an environment where data moves with the user, it was necessary to develop a replica algorithm to ensure the data moved is consistent during the handoff. A counter in the MSC counts the number of database packets that are replicated from one base station to another.

In wireless ATM, the communication medium between the mobile units and base stations is the air so interference is a potential problem. Consequently, the Quality of Service (QoS) guarantee is very important in a wireless LAN. Bit-Error-Ratio (BER) and Cell Transfer Delay (CTD) are the critical parameters in QoS management. The Quality of Service should conform to the traffic contract when the mobile units access the fixed network. In this model, jammers are simulated to interfere with the receiver of the mobile unit so that the QoS parameters will be seriously tested. The purpose of a QoS guarantee algorithm is to ensure communication quality when the mobile unit accesses the network and the BER and/or CTD do not comply with the traffic contract. In this thesis, a BER guarantee algorithm has been developed and will be active if the BER measure does not comply with the traffic contract.

There are two parts to the BER guarantee algorithm: automatic-frequency-hop (AFH) and dynamic-power-control (DPC). The first operates, if the BER received by the

MU is larger than the traffic contract and there exists at least one available, idle channel in that BS area. The receiving frequency of MU and the transmitting frequency of BS should hop automatically and simultaneously to the available channel in order to avoid the interference from the jammers. The AFH is an efficient algorithm for QoS management in broadband wireless ATM networks.

The DPC is implemented if all the channels in that BS area are occupied or busy when the received BER is high. In this scenario it is impossible for both MU and BS to hop to another frequency channel and avoid the interference. The transmitter power of the BS should be dynamically adjusted by the algorithm according to the BER level.

Both of these BER guarantee algorithms could be implemented using in-band management messages. In our model, we assumed that during the entire simulation time (200 seconds), MU_1 cannot frequency hop because there is no available channel. There are, however, enough channels for MU_2 to hop to when the interference is serious. In summary, the dynamic-power-control algorithm is used for MU_1 access and the automatic-frequency-hop algorithm is used for MU_2 access.

4.3.2 Collected Statistics

There were 3 types of local probes were selected for statistics in wireless ATM mobility management modeling:

1) MU received power.
2) BS counter to calculate the number of packet send from the MSC to each MU.
3) In MSC for counting the number of database packets replicated during handoff.

There were 4 types of local probes used for statistics in wireless ATM QoS management modeling:

1) BER for each MU. This probe retrieves the Bit-Error-Rate statistic of each receiving MU.
2) Average_BER for each MU. This probe records the statistics of average BER of each receiving MU.
3) Power for each BS. This probe records the power adjustment of each base station.
4) Frequency for each BS and MU. This probe records the frequency changes for each base station and mobile user.

4.3.3 Simulation Results

4.3.3.1 Error-free Handoff Algorithm

Figure 11 (a)(b)(c)(d)(e)(f) shows the results of a handoff algorithm that assumes error-free communicating. The first three are the results of the error-free handoffs of MU_1, MU_2 and MU_3 in the broadband wireless ATM model. The last three are the results of the receiving power of each MU.

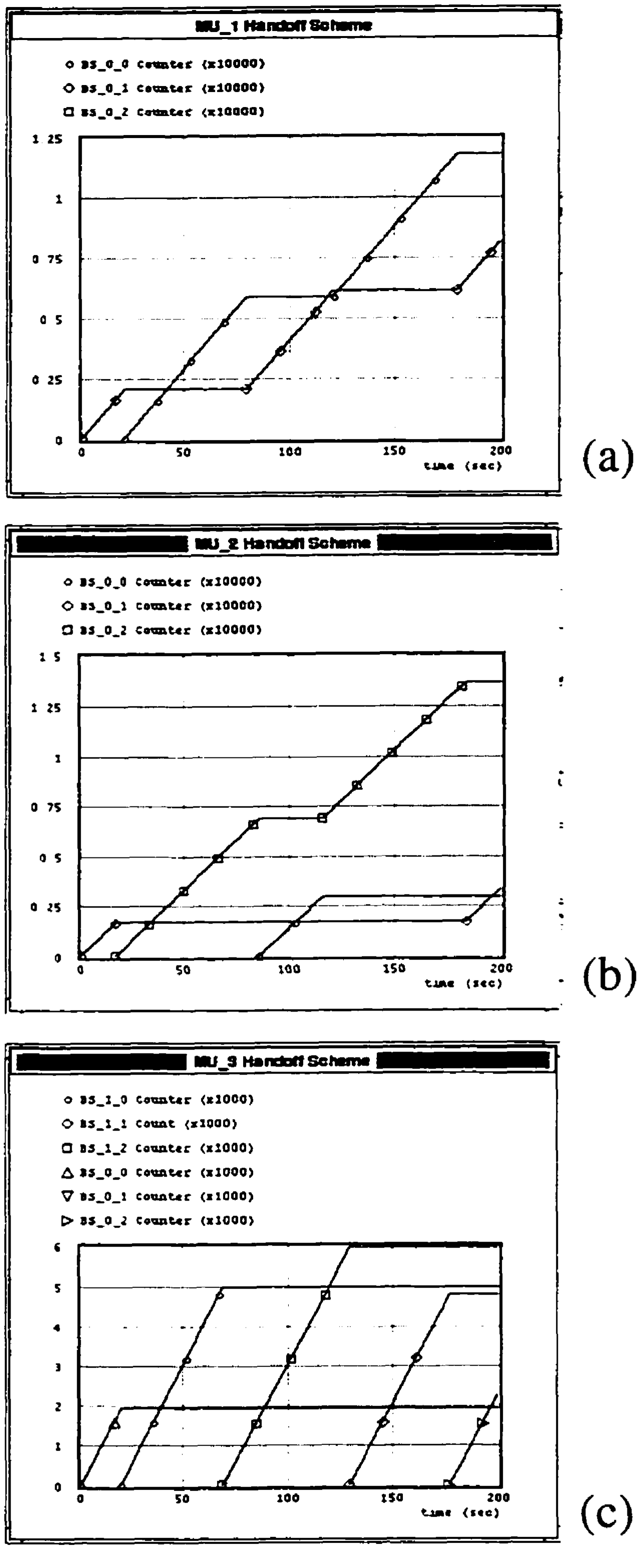
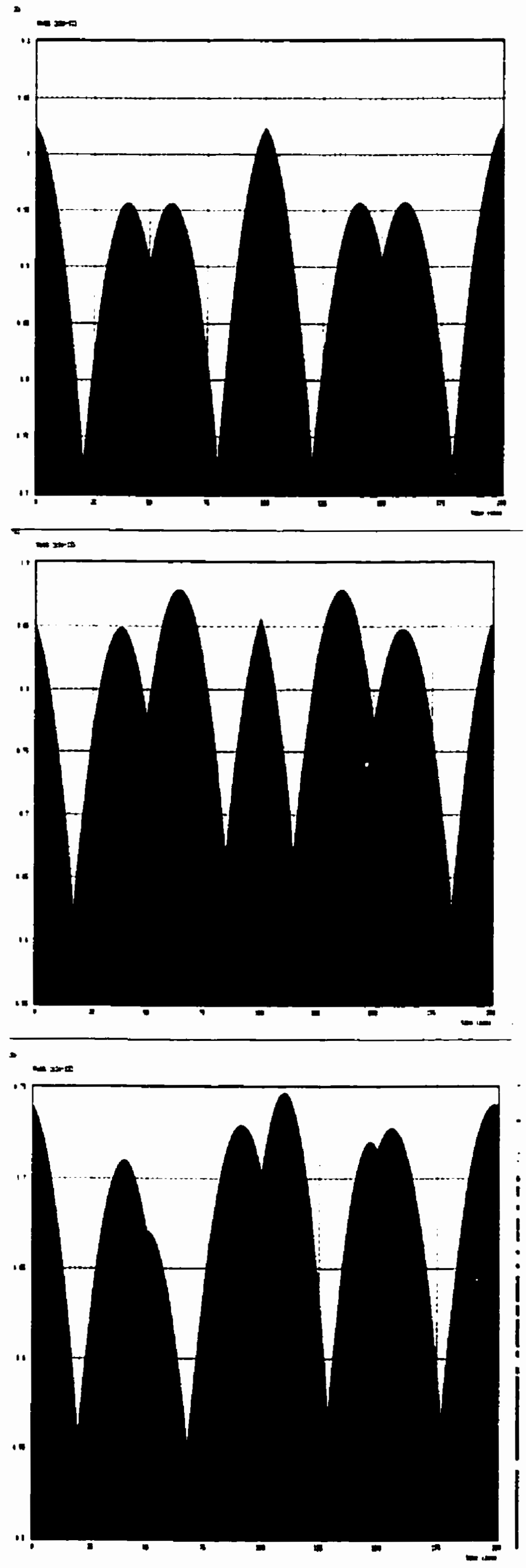


Figure 11 (a) - (c). Results of error-free handoff algorithm and receiving power for MUs.



(d)

(e)

(f)

Figure 11 (d) – (f). Results of error-free handoff algorithm and receiving power for MUs.

The example of MU_2 (b), (e) is used to explain the algorithm and the data presented. In (b), X-axis represents "time" while Y-axis is the number of packets counted by that BS. Initially, the mobile unit is near BS_0_1 so the received power from that station is the highest (see (e)). It decreases during the first leg of the trajectory while the received power from BS_0_2 increases. At about 20 seconds, the signal strength from BS_0_2 is stronger than that from BS_0_1. This is where the handoff occurs. We see that at the beginning the counter in BS_0_1 which is communicating with MU_2 increases. At the handoff point, about 20 seconds, MU_2 switches its communication data stream from BS_0_1 to BS_0_2. After the handoff, the counter in BS_0_2 increases while the counter plateaus. Meanwhile, MU_2 continues to communicate with BS_0_2 until (about 85 seconds) the receiving power from BS_0_0 is stronger than that from BS_0_2, another handoff occurs and the process repeats. At 100 seconds MU_2 reaches the end of its trajectory and retraces its path back to the origin. Subsequent handoffs occur at approximately 115 seconds and 180 seconds. At the end of the simulation time (200 seconds), MU_2 arrives back at its origin. All the transitions occur precisely at the expected handoff points and the figure indicates that the packet stream shifts to the BS with the best signal strength.

The error-free algorithms for MU_1 and MU_3 are the same as those for MU_2. MU_1 roams between BS_0_1 and BS_0_0, while MU_3 moves through the five BSs and across two MSCs. The results in Figure 11 illustrate that the algorithm works appropriately for all MUs.

4.3.3.2 Replica Algorithm

Because the problems in distributed database network are related to mobile communication, I employed a replication scheme, or replication algorithm, as a starting point to model aspects of mobile distributed data management. This replication algorithm is implemented via the handoff scheme simultaneously throughout the simulation. In this thesis, the replication algorithm focuses on the consistency scheme of the replicas because the consistency of the object replicas in both servers is very important to the error-free handoff scheme. For instance, when handoff occurs at the overlapping area, the replicas in both servers should be consistent in order to make the current service transparent and correct. If not, the handoff will be prone to error.

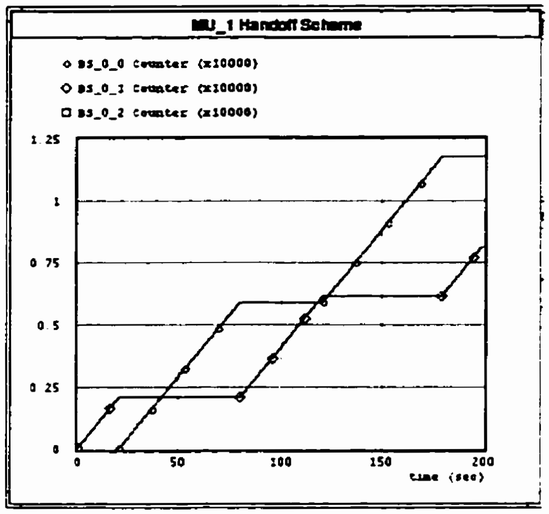
MU_1 is selected as an example to describe the replica algorithm (see Figure 5). MU_1 is moving between BS_0_0 and BS_0_1, which are both connected via a mobile switch center (MSC) to broadband networks (such as ATM, SONET). Replicas are created and maintained on the base stations on demand (i.e., accessing an object from BS_0_1 causes a replica to be created in BS_0_0).

In this simulation, the immediate update technique was used. With immediate update, modification of an object in BS_0_1 causes the replica to be updated in BS_0_0 as soon as the object is closed. Therefore, the object replicas in both BS_0_1 and BS_0_0 are absolutely consistent at any time, including during handoff. The drawback with immediate update is the potential number of wasted updates. That is, replica BS_0_1 may be updated many times before they are actually used by BS_0_1 making cost of sending updates after every modification prohibitive. Our network is small, so the problem was not obvious, but it would be in very large networks. Consequently, in large networks other techniques such as invalidation and partial update would have to be implemented. This is the subject of future work.

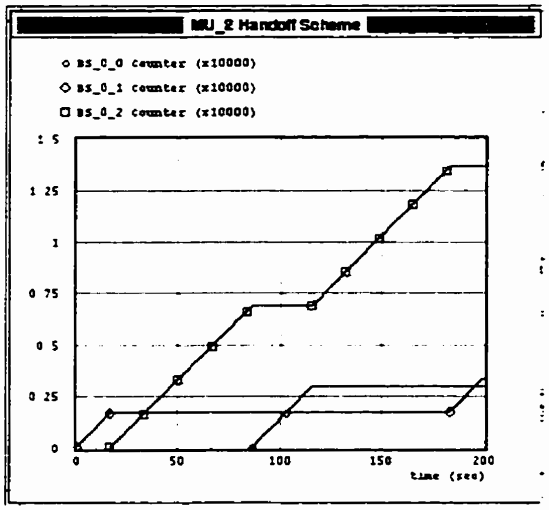
Figure12 (a)(d) shows us simulation result of replica algorithm for MU_1. Because the handoff algorithm is implemented four times during the simulation, the replica algorithm is also utilized 4 times. Using the "trace" function in the OPNET, we observed that at the 18.78 seconds point (a), the handoff request was dispatched initiating the replica algorithm (d). After, with both databases in the two base stations consistent, at 20.78 seconds, the mobile client switches from BS_0_1 to BS_0_0 using the error-free handoff algorithm. The second handoff point is similar. First the replica algorithm begins at 77.79 seconds. After 2 simulation seconds (at 79.79 seconds), the databases are consistent and the error-free handoff occurs. The scenario is repeated at the third and fourth handoff points (say 120.68 and 179.99 simulation seconds). For each replication, we expected the size of replicated object to be different, and it was according to the counter in MSC.

MU_2 ((b)(e)) and MU_3 ((c)(f)) are similar.

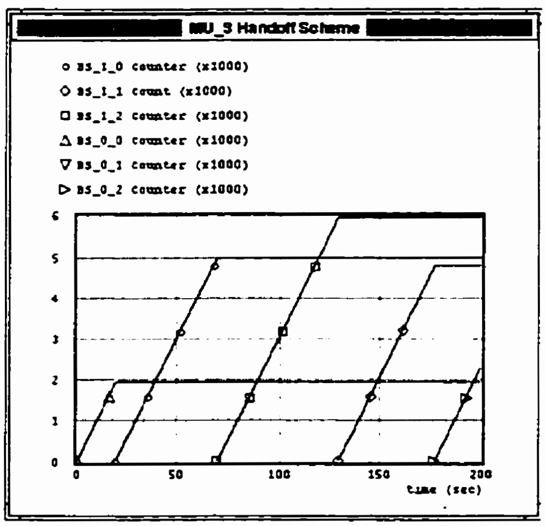
In this study, only a simple replication algorithm was investigated in a distributed database environment and it was not implemented completely. More details should be investigated in this model. Deep studies in this area will be done in future work.



(a)

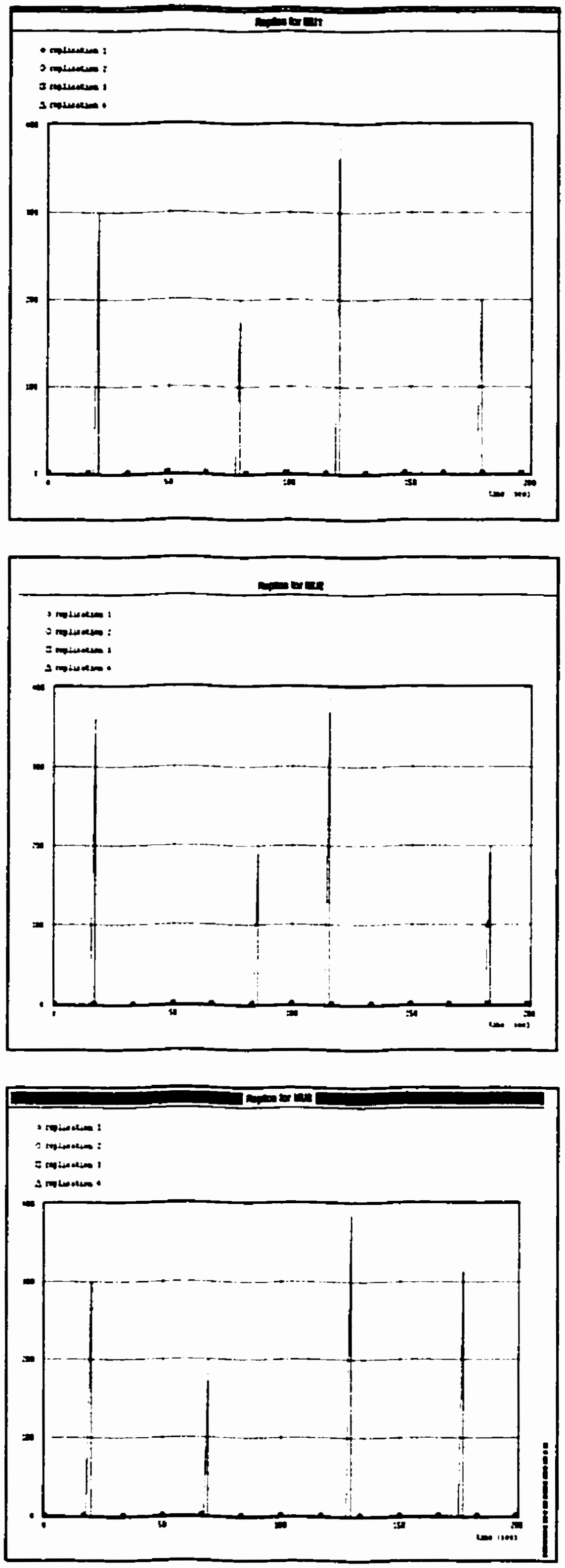


(b)



(c)

Figure 12 (a)- (c). Results of replications algorithm



(d)

(e)

(f)

Figure 12 (d) – (f). Results of replications algorithm

4.3.3.3 SM for Network Management Model

A neural network is an intelligent tool with potential use in network management. In this section an artificial neural network known as a self-organizing map (SM) is used to determine when to implement the dynamic power control or frequency hopping algorithm. In this wireless ATM QoS management modeling, because there are no target vectors as training data, a self-organizing map (SM) was selected as better method to investigate the problem than a supervised neural network algorithm.

The data selected for training were cell Bit-Error-Rate (BER) and Cell Transfer Delay (CTD). All the data were normalized from 0 to 1. Self-organizing maps differ from conventional hard competitive learning in the manner in which neurons get their weights updated. Instead of updating only the winner, feature maps update the weights of the winner and its neighbors. Neighboring neurons, therefore, have similar weight vectors and are responsive to similar input vectors. In this study, 25 neurons were used. Figure 13 shows the results of self-organizing maps after training. In Figure 13, the ' + ' represents the experience data (input vectors) while the dots represent the weight vectors (output vectors). After 5000 cycles, the map was evenly spread across the input space.

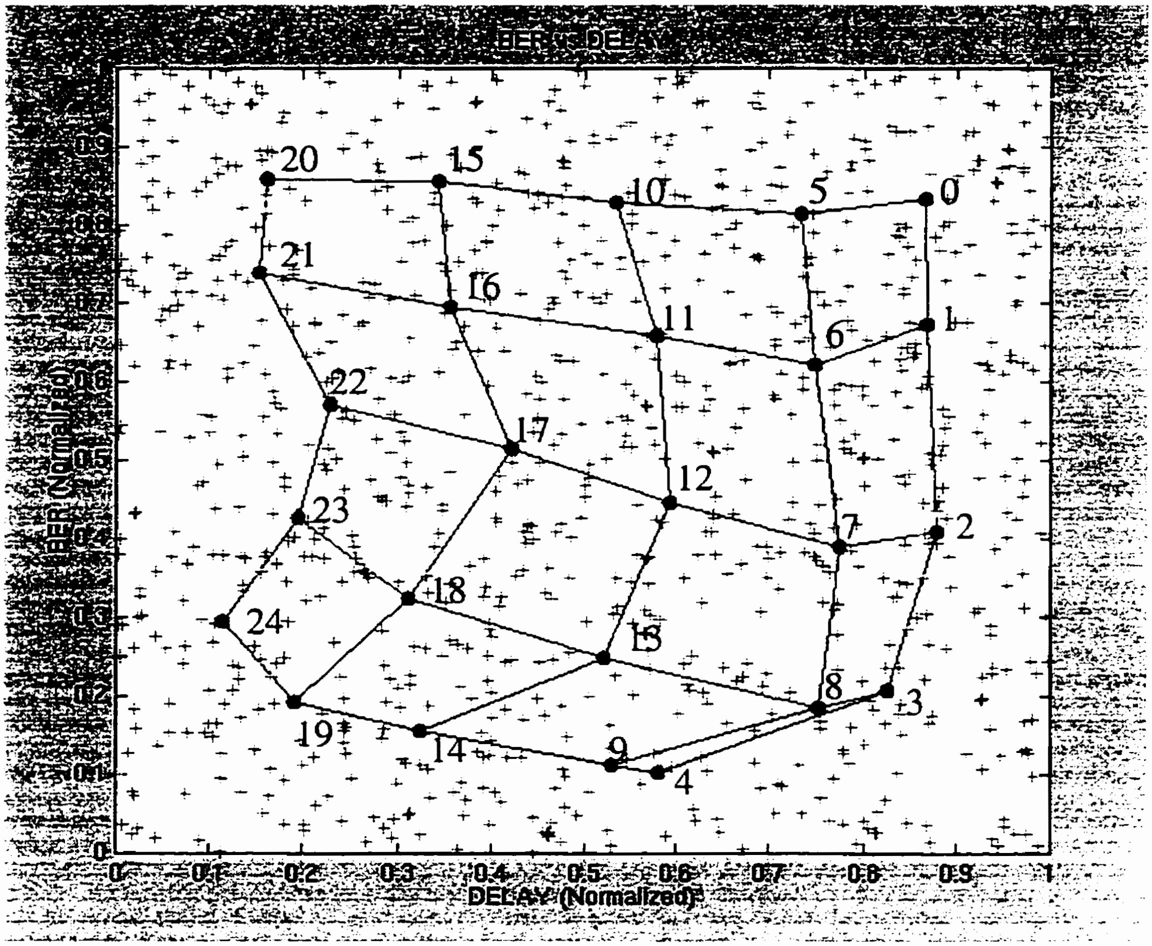


Figure 13. Result of self-organizing maps training.

After data training, the neural network has enough intelligence to decide the output for each new input vector.

Table 4 shows the classification of the output number vs. QoS guarantee management solutions. The output points of 19 and 24 represent lower delay and BER than others. In this study, these two points are regarded as tolerable condition and no QoS guarantee algorithm is used. The points of 20-23 represent the low delay value but high BER value. So if the output is one of these points, the automatic-frequency-hop or dynamic-power-control algorithm is implemented to ensure the QoS. On the other hand, the output points of 3,4,8,9 and 14 show the low BER value but high delay value. So the cell-transfer-delay guaranteed algorithm is implemented. Besides all the above output points, the others represent both conditions of high BER and delay values. So automatic-frequency-hop or dynamic-power-control algorithm and the cell-transfer-delay guarantee algorithm is used simultaneously for those output.

Output Number	Solutions
19,24	Tolerable condition for both BER and CTD, no QoS guaranteed algorithm used.
20,21,22,23	Tolerable condition for CTD but not for BER, AFH / DPC algorithm is implemented to ensure QoS.
3,4,8,9,14	Tolerable condition for BER but not for CTD, CTD-guaranteed algorithm is implemented to ensure QoS.
0,1,2,5,6,7,10,11, 12,13,15,16,17,18	Non-tolerable condition for both BER and CTD, AFH / DPC and CTD-guaranteed algorithms are implemented to ensure QoS simultaneously.

Table 4. Output number vs. solutions

4.3.3.4 Dynamic-Power-Control Algorithm

It is possible that all the channels are occupied when the MU encounters a high BER. Under this scenario, the power of the BS transmitter should be dynamically adjusted by the dynamic-power-control algorithm immediately so that the signal strength received by the MU from the BS is much stronger than that received from the jammer. The degree of the power adjustment is related to the received BER level. For example, if the received BER is only slightly larger than the traffic contract, the power increment should be less. On the other hand, if the BER received by the MU is much larger than the traffic contract, the BS transmitter power needs to be increased more in order to guarantee quality of service. When the received BER conforms to the traffic contract, the algorithm will make the BS transmitter revert to its original power level. Table 5 shows the neural network output number and the corresponding change in power level.

Output Number	Power level adjustment
3,4,8,9,14,19,24	Don't change
2,7,12,13,17,18,22,23	Original power x 20
0,1,5,6,10,11, 15,16,20,21	Original power x 80

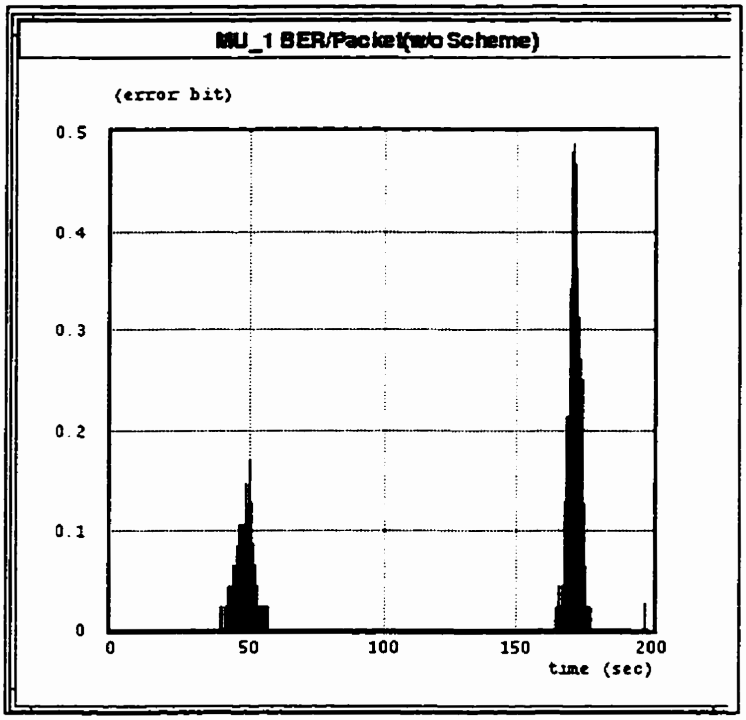
Table 5. Neural network output vs. power level adjustment

Figure 14 shows the BER-per-packet and average-BER received by MU_1 with and without the BER guarantee algorithm, and the resulting changes to the BS transmitter's power level. In Figure 14 (a) and (b) we see that without the QoS guarantee algorithm, the average BER received by the MU_1 is approximately 10^{-3}. The main interference to MU_1 occurs at approximately 50 seconds and 170 seconds. According to the above assumption, there is no available channel for the MU_1 to hop to. So during both of these inference time periods, the dynamic-power-control algorithm is used. The BS transmitter power is adjusted by this algorithm dynamically, depending on the received BER value.

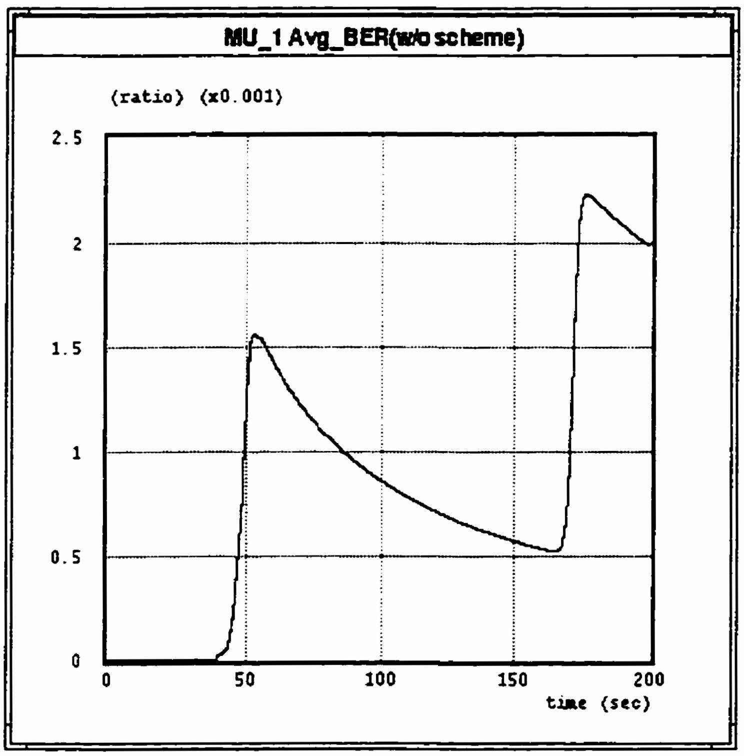
Figure 14 (c) shows the result of the power level which is dynamically controlled by the algorithm of BS_0_0. At about 40 seconds, the BER is not very high, so the

BS_0_0 transmitting power is adjusted from 10 mW to 200mW. The BER increases soon after and the BS_0_0 transmitting power is adjusted to 800 mW to eliminate the error bits. When the BER conforms to the traffic contract, the transmitting power is adjusted to its initial 10 mW automatically. At about 170 seconds the received BER is again very high and the BS_0_0 transmitting power is adjusted to 800 mW.

When the dynamic-power-control algorithm is working, the average BER received by the MU_1 decreases about 100 times from 10^{-3} to 10^{-5} (see Figure 14 (d) (e)). The DPC algorithm is again required here as there were no more available channels for the MU and BS to hop to.

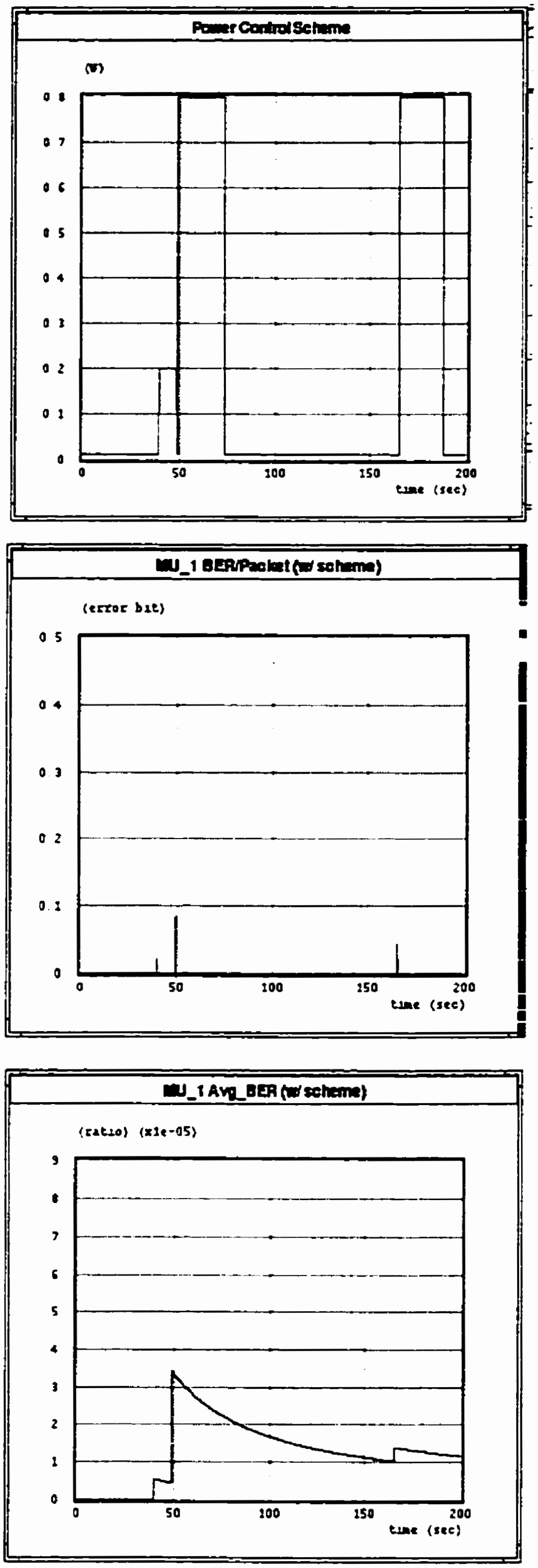


(a)



(b)

Figure 14 (a) – (b). BER guarantee algorithm in MU_1.



(c)
(d)
(e)

Figure 14 (c) – (e). BER guarantee algorithm in MU_1.

4.3.3.5 Automatic-Frequency-Hop Algorithm

According to the above assumptions, there were enough available channels for MU_2 to hop to during the simulation time. Figure 15 shows how the automatic-frequency-hop algorithm works. This algorithm is a very efficient one in broadband wireless LAN access networks.

Figure 15 (a) and (b) indicate that without the QoS guarantee algorithm, the average BER received by the MU_2 is approximately 10^{-3}. The main interference to MU_2 occurs at approximately 80 seconds, 100 seconds and 170 seconds. During these interference time periods, the automatic-frequency-hop algorithm is engaged. In Figure 15 (c), we see that, without the algorithm, the receiving frequency of MU_2 has only three kinds of frequency hops because it moves through three BS areas: BS_0_1 to BS_0_2 to BS_0_0 and returns back on the same trajectory. The frequency hops match each of the BS transmitting frequencies. When the detected BER is high, both simultaneously hop to another available channel. This explains why there are more frequency hops in Figure 15 (f). When the algorithm is used, the average BER decreases almost 1000 times from 10^{-3} to 10^{-6} (Figure 15 (d) (e)).

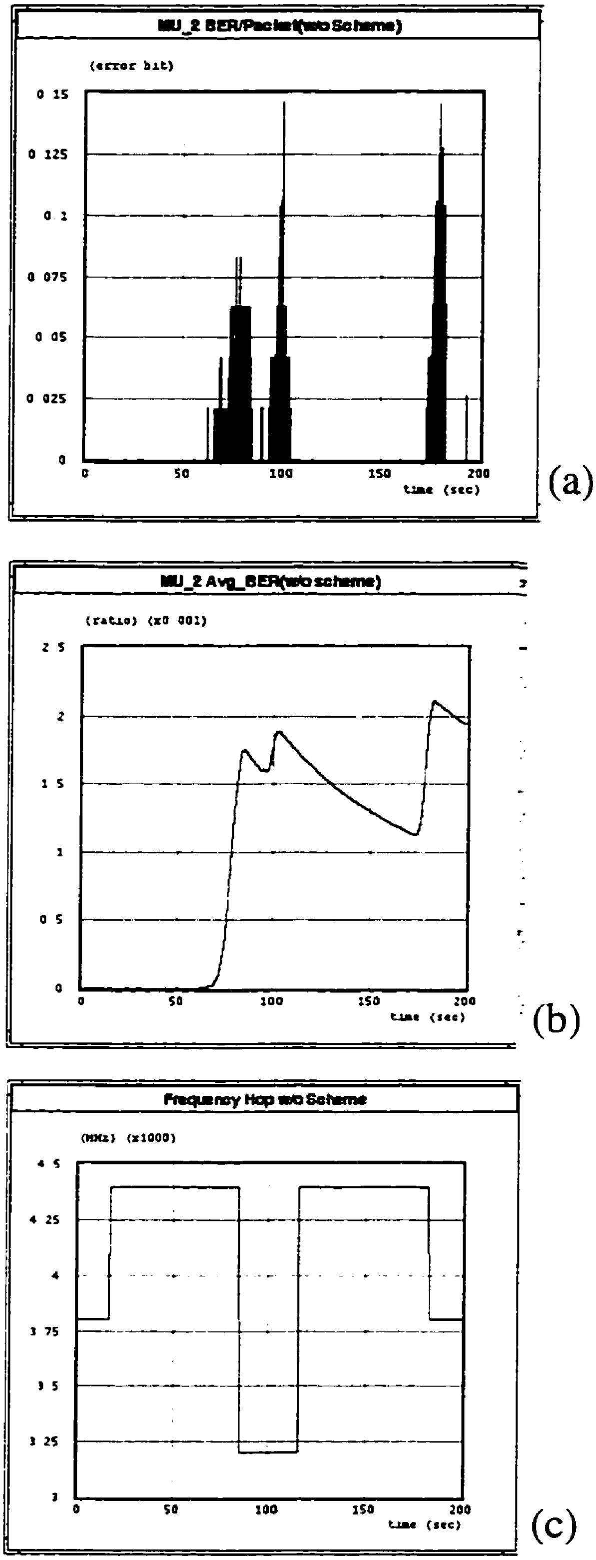
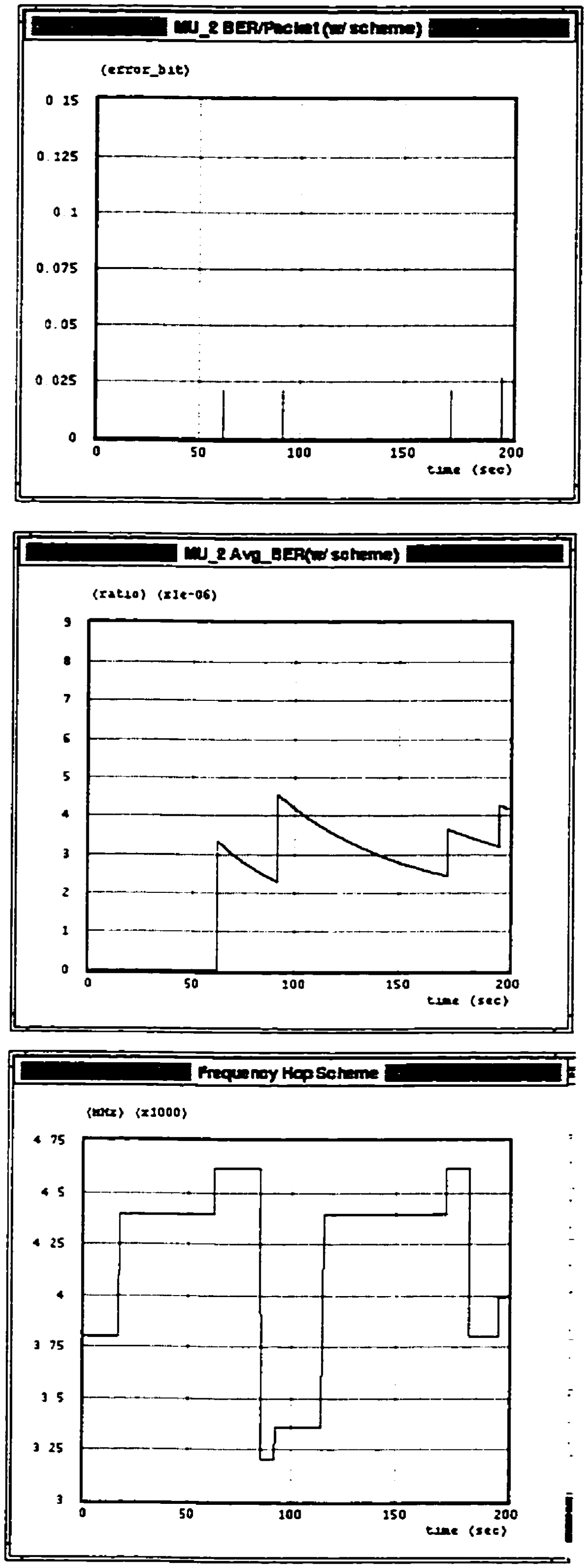


Figure 15 (a) – (c). BER guarantee algorithm in MU_2.



(d)

(e)

(f)

Figure 15 (d) – (f). BER guarantee algorithm in MU_2.

4.4 Summary

Broadband wireless mobility communications make personal communications more convenient. However, because of its principle characteristics - wireless and mobility - wireless access management is complicated.

In wireless ATM mobility management modeling, when the mobile unit moves from one cell to another, handoff occurs. The simulation results show that the error-free handoff ensures continuous communications and smooth data stream shift. In this model the error-free handoff algorithm is based on the received power measured by the mobile user as it transits the trajectory.

Since one goal of the network model was to have data follow mobile users, a replica algorithm is developed to guarantee the availability of the data at the next location during the handoff time period. The simulation results also show the replication algorithm working correctly.

Algorithms involved in QoS management were also discussed. Because the communication medium between mobile units and base stations is the air, it is prone to interference. If interference occurs, the quality of service will not conform to the traffic contract.

The purpose of the BER algorithm is to eliminate as many error bits as possible while ensuring QoS conformance. The BER guaranteed algorithm includes two parts - automatic-frequency-hop and dynamic-power-control - which can be implemented separately or simultaneously. If there is at least one idle channel available, the automatic-frequency-hop algorithm is implemented causing the mobile unit and base-station to hop to a non-interfered channel. The simulation verified the efficiency of the algorithm. If there are no unoccupied channels available, the dynamic-power-control algorithm is implemented to guarantee the quality of service. The BS transmitting power is adjusted dynamically according to the level of received BER. The higher the BER, the higher the transmitter power of the BS.

CHAPTER 5 NEURAL NETWORK BASED ATM QoS ESTIMATION

Chapter 5 presents the use of an artificial neural network to improve QoS by determining suitable service classes through the measurement of cell loss ratio and cell delay variation.

5.1 Training Data

5.1.1 Specified QoS Classes

The following specified QoS classes are defined by the ATM Forum. In the future, more QoS Classes may be defined for a given service class.

Specified QoS Class 1: supports a QoS that meets service class A (circuit emulation, constant bit rate video) performance requirements. This class should yield performance comparable to current digital private line performance.

Specified QoS Class 2 : supports a QoS that meets service class B (variable bit rate audio and video) performance requirements. This class is intended for packetized video and audio in teleconferencing and multimedia applications.

Specified QoS Class 3 : supports a QoS that meets service class C (connection - oriented data transfer) performance requirements. This class is intended for interoperation of connection-oriented protocols, such as Frame Relay.

Specified QoS Class 4 : supports a QoS that meets service class D (connectionless data transfer) performance requirements. This class is intended for inter-operation of connectionless protocols, such as IP or SMDS.

Figure 16 gives a concrete example of how the QoS parameters for cell loss rate for the CLP=0 (Cell Loss Priority) flow and CDV might be assigned for the four specified QoS classes. In the following section, we focus on one of the classes (Class 1) as the others can be treated in a similar way.

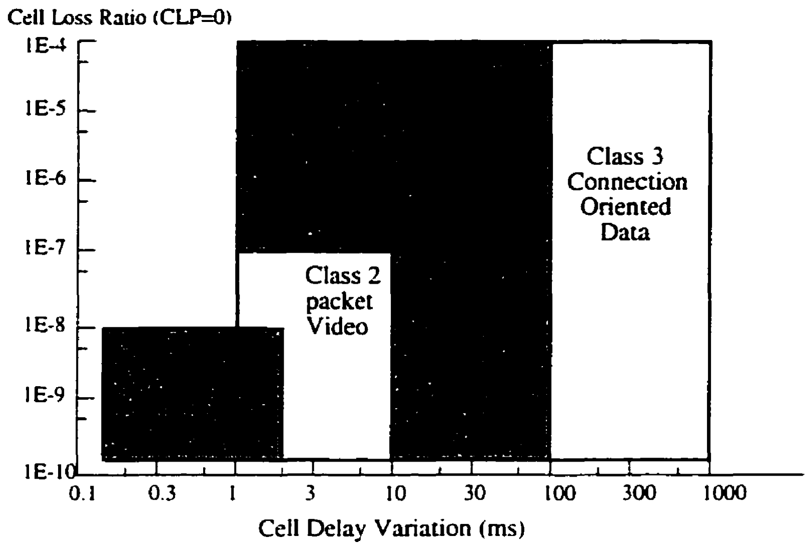


Figure 16. Example of QoS class value assignments (CLP=0)

5.1.2 Levels in QoS Classes

The levels in each class can be the same or different. The value of each level is decided by the service-providers depending on their network and customers' request. What is clear is that the smaller the values of CDV and CLR, the better the QoS will be. For instance, in Specified QoS Class 1, if CLR=1.1E-10 and CDV=0.1 ms (both are small), QoS should be in an excellent level. Alternatively, if CLR =1.1E-8 and

CDV=2.1 ms (both are large), QoS should be at a bad or unacceptable level. If CLR=1E-9 but CDV=2.1ms (one is small but the other one is large), it should also be in a bad level. Following the same principle, several levels in each QoS class can be estimated. In Table 6, seven levels of service are estimated for each of the four QoS classes.

QoS Level	Description
1	Excellent
2	Slight Alarm (from Excellent to Good)
3	Good
4	Regular Alarm (from Good to Fair)
5	Fair
6	Serious Alarm (from Fair to Bad)
7	Bad (will be disconnected)

Table 6. Rank of QoS levels

5.1.3 Normalization of the Training Data.

For the neural network used here, the network is trained using input data, in this case CLR and CDV and should be normalized into the range from 0 to 1. For instance, the CLR in QoS Class 1 is from 1E-10 to 1E-8. Any input data X in this class should be normalized as:

$$(X-1E-10)/(1E-8 - 1E-10) = [\ 0\ 1\ ].$$

With the same principle, the input data Y of CDV in QoS Class 1 should be normalized as:

$$(Y-0.1)/(2-0.1) = [\ 0\ 1\ ].$$

Using the above method, each data in each class can be normalized into the range of [0 1].

5.1.4 The Generation of the Training Data

The training data can be decided by the service-provider depending on their customers' service quality. Figure 17 shows the distribution of the training data assumed in this thesis. If CLR is larger than 1E-8 or CDV is larger than 2, the result of normalization will larger than 1. That is the reason why in Figure 17 some data are larger than 1 after normalization.

Table 7 shows the number of training data in each level. 324 training data were selected as QoS level 1, 78 data are as QoS level 2, and so forth. There were a total of 2320 training data for the seven QoS levels (Table 7).

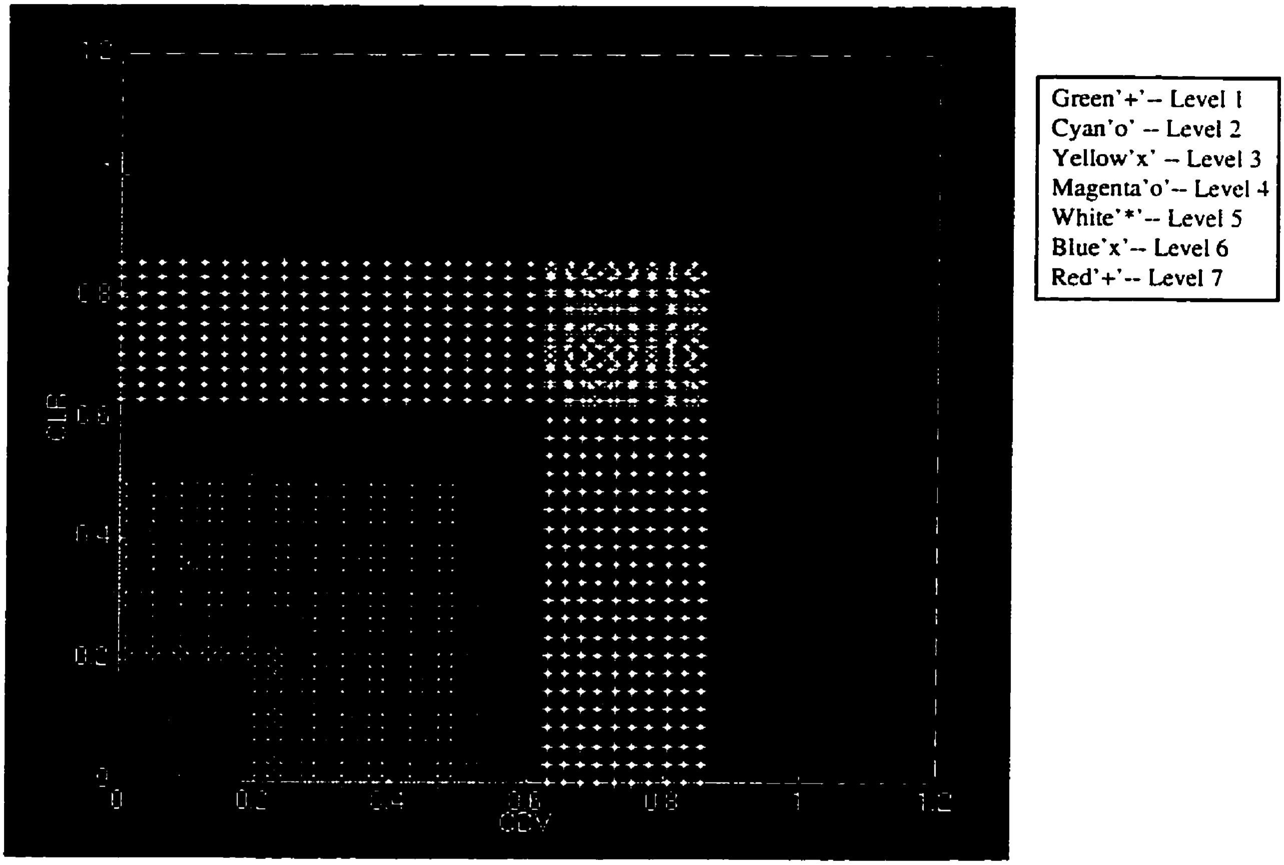


Figure 17. Training data

QoS	Number of training
1	32
2	78
3	70
4	12
5	58
6	21
7	30
	Total training data: 2320

Table 7. The number of training data for each QoS level

5.2 Testing Data

After training, the neural network is tested. The testing data is obtained randomly. Figure 18 shows the 877 testing data points we used. These testing data are generated by some functions (i.e., y=ax+b with different a and b).

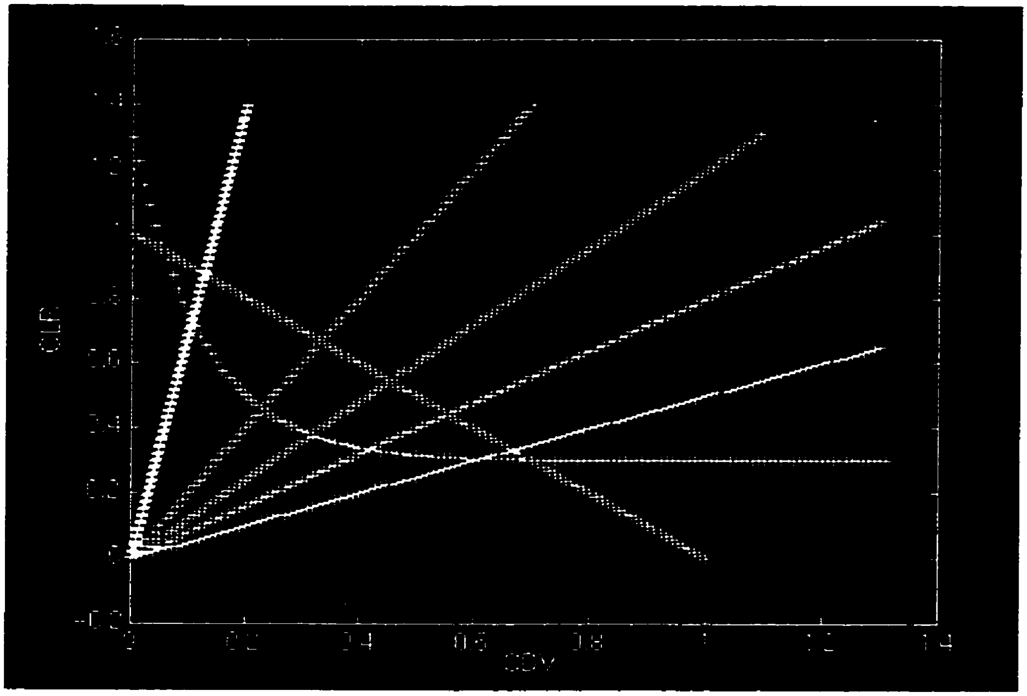


Figure 18. Testing data ('+' ---- random testing data)

5.3 Conclusion

5.3.1 Results & Discussion for LVQ Method

Because the LVQ is a supervised method, target vectors are needed. In the network architecture there are 7 outputs representing each level. They can be described as the following column vectors:

Level 1: [1;0;0;0;0;0;0]

Level 2: [0;1;0;0;0;0;0]

Level 3: [0;0;1;0;0;0;0]

Level 4: [0;0;0;1;0;0;0]

Level 5: [0;0;0;0;1;0;0]

Level 6: [0;0;0;0;0;1;0]

Level 7: [0;0;0;0;0;0;1]

In the architecture used for the QoS estimation, R=2, S2=7, and S1 is changeable. That means there are two inputs, several hidden neurons and seven outputs in the LVQ network. From the results we recognize that, although in theory more hidden neurons produce a better result, in practice it is costly in term of hardware cost.

Table 8 shows the training result of estimation of QoS levels using LVQ for different number of hidden neurons, training epochs and learning rate.

In the experiment, a network with 40 Hidden neurons, 10,000 epochs, and a learning rate=0.1, was seen to provide the best results for this experiment. Figure 19 plots the input vectors and the final weights of the competitive layer (The 'o' markers represent weight vectors for each level of competitive neurons, while the ' + ' markers represent the input vectors or training data).

Hidden Neurons	Training Epochs	Learning Rate	# Correct Data	# Incorrect Data	Correct Rate
40	5000	0.06	643	234	73.32%
40	7500	0.06	669	208	76.28%
40	10000	0.06	718	159	81.87%
40	15000	0.06	626	251	71.38%
40	20000	0.06	674	203	76.85%
40	10000	0.02	665	212	75.83%
40	10000	0.06	718	159	81.87%
40	**10000**	**0.1**	**720**	**157**	**82.10%**
40	10000	0.15	690	187	78.68%
40	10000	0.2	592	285	67.50%
20	10000	0.1	619	258	70.58%
30	10000	0.1	703	174	80.16%
40	**10000**	**0.1**	**720**	**157**	**82.10%**
50	10000	0.1	684	193	77.99%
60	10000	0.1	685	192	78.11%
100	10000	0.1	673	204	76.74%

Table 8. Testing result using LVQ

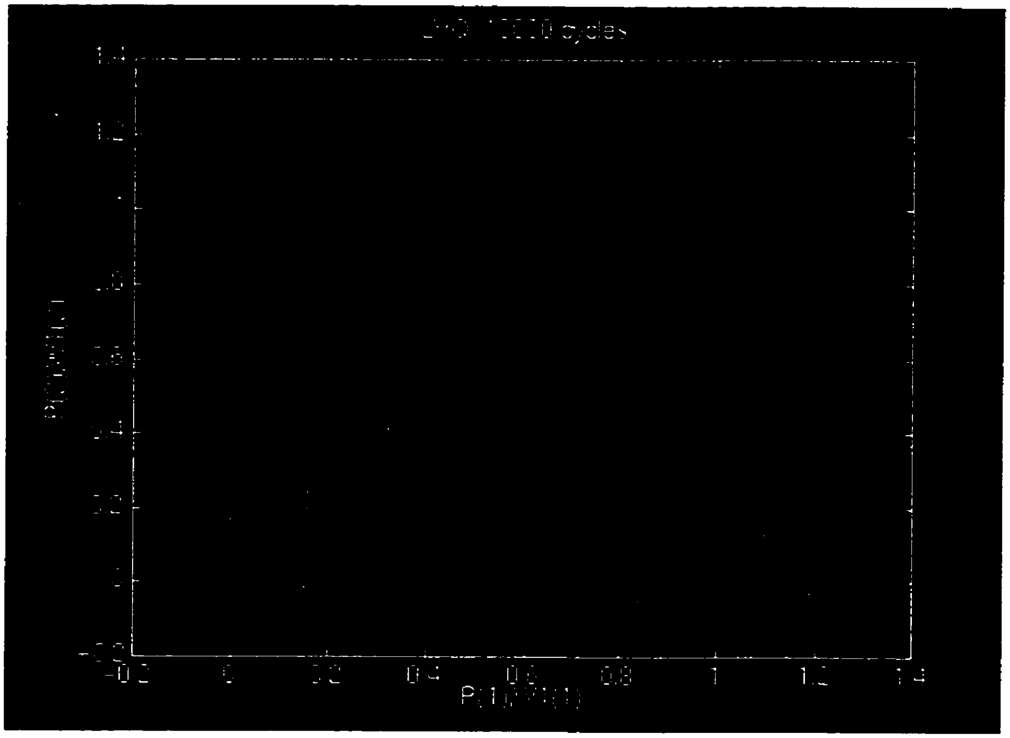


Figure 19. Ideal result using LVQ
('+' – training data, 'O' – weight vectors of competitive neurons)

5.3.2 Results and Discussion for SM Method

Figure 20 shows the plot of a self-organizing map after 5000 epochs, with 90 (10x9) neurons and an initial learning rate=1. Table 9 shows the classification of 90 outputs according to the 7 levels.

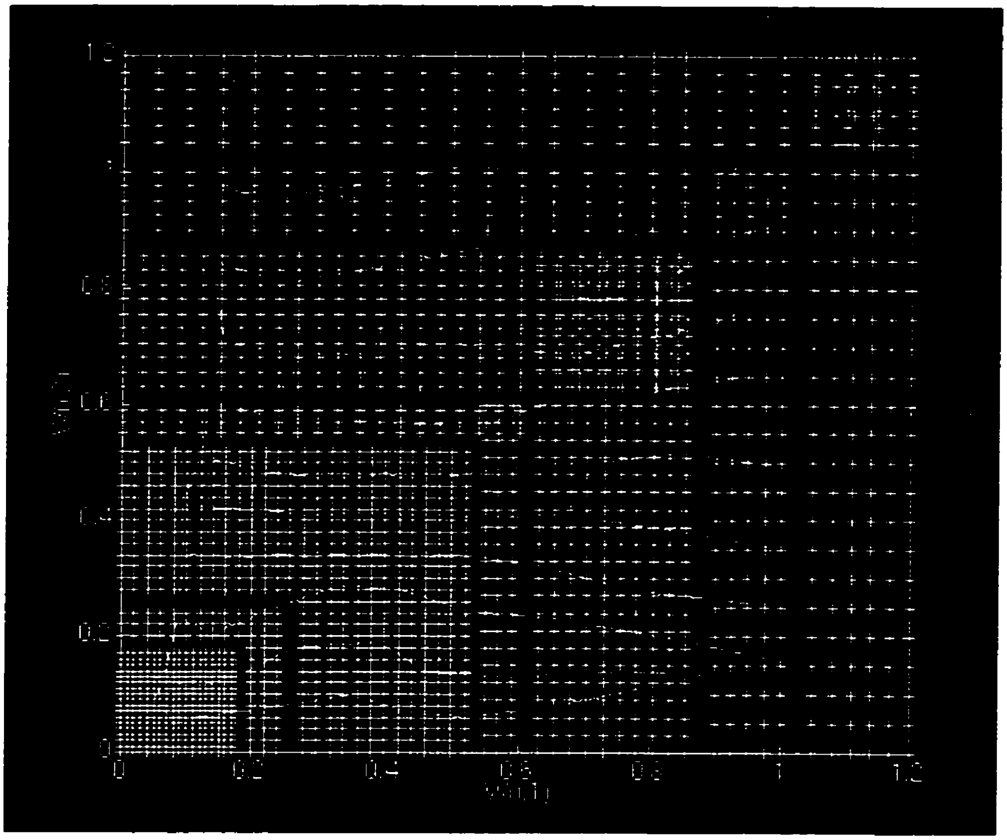


Figure 20. Ideal result using SM
('+' – Training Data, 'O' – Weight Vector of Competitive Neurons)

Rank of QoS Levels	The Output Neuron Number
1	79,80,81,88,89,90
2	70,71,72,78,87
3	41,42,43,44,45,50,51,52,53, 54,59,60,61,62,63,68,69,76, 77,85,86
4	32,33,34,35,36,40,49,58,67, 75,84
5	12,13,14,16,17,18,21,22,23, 24,25,26,27,29,30,31,39,48, 56,57,66,74,82,83
6	3,4,5,8,9,11,15,20,28,38,47, 55,65,73
7	1,2,6,7,10,19,37,46,64

Table 9. The classification of the outputs of SM

For example, if any one of the neurons 79, 80, 81, 88, 89 or 90 is on, the input vectors will be classified as Level 1. The possible results of the test are listed in Table 10.

Output neurons	Epochs	Correct Data	Incorrect Data	Correct Rate
90 (10x 9)	5000	677	200	77.19%
90 (10x 9)	8000	679	198	77.42%
90 (10x 9)	10000	679	198	77.42%

Table 10. Testing result using SM

A longer training time (number of epochs) would have produced a more even and accurate map, but for the purpose of the experiment the difference is negligible.

Compared to the best results of LVQ, SM obtains lower correct rate in this experiment.

5.3.3 Other Methods

Other methods, including Back Propagation and Radial Basis Functions, have been used for the problem of QoS estimation but they take longer to train. A 3-layer network with 20 hidden neurons has also been used (with an ε (error)=0.1 and training data N> W/ε, where N is the number of training data=2320, W is the number of total weights=180), but was not suitable for solving the problem presented in this thesis. From experiments it was found that using either Back Propagation or Radial Basis Functions it took nearly 100 hours to complete the training and it is unacceptable for this study.

5.3.4 Summary

In this chapter, we discussed the neural network based methods to estimate the QoS levels in the QoS classes for ATM management. If using target vectors as training data (supervised learning), Learning Vector Quantization (LVQ) was a feasible and reasonable method to estimate the QoS levels in QoS classes. On the other hand, if there are no target vectors as training data, a self-organizing map was a better method.

CHAPTER 6 VARIANCE FRACTAL DIMENSION BASED TRAFFIC ESTIMATION

Chapter 6 presents a variance fractal dimension traffic estimation method that may be useful as a descriptor for estimating traffic for use within the network management scheme.

6.1 Wireless ATM Simulation

In order to obtain the traffic parameters of the wireless ATM LAN, the network simulation tool OPtimized Network Engineering Tool (OPNET version 3.0.B DL_0/Radio) was again utilized. As previously mentioned, OPNET provides the capability to examine higher level and more complex behavior of wireless ATM networks.

The model used in the study here is a simple wireless ATM LAN as shown in Figure 21. The radius of the network is less than 100 m, like a micro-cell in a large wireless ATM LAN. A simple four-node topology consisting of one stationary transmitter node (WATM_TX), one stationary receiver node (WATM_RX) with a parameterized antenna, and two mobile jammers node (Jammer_1, Jammer_2) satisfied the objective of traffic data capture.

The transmission node WATM_TX, with 1Mbps bandwidth and 1W channel power, generates 64kbps data signals that were transmitted to the receiver node

WATM_RX with the signal modulation of BPSK. It uses an antenna with an isotropic pattern for a uniform gain in all directions.

Each of the mobile jammer nodes contains a dedicated transmitter that creates radio noise affecting the stationary receiver node. Each jammer (with 20 W channel power) moves along a trajectory taking it in and out of range of the receiver node.

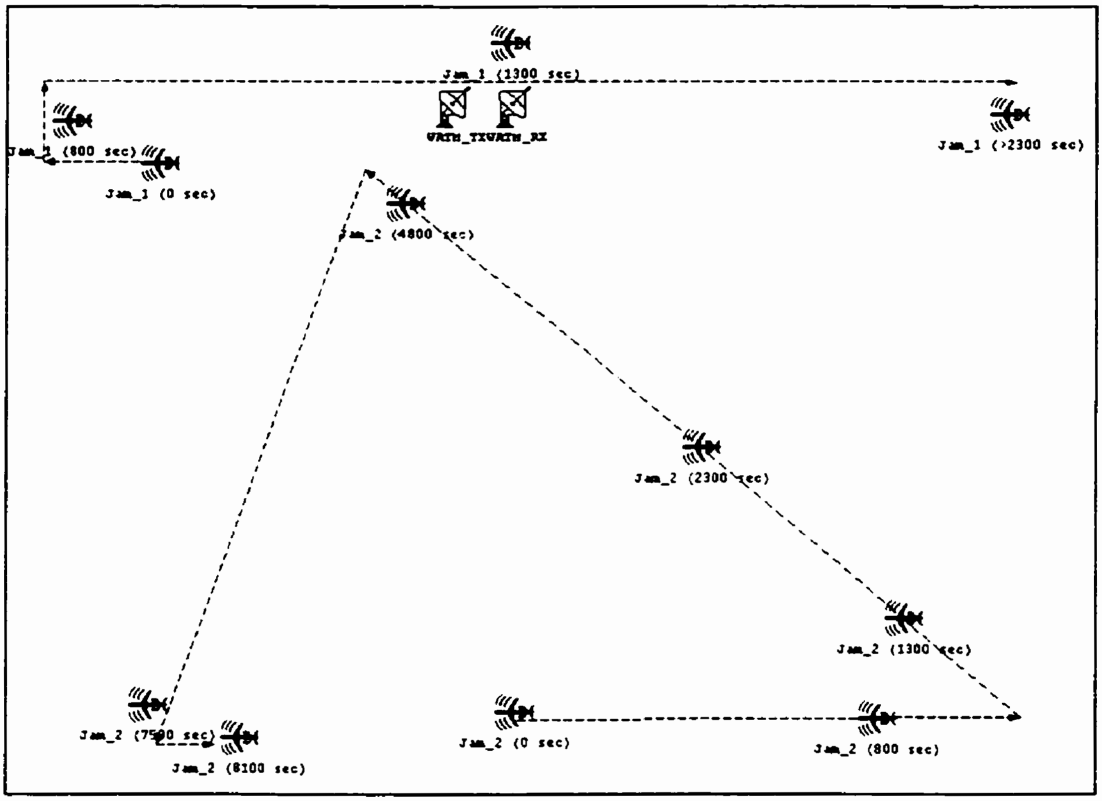


Figure 21. Wireless ATM LAN with dynamic trajectory of jammers.

The network jammer behaves much like the stationary transmitter node, with the channel power and signal modulation as the only characteristic differences. These differences cause the packets transmitted by the jammer node to appear as unrecognizable noise to the receiver node.

The receiver node WATM_RX measures the quality of the signal emitted by the stationary transmitter node WATM_TX. In this study, the received traffic parameter, bit errors per packet (bers_per_pk), is measured and tested as the input data for the variance fractal dimension computation. WATM_RX receives transmissions from both the WATM_TX and jammer nodes, and analyzes their respective strengths and traffic parameters.

All four nodes are operated at the radio frequency of 300MHz (UHF). The total simulation time in this study was 10300 seconds and the sampling frequency was 1Hz (once per second). The dynamic trajectories of two jammers (Jammer_1, Jammer_2) are also recorded in Figure 21.

6.2 Variance Fractal Dimension

Although a time series representing a chaotic or nonchaotic process can be analyzed by its power spectrum or power spectrum density, it can also be analyzed directly in real time by analyzing the spread of the increments in the signal amplitude or variance (σ^2). The variance fractal dimension is based on the properties of fBm. The calculations required are performed on samples separated by various time increments, making the variance fractal dimension a natural choice in the study of temporal signals.

The variance (σ^2) of a random process is defined by the second moment taken about the mean:

$$\sigma^2 = \mu_2 = E[\,(y - \mu_1)^2] = E\,(y^2) - \mu_1^2 \qquad (1)$$

where E denotes the expectation function, y is the random variance and μ_1 is the mean.

$B(t)$ is one-dimensional Brownian motion (Bm) in time (brown noise) with the initial condition $B(0)=0$. If $B(t)$ is assumed continuous in time t, then the variance, σ^2, of its amplitude increments over a time increment is related to the time increment according to the following power law:

$$\mathrm{Var}\,[\,(\Delta B)_{\Delta t}\,] \sim \Delta t^{2H} \qquad (2)$$

where $(\Delta B)_{\Delta t} = B(t_2)-B(t_1)$, $\Delta t = |\, t_2 - t_1 \,|$, and H is the Hurst exponent. Then the exponent H can be calculated from a log-log plot using:

$$H = \lim_{\Delta t \to 0} \{ \log [\, \mathrm{Var}(\Delta B)_{\Delta t} \,] / \log (\Delta t) \} / 2 \qquad (3)$$

Finally, for the embedding Euclidean dimension E, the variance dimension can be computed from:

$$D_\sigma = E + 1 - H \qquad (4)$$

For more details, see [PEI92],[Kin95] and [GRI96].

6.2.1 Program Design and Implementation

The variance dimension algorithm can be used to calculate the fractal dimension of a sequence of signal samples. The sequence contains N samples at times T (window size), given by

$$T = (\, N-1 \,)\, \delta t \qquad (5)$$

The time displacement used for the measurement of the increments (Δt) or the cumulative time displacement can be chosen as a multiple of the atomic time displacement; the multiples can be chosen in either a unit-distance or a dyadic selection as shown respectively by

$$\Delta t = n\, \delta t \,,\; n=2, 3, 4, \ldots, n_{max} \qquad (6)$$

or

$$\Delta t = n\, \delta t\ ,\ n=2^1, 2^2, 2^3, \ldots, n_{max} \qquad (7)$$

where n is the cumulative time displacement index and n_{max} is the largest displacement index desired.

Usually, in order to have a sufficiently large covering of the interval T by Δt, there should be more than 30 Δt in T (e.g., if T=512, the largest Δt should be 16) [KIN95].

In practice, at current k, the variance can be calculated by

$$\mathrm{Var}(\Delta B)_k=\Big[\sum_{j=1}^{N_k}(\Delta B)^2_{jk}-\Big(\sum_{j=1}^{N_k}(\Delta B)_{jk}\Big)^2 / N_k\Big] / (N_k-1) \qquad (8)$$

The two log values for the plot are calculated as

$$X_k = \log\, [n_k] \qquad (9a)$$

$$Y_k = \log\, [\ \mathrm{Var}\, (\Delta B)_k\ ] \qquad (9b)$$

After K stages for calculating the variance in T with different Δt, the slope s for the log-log plot can be computed by

$$s=\Big\{K\sum_{i=1}^{K}XiYi-\sum_{i=1}^{K}Xi\sum_{i=1}^{K}Yi\ \Big\} / \Big\{K\sum_{i=1}^{K}Xi^2-\Big(\sum_{i=1}^{K}Xi\Big)^2\Big\}$$

$$= \mathrm{Cov}(X,Y) / \mathrm{Var}(X) \qquad (10)$$

The Hurst exponent can be calculated from the slope s

$$H= s/2 \qquad (11)$$

Then the variance dimension can be obtained from (4).

In order to analyze the entire received signal or traffic parameters such as bers_per_pk, the variance fractal dimension trajectory should be formed. As shown in

Figure 22, a sequence of the localized samples calculated on windows starting at evenly spaced intervals form a fractal dimension trajectory of the original signal. The space can be changed according to the user. In this study, spaces were tested as 32, 64, 128, 256, and 512, where the window size was 512. If the space is less than window size, the scheme is regarded as overlapping. Otherwise, it is considered as a non-overlapping scheme [GRI96].

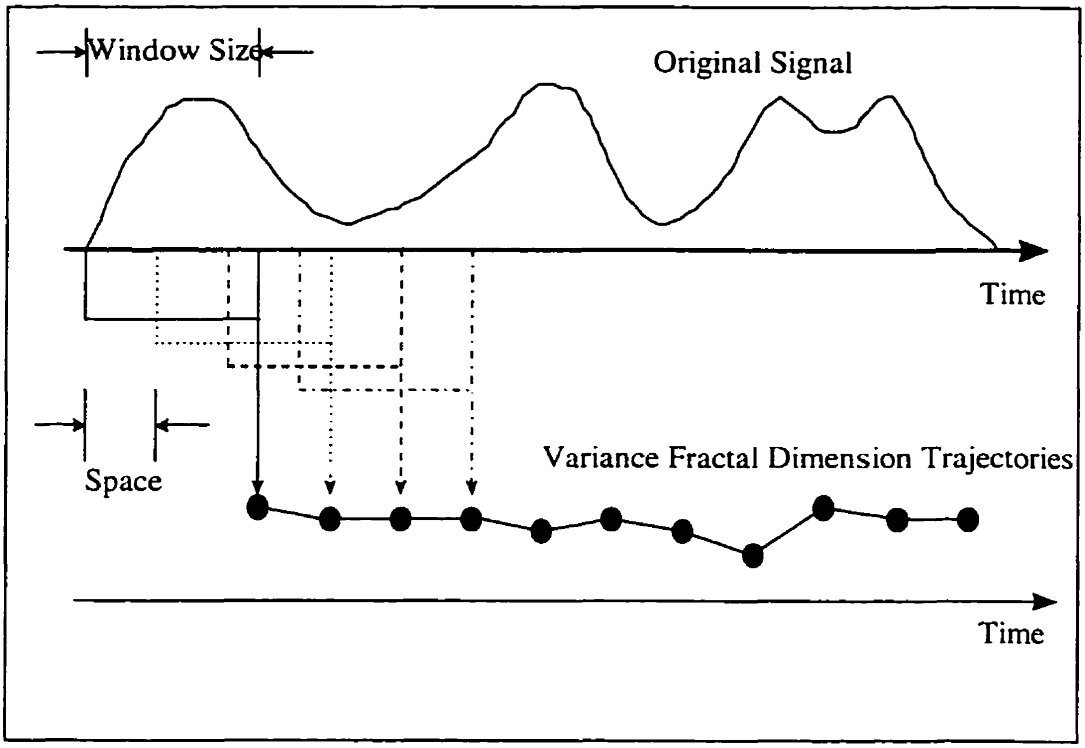


Figure 22. Variance fractal dimension trajectory

The spacing interval of the starting points for the window and the size of the window are important parameters to consider when calculating the dimension trajectory. The starting points of the windows can be spaced at intervals ranging from a single sample to the width of the window. When the spacing interval is small, many redundant calculations are made. When the spacing approaches the size of the window, critical transitions in the dimension may be missed. If the window size chosen is too small, the dimension may be very erratic because the window is smaller than the self-similarity or affinity features within the relative waveforms. If the window size chosen is too large, distinct features within different segments of the relative waveform may become blurred as the windows are larger than the segments.

6.3 Experimental Results and Analysis

Experiments were performed using the above system design constraints. The first set of experiments tested the accuracy of the variance fractal dimension algorithm by applying it to a set of fBm waveforms that were artificially generated to have a specific dimension. A second set of experiments was performed on a set of bers_per_pk parameters received by the radio receiver WATM_RX in the simulated wireless ATM LAN network.

6.3.1 Software Verification

As indicated, the test data used to verify the variance fractal dimension algorithm was generated by fBm to produce waveforms with a known fractal dimension. Small particles of solid matter suspended in a liquid can be seen under a microscope moving about in an irregular and erratic way. This is the so-called Brownian motion, caused by the random molecular impacts of the surrounding particles. The Brownian motion is not only important for describing physical systems such as suspended particles, but it also serves as the basis for fractal characterization in many other models. In simulation, the straightforward and most common way to produce a Brownian motion is *random midpoint displacement*. It has several advantages over the method of summing up white noise, the most important being its ability to generalize for several dimensions.

In this experiment, nine waveforms of fBm were generated to have fractal dimension from 1.1 to 1.9 spaced at equal intervals. Each waveform contained 32769 ($2^{15}+1$) points with 15 iterations. In the variance dimension calculation, the window size was 512 points, and the space size was 128 and 512 points for overlapping and non-overlapping methods respectively.

The Gaussian random numbers (G) used in generating the fBm list the conversion of the dice sum to an approximate normalized Gaussian random number. The Gaussian random numbers are generated using the following formula

$$G = (2/35)^{0.5} * (Y1+\ldots+Y6-21)$$

where Y1... Y6 are random integer between [1, 6].

The variance fractal dimension of each of these waveforms was calculated using both the overlapping and non-overlapping techniques. The results shown in Figure 23, indicate that there was no apparent difference between the two techniques. Although there was some error, the variance fractal dimension lines are nearly parallel to the expected dimension line. The error did not substantively affect the analysis.

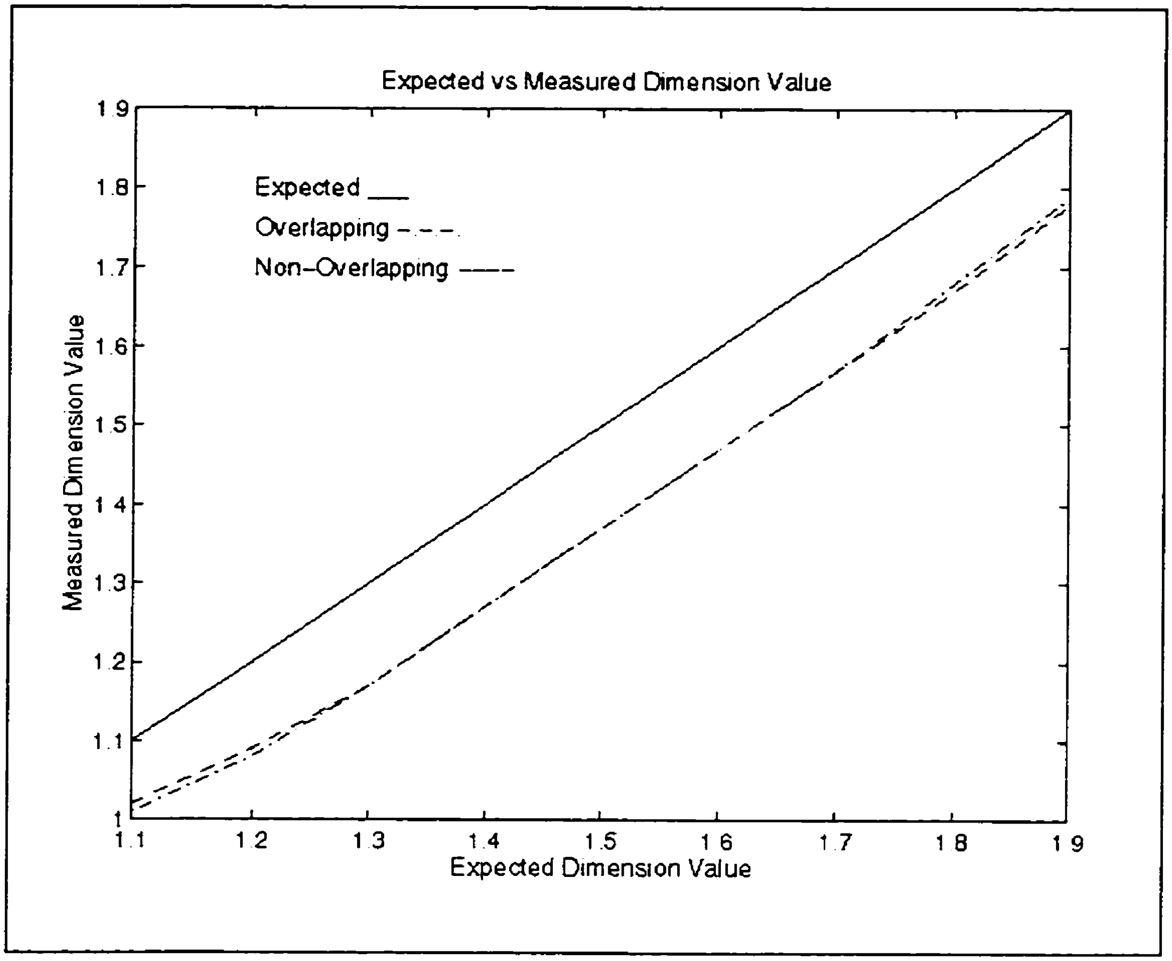


Figure 23. Expected vs. measured dimension value.

6.3.2 Wireless ATM Bers_per_pk

Recall the network configuration simulated in OPNET shown in Figure 21, the simulation time was set to 10300 seconds and the sampling frequency to 1 Hz (once every second). Figure 24 shows the traffic parameter, bit errors per packet (bers_per_pk), received by the radio receiver WATM_RX over the simulation time. The horizontal axis represents time in seconds while the vertical axis shows the number of bit errors within the packet after completing arrival at the receiver channel.

After simulation, a set of data sampled by time was generated. By saving these data in the same file as the original input data for the system described in Sec. 6.2.2, the variance fractal dimension trajectories can be obtained. Different spacing size and window size were experimented with to arrive at the most suitable parameters for traffic analysis.

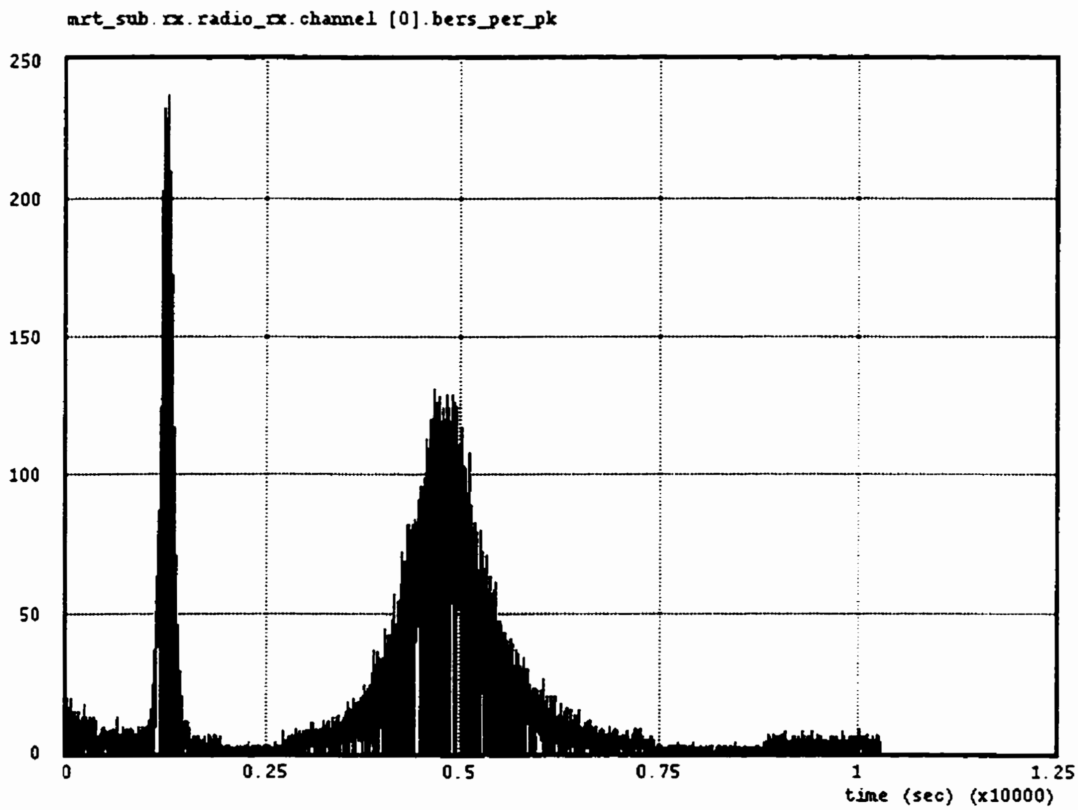


Figure 24. Bit errors per packet in simulation

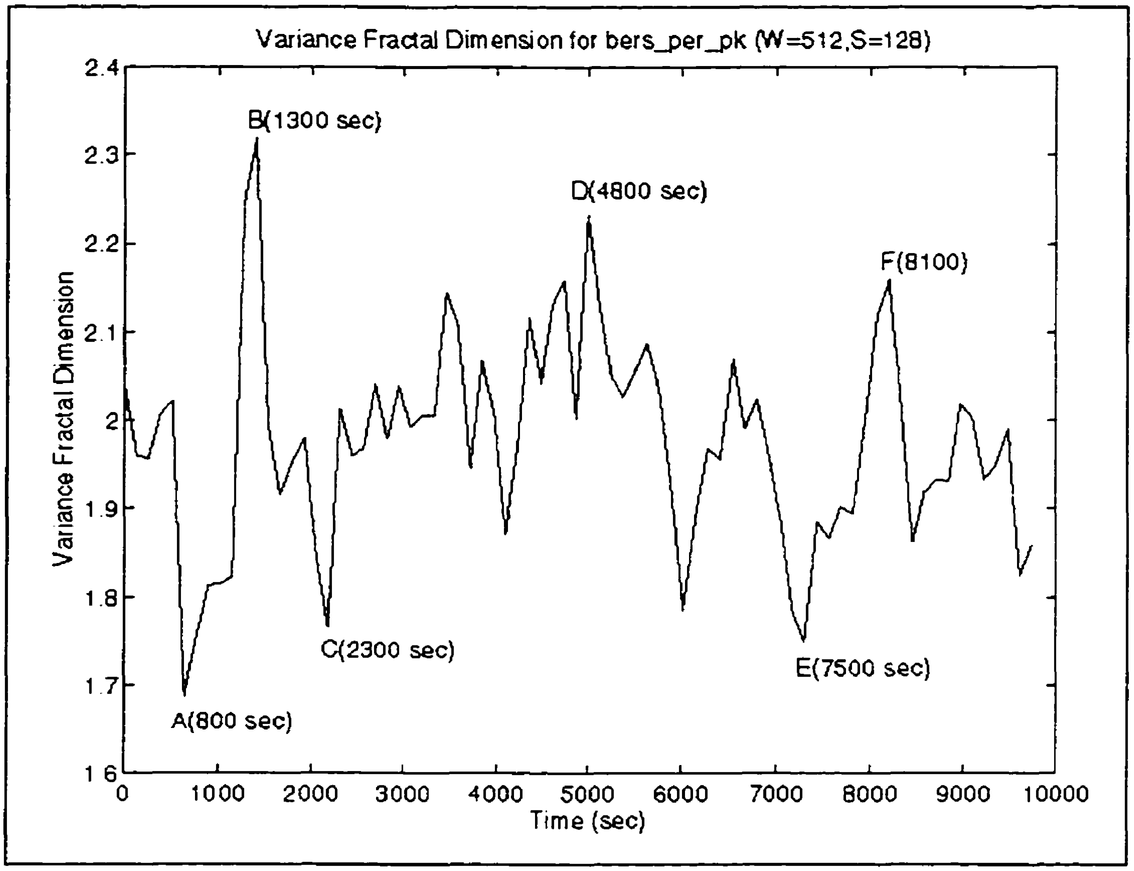


Figure 25. Variance fractal dimension measured with window_size=512, space size=128

From these results, it is clear that the spacing interval for the window and the size of the window are important parameters to consider when calculating the dimension trajectory. For example when the spacing interval is too small (32 seconds), many redundant calculations are made and the trajectory curve is not smooth. Alternatively, when the spacing approaches the size of the window (512 seconds), critical transitions in the dimension may be missed and the original waveform can not be characterized accurately and correctly. As previous mentioned, if the window size is chosen too small, the dimension is very erratic because the window is smaller than the self similarity or affinity features within the relative waveforms. If the window size is chosen to be too large, distinct features within different segments of the relative waveform are blurred together because the windows are larger than the segments. By changing the window size and spacing interval in many experiments, we conclude that the window size of 512 seconds and spacing interval of 128 seconds are suitable for characterizing the original bers_per_pk waveform, as shown in Figure 25.

Figure 25 shows 6 representative points from A to F. The variance fractal dimensions of points A, C, E, whose time axes range from 800 seconds to 2300 seconds and 7500 seconds respectively, range from 1.68 to 1.7 significantly lower than B, D, F, (whose time axes range from 1300 seconds to 4800 seconds and 8100 seconds respectively) which variance fractal dimensions between 2.16 to 2.34. Notice that the lower variance fractal dimension characterizes the lower bit errors in each received packet and the higher one represents the higher bit errors per packet. This means that at 800, 2300 and 7500 seconds, the number of bit errors in the received packet by the radio receiver are low; in other words, there is less noise interference and the air transmission condition is better. On the other hand, at 1300, 4800 and 8100 seconds, the noise interference is serious and the bit errors in each received packet are higher than under normal conditions.

Recall Figure 21, showing us the dynamic trajectories of two jammers (Jammer_1, Jammer_2) with time indicated below each icon. Each jammer generates noise at the same frequency of the radio receiver (WATM_RX). It is known that if a jammer moves closer to the receiver, more noise interference is produced at the radio receiver. When the network simulation runs, the two jammers move according to predetermined trajectories.

In this case, Jammer_1 was set to a trajectory time of 2300 seconds and Jammer_2 was set to a trajectory time of 8100 seconds.

During the first 800 seconds, the two jammers move away from the radio receiver WATM_RX, decreasing the noise interference and lowing the bit errors per packet, resulting in a variance fractal dimension of only 1.68 (point A in Figure 25).

From 800 to 1300 seconds, they move towards WATM_RX, increasing the noise interference, until Jammer_1 is nearest to the radio receiver. This is reflected in the large peak in bit errors per packet, shown in Figure 24, and the variance fractal dimension peak (2.34) shown in Figure 25 (point B).

From 1300 to 2300 seconds, Jammer_1 moves away from the radio receiver, coming to rest at the farthest location from it. But Jammer_2 continues to moves towards it. At 2300 seconds, noise interference is low, and the variance fractal dimension is only 1.76 (point C in Figure 25).

In the period 2300 seconds – 4800 seconds the interference from Jammer_1 remains fixed as it is at the end of its trajectory, while Jammer_2 continues to move, coming to the point nearest to WATM_RX. At 4800 seconds noise interference from Jammer_2 is again very high (shown as another high peak in Figure 24), and the variance fractal dimension close to 2.24 (Figure 25, D).

During the time interval 4800 seconds – 7500 seconds, Jammer_2 moves away from radio receiver, and the bit errors in each packet decrease gradually until Jammer_2 reaches the other farthest point from the receiver WATM_RX. At 7500 seconds the variance fractal dimension is as low as 1.75 (point E in Figure 25) indicating less noise interference.

Jammer_2 resumes its trajectory moving closer to WATM_RX and increases the noise interference. By the 8100 seconds point the variance fractal dimension is as high as 2.16 (point F in Figure 25).

From the above analysis, we conclude that the variance fractal dimension reflects the characteristics of the original traffic parameter waveform such as bers_per_pk. By analyzing the variance fractal dimension of the traffic parameters, we can set the boundary between the serious interference by noise and the normal condition according to the value that is set by the network provider according to different service qualities. Another important advantage of using variance fractal dimension is that it can be used for real-time computation. By analyzing the trend of the variance fractal dimension future traffic can be forecast. In practice, the results can be transmitted to the switch or network management which can act accordingly (e.g. delete the high bit error packet, send a re-transmission request to the transmitter or send a handoff request).

6.4 Summary

Transmitting ATM packets over wireless networks presents unique challenges due to the inherent nature of radio communications (i.e. interference, etc). Traffic analysis is, of course, very important and variance fractal dimension is a practical and suitable tool. It is calculated on small windows within the complete signal or received traffic parameters, and is based on the variance over selected time increments. By analyzing the dimension trajectory, it is possible to obtain some characteristic features of the original traffic parameters. In this study, the traffic parameter bers_per_pk was captured by a simulated wireless ATM LAN. A low variance fractal dimension indicates that there are low bit errors in each packet and it is in normal condition. On the other hand, a high variance fractal dimension means high bit errors in each packet and the condition of the transmission is poor, a result of external interference or other reasons. In this way we can set boundary between the severe noise interference and the normal condition. Furthermore, in real-time computation, the trend of the variance fractal dimension can be estimated and forecast. Based on this feedback information, the switch and the network management can act accordingly. Using the same variance fractal dimension principle, the method can also be applied to other traffic parameters in wireless ATM LAN such as bit_throughput and SNR (signal/noise ratio).

CHAPTER 7 CONCLUSIONS

7.1 Summary

Wireless personal communication services (PCS) and broadband networking for delivery of multimedia information represent two important areas in telecommunications. Given that ATM is now viewed as a universal base technology for broadband networks, it is reasonable to consider extension of standard ATM services into next-generation microcellular wireless and PCS scenarios.

In this study, focus was on the mobility management and quality of service (QoS) management in such a wireless ATM system. Several algorithms were investigated using OPNET. The first was the error-free handoff algorithm in an environment where the data follows the mobile user, which guarantees continuous communication. The second, the replica algorithm, ensures consistency of the moving data. Dynamic-power-control and automatic-frequency-hop algorithms were also investigated. When at least one channel is idle, automatic-frequency-hop algorithm will be implemented to avoid interference. If there are no available channels for hopping, the dynamic-power-control algorithm will be used to ensure the quality of service. Finally, variance fractal dimension method was investigated to estimate the traffic.

7.2 Contributions

This thesis makes several contributions to wireless ATM modeling, including:

1) Developing a user-defined wireless ATM model using OPNET. There is no default wireless ATM model in the OPNET. All the models, from network-level to process-level, are established independently by the author.
2) Providing a test-bed for wireless ATM algorithm exploration. Although only two algorithms were investigated (mobility management and QoS management), the model can be extended to other algorithms and protocols. In this sense, the model provides an open environment for wireless ATM research.
3) Investigating an error-free handoff algorithm with replica control. This algorithm is potentially efficient for mobility management in wireless ATM network.
4) Investigating several QoS management algorithms, including dynamic-power-control and automatic-frequency-hop algorithms.
5) Investigating a prototype for neural network-based network management, which provides a method for putting intelligence into wireless ATM network management.
6) Investigating a variance fractal dimension based upon traffic estimation within the wireless ATM model.

7.3 Future Work

There remains room for improving the model developed here. For example the model could be regarded as a test-bed for broadband wireless LAN. Additional algorithms, protocols and nodes could also be realized in this model. Future work consists of following:

- Develop a hybrid algorithm for QoS management. In this study two kinds of QoS guarantee algorithms, automatic-frequency-hop and dynamic-power-control were implemented separately in the MUs. Hybridizing them for each MU means MU if the received BER is high and there are enough channels for hopping, the automatic-frequency-hop algorithm will be implemented at first. If there are not enough channels, then the dynamic-power-control algorithm will be used immediately.
- Develop an ATM interface for a broadband wireless LAN. The wireline ATM model developed here can be connected to a broadband wireless LAN.
- Investigate MAC network protocols, such as FDMA, CDMA, and TDMA.
- Use a neural network to allocate the bandwidth more intelligently. Currently bandwidth allocation is a core problem in wireless ATM.

REFERENCES

[ADD95] Ronald G. Addie, Moshe Zukerman and Tim Neame, "Fractal Traffic: Measurements, Modeling and Performance Evaluation", IEEE Communication, 1995.

[BLA96(a)] Uyless Black, "*Mobile and Wireless Networks*", Prentice Hall PTR, 1996

[BLA96(b)] G. S. Blair, N. Davies, A. Friday and S. P. Wade, "Quality of service support in a mobile environment: an approach based on tuple spaces", Technical Report, Department of Computing, Lancaster University, 1996.

[CHE95] K. C. Chen and L. Lien, "Multi-media wireless local area networking", in Rec. Technology Summit at TELECOM 95, International Telecommunication Union, Geneva, 1995.

[CLA96] Martin P. Clark, "*ATM Networks Principles and Use*", John Wiley & Sons Ltd, 1996.

[COR95] D.E. Comer, J.C. Lin, V.F. Russo, " An Architecture For A Campus-Scale Wireless Mobile Internet". Purdue Technical Report CSD-TR 95-058, Department of Computer Sciences, Purdue University. West Lafayette, Indiana 47907, USA.. 1995

[ELM95] A. Elmagarmid, J. Jing, T. Furukawa, "Wireless Client/Server Computing for Personal Information Services and Applications", Purdue Technical Report. Department of Computer Sciences. Purdue University. West Lafayette, IN 47907 USA. 1995.

[FAE96] Tor Erlend Faegri, "Replication in Distributed Object Systems", http://www.dcs.gla.ac.uk/~faegri/ replication-summary.html

[FAI95] M. Faiz, A. Zaslavsky, "Database Replica Management Strategies in Multi-database Systems with Mobile Hosts", presented at the 6th International Hong Kong Computer Society Database Workshop: Database Reengineering and Interoperability, Hong Kong, March, 1995.

[GOO89] D. J. Goodman, R. A. Valenzuela, K.T. Gayliard, and B. Ramamurthi, "Packet reservation multiple access for local wireless communications", IEEE Trans. Comm., vol. 37, pp. 885-890, 1989.

[GRA96] J. Gray, P. Helland, P. O'Neil, and D. Shasha, "The Dangers of Replication and a Solution", SIGMOD'96 6/96 Montreal, Canada, 1996 ACM 0-89791-794-4/96/0006.

[GRI96] Warren S. Grieder, " Variance Fractal Dimension For Signal Feature Enhancement And Segmentation From Noise", Thesis of M.Sc., Dept. Electrical and Computer Eng., University of Manitoba, Winnipeg, Manitoba, Canada, September 1996.

[HSI97] T. Russell Hsing, Donald C. Cox, Li Fung Chang and Thierry Van Landegem, "Wireless ATM." IEEE Journal On Selected Areas In Communications, Vol. 15, No. 1, January 1997, pp. 1-3.

[HUT96] D. Hutchison, G. Coulson, A. Campbell and G. Blair, "Quality of Service Management in Distributed Systems", Technical Report. Computing Department, Lancaster Unversity, UK, 1996.

[KEU95] Young-Keun Park and Gyungho Lee, "Applications of Neural Networks in High-Speed Communication Networks", IEEE Communications Magazine, October 1995.

[KIN95] W. Kinsner, "Fractal and Chaos Engineering", Lecture Notes, Electrical and Computer Engineering, University of Manitoba, January 1995.

[KYA95] Othmar Kyas, "*ATM networks*", International Thomson Publishing, 1996

[McD95] David E. McDysan and Darren L. Spohn, "*ATM: Theory and Application*", McGraw-Hill, Inc, 1995.

[McT94] M.J. McTiffin, A.P. Hulbert, T.J. Ketseoglou, W. Heimsch, G. Crisp, "Mobile Access to an ATM network Using a CDMA Air Interface", IEEE Journal on Selected Areas in Communications, Vol. 12, No. 5, June 1994.

[MIL96] MIL 3 Company. "OPNET Manual", On-line Documentation, 1996.

[PEI92] H. -O. Peitgen, H. Jurgens, and D. Saupe, "*Chaos and Fractals*", New York: Springer-Verlag, 984pp., 1992.

[RAP96] Theodore S. Rappaport, "*Wireless Communications: Principles and Practice*", Prentice Hall PTR. 1996.

[RAY94] D. Raychaudhuri and D. Wilson, "ATM-Based Transport Architecture for Multiservices Wireless Personal Communication Networks", IEEE Journal On Selected Areas In Communications, vol 12, No 8, Oct. 1994, pp 1401-1413.

[RAY95] D. Raychaudhuri, R. Yuan, A. Iwata and H. Suzuki, "Rationale and Framework for Wireless ATM specification", ATM Forum/95-1646/PLEN, pp1-7.

[RAY96] D. Raychaudhuri, "Wireless ATM Networks: Architecture, System Design and Prototyping", IEEE Personal Communications Magazine, 1996.

[RAY97] D. Raychaudhuri, "Mobile ATM: A Generic, High-Performance Infrastructure Network for PCS, Cellular and Wireless Data", OEM Magazine, Jan 1997, pp: 1-6.

[SAL96] D. Salsbury, LT, USN, J. Thompson, LT(jg), USCD, " Modeling an Error-Free Handoff Scheme For a Wireless ATM Network", Technical Report, Naval Postgraduate School, Monterey, CA. 1996.

[SAN96] H.S. Sandhu, S. Zhou, "Cluster-Based File Replication in Large-Scale Distributed Systems", Proceedings of the 1996 ACM Sigmetrics and Performance Conference, Newport, Rhode Island, pp. 91-102.

[SCH96] G. Schussel, "Replication, The Next Generation of Distributed Database Technology", http://www.dci.com/speakers/archive/replica.html.

[SHE97] W. Sheng, A.J. Rueda, D.C. Blight. "Neural Network based0 ATM QoS Estimation", Proceeding, WESCANEX'97, May 1996, pp1-6.

[TAR95] Ahmed A. Tarraf, Ibrahim W. Habib, and Tarek N. Saadawi, "Intelligent Traffic Control for ATM Broadband Networks", IEEE Communications Magazine, October, 1995.

[YAO96] Zhonghui Yao, "ATM Network Models for Traffic Management", M. Sc. Thesis, Electrical and Computer Engineering, University of Manitoba. 1996.

[YUA96] R. Yuan, S. K. Biswas, L. J. French, J. Li and D. Raychaudhuri, " A Signaling and Control Architecture for Mobility Support in Wireless ATM Networks", Technical Report, NEC USA, C&C Research Laboratories, Princeton, New Jersey, 1996.

IMAGE EVALUATION
TEST TARGET (QA-3)

150mm

6"

APPLIED IMAGE . Inc
1653 East Main Street
Rochester, NY 14609 USA
Phone: 716/482-0300
Fax: 716/288-5989