ABSTRACT

SATURATION ROUTING FOR ASYNCHRONOUS TRANSFER MODE (ATM) NETWORKS

by
José Luis Uclés

The main objective of this thesis is to show that saturation routing, often in the past considered inefficient, can in fact be a viable approach to use in many important applications and services over an Asynchronous Transfer Mode (ATM) network. For other applications and services, a hybrid approach (one that partially uses saturation routing) is presented. First, the minimum effects of saturation routing are demonstrated by showing that the ratio, defined as f, of routing overhead cells over information cells is small even for large networks. Second, modeling and simulation and M/D/1 queuing analysis techniques are used to show that the overall effect on performance when using saturation routing is not significant over ATM networks. Then saturation routing ATM implementation is also provided, with important extensions to services such as multicast routing.

After an analytical comparison, in terms of routing overhead, is made between Saturation Routing and the currently proposed Private Network-Network Interface (PNNI) procedure for ATM routing made by the ATM forum. This comparison is made for networks of different sizes (343-node and 2401-node networks) and different number of hierarchical levels (3 and 4 levels of hierarchy). The results show that the higher the number of levels of hierarchy and the farthest (in terms of hierarchical levels) the source and the destination nodes are from each other, the more advantageous saturation routing becomes. Finally, a set of measures of performance for use by saturation routing (or any

routing algorithm), as metrics for routing path selection, is proposed. Among these measures, an innovative new measure of performance derived for measuring quality of service provided to Constant Bit Rate (CBR) users (e.g., such as voice and video users) called the Burst Voice Arrival Lag (BVAL) is described and derived.

SATURATION ROUTING FOR ASYNCHRONOUS TRANSFER MODE (ATM) NETWORKS

by
José Luis Uclés

A Dissertation
Submitted to the Faculty of
New Jersey Institute of Technology
In Partial Fulfillment of the Requirements for the Degree of
Doctor in Philosophy in Electrical Engineering

Department of Electrical and Computer Engineering

August 2001

APPROVAL PAGE

SATURATION ROUTING FOR ASYNCHRONOUS TRANSFER MODE (ATM) NETWORKS

Jose Luis Ucles

Dr. Constantine Manikopoulos, Chair of the Committee — Date
Associate Professor, Department of Electrical and Computer Engineering, NJIT

Dr. Ali Akansu, Committee Member — Date
Professor, Department of Electrical and Computer Engineering, NJIT

Dr. Jay Jorgenson, Committee Member — Date
Associate Professor, Department of Mathematics, CUNY

Dr. Symeon Papavassiliou, Committee Member — Date
Assistant Professor, Department of Electrical and Computer Engineering, NJIT

Dr. Sirin Tekinay, Committee Member — Date
Assistant Professor, Department of Electrical and Computer Engineering, NJIT

BIOGRAPHICAL SKETCH

Author: José Luis Uclés

Degree: Doctor of Philosophy

Date: August 2001

Undergraduate and Graduate Education:

- Doctor of Philosophy in Electrical Engineering
  New Jersey Institute of Technology, Newark, NJ, 2001

- Master of Science in Electronic Engineering
  Monmouth College, West Long Branch, NJ, 1986

- Bachelor of Science in Electronic Engineering
  Monmouth College, West Long Branch, NJ, 1984

Major: Electrical Engineering

Presentations and Publications:

C. Manikopoulos, J. Ucles
"Average Information Staleness (AIS) as a System Measure of Performance," Proceedings of the Third IEEE Symposium on Computers & Communications (ISCC'98), July 1998

C. Manikopoulos, J. Ucles
"Saturation Routing for Asynchronous Transfer Mode (ATM) Networks," IEEE Military Communications Conference (MILCOM 98), October 1998

J. Ucles, C. Manikopoulos
"Use of the Burst Voice Arrival Lag (BVAL) as a Measure of Performance That Detects Packet Burst Errors (PBER) for Voice over IP Systems," IEEE Military Communications Conference (MILCOM 99), October 1999

J. Ucles, C. Manikopoulos
"Analysis of the Overhead of Saturation Routing for ATM Networks", 4th World Multiconference on Circuits, Systems, Communications and Computers (CSCC 2000), July 2000.

J. Ucles, C. Manikopoulos

"Analysis of the Overhead of Saturation Routing for ATM Networks", in Proceedings, Systems and Control: Theory and Applications, N. Mastorakis (Ed.), World Scientific and Engineering Society Press, ISBN: 960-8052-11-4, 2001, pp. 90-96.

Z. Zhang, J. Li, C. Manikopoulos, J. Jorgenson, J. Ucles

"A Hierarchical Anomaly Network Intrusion Detection System Using Neural Network Classification", Advances in Neural Networks and Applications, N. Mastorakis (Ed.), World Scientific and Engineering Society Press, ISBN: 960-8052-26-2, 2001, pp. 333-338.

To my parents, who taught me about the existence of ***la luz*** *and the benefits of it.*

To my family, who made extensive sacrifices allowing me to pursue ***la luz****.*

To my coworkers, who in numerous occasions provided me with encouragement to finally reach ***la luz****.*

To my advisor, who always encouraged me and guided me to ***la luz****, regardless of how far away it appeared to be.*

ACKNOWLEDGMENT

I would like to express my greatest appreciation to Dr. Dinos Manikopoulos, who provided me with endless hours of guidance and discussion, in a positive and encouraging way. Special thanks to the committee members, Dr. Ali Akansu, Dr. Symeon Papavassiliou, and Dr. Sirin Tekinay for taking their time on reviewing this dissertation and providing me with endless advice. Also, special gratitude to Dr. Jay Jorgenson, who took time out of his busy schedule to become a member of this committee.

TABLE OF CONTENTS

TABLE OF CONTENTS
(Continued)

Chapter **Page**

LIST OF TABLES

LIST OF FIGURES

LIST OF FIGURES
(Continued)

Figure **Page**

CHAPTER 1

INTRODUCTION

Saturation Routing has always been discounted in the past as a highly inefficient routing mechanism for a packet type of network. However, in this thesis, it will be shown that although the overhead may be higher than other typical routing algorithms, the amount of overhead spent is offset by the savings produced when obtaining a more efficient path. That is, the path efficiencies gained in terms of throughput and delay will surpass the expense associated with the saturation routing process. This is especially true in networks, such as the Asynchronous Transfer Mode (ATM), which exhibit a low ratio of the routing effort over the amount of information exchanged. Then following illustrates research done in the literature that substantiates the use and advantages of using Saturation Routing.

Routing efficiency - For example, the Private Network-Network Interface Specification, Version 1.0 [1] indicates the following:

> "ATM is a connection oriented network technology. This means connections will be maintained for a long period of time. Inefficient routing will affect connections for as long as they remain open. It is critical that paths are selected carefully".

It is clear that the intent of the specification is to highlight the importance of selecting the best path in a network such as ATM – i.e., a network on which connections will stay open for a long time (presumably with a substantial amount of information being exchanged). One of the main advantages of the Saturation Routing algorithm is that it will attempt every path in the network. As a result, the path selection can optimize

network resources at a maximum level. In fact, Saturation Routing can also be adapted to provide multipath benefits that have been reported in the literature [6].

Routing Accuracy - The same PNNI specification specifies the use of a flooding mechanism to advertise the link information that is necessary to maintain the link state routing tables required to perform their routing calculations. They use a saturation routing scheme to exchange their most important pieces of information (i.e., the routing information that allows routing to occur).

Routing Simplicity – The PNNI specification provides the argument that a hierarchy needs to be created because maintaining information for every physical link and for every node in the network would create enormous overhead in larger networks. Saturation routing does not maintain that information at every node, simplifying in this way its implementation complexity. In [12], it has been forecasted that future routing protocols may result to "old" switching techniques, such as saturation routing, where transmission efficiency is traded for simplicity

Routing Overhead – The PNNI type of algorithms requires the advertisement of external addresses (which can be in the thousands), which create substantial overhead in these networks. By its design, Saturation Routing can eliminate the need for these advertisements. Only each node needs to know the external addresses that it can reach. The rest of the nodes will know about it, when they desire to open connections to those external addresses. It is also claimed in [5] that PNNI does not reduce information to aggregate information to accomplish scalability and efficiency in large ATM networks.

Routing Consistency – The PNNI specification indicates that routing loops occur when either switches implement different routing protocols or there is inconsistency in

routing databases among the switches (typically due to changes in topology that have not been fully propagated). By its nature, Saturation Routing is a loop-free algorithm.

Routing Protocol Combination – The PNNI specification allows for the use of multiple routing protocols. A hybrid approach is recommended in this thesis for those networks that have already implemented the PNNI routing algorithm. Indications of how to use routing information to conduct other type of services (such as broadcast) have been reported in the literature [8].

Routing Protocol Efficiency for Added Services – the literature already describes how flooding based algorithms can enhance services such as Multicast [21] – this is proposed in this thesis as well. It has been reported that multicasting has emerged as one of the most focused areas in networking [2]. Even combinations of multicast with Quality of Service (QOS) support have been provided in the literature [2]. Anycasting services (a feature described in Internet Protocol Version 6 (Ipv6)) and reported in the literature [18 and 23] can also be provided by Saturation Routing.

To support the claim of the benefits of the use of Saturation Routing, this thesis has been organized as follows. The rest of Section 1 is an introduction section that includes an overview of the ATM operation, overview of routing protocols, overview of PNNI (the recommended routing algorithm for current ATM networks), and an overview of Modeling and Simulation techniques used in ATM networks. Section 2 includes the following: description of the Saturation Routing Algorithm in an ATM environment, an analysis of the overhead incurred by the Saturation Routing, a comparison of overhead with the existing PNNI approach, and a proposed alternative hybrid routing approach.

Section 3 describes the measures of performance recommended for use by the algorithm. Among those, the Burst Voice Arrival Lag (BVAL) is introduced and recommended. Section 4 describes potential implementations of Saturation Routing for multicast services. Section 5 provides conclusions and recommendations for follow-on research. Sections 2 through 4 contain the original contributions of this dissertation.

1.1 ATM Operation

This section provides a description of Asynchronous Transfer Mode (ATM) technology and systems. ATM systems are based on ITU-T Recommendations and ATM Forum Standards. These standards apply to a layered reference model. ATM is an integrating telecommunications concept that enables all types of information from voice to data to video to be transported by common transmission and switching facilities. ATM networks use fixed size packets (called cells) to transfer information, regardless of the type of information. The constant size of the cells enables use of fast packet switching technologies. Standards for ATM were developed based on the availability of highly reliable and high bandwidth transmission facilities (such as fiber, cable, SONET). The following subsections describe the reference model and layered structure of ATM protocols. In addition, it identifies some of the principle defining standards. It also provides a description of ATM networks.

1.1.1 Reference Model

The ATM reference model protocol architecture is illustrated in Figure 1. The model is divided into several planes and layers.

Control plane

The control plane performs the call/connection control functions usually referred to as signaling. These functions include call set up, call supervision, and call release. The signaling protocols are included as *higher layers* in the reference model of Figure 1.1. All the layers below support signaling. Figure 1.2 shows the control plane layers in an ATM network. Note that in Figure 1.2 *higher layers* denotes those layers above the signaling layer.

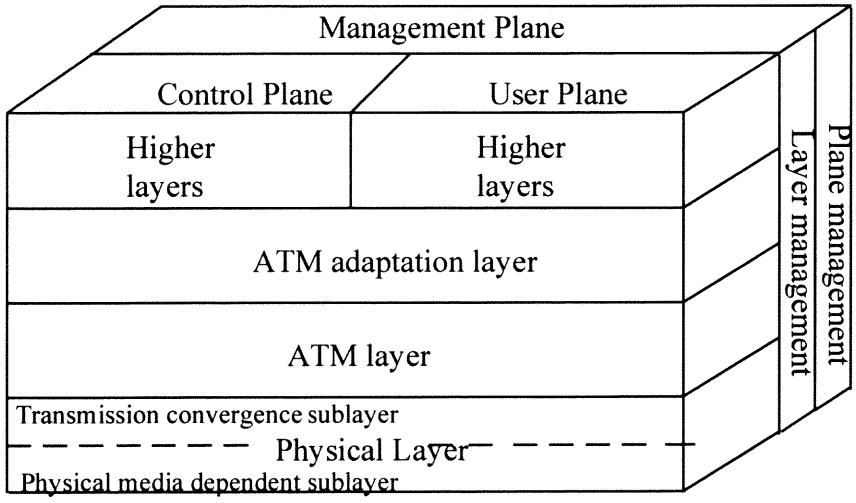

Figure 1.1 ATM Protocol Reference Model
Source: ITU-T Recommendation I.321, B-ISDN Protocol Reference Model and Its Applications, 1991.

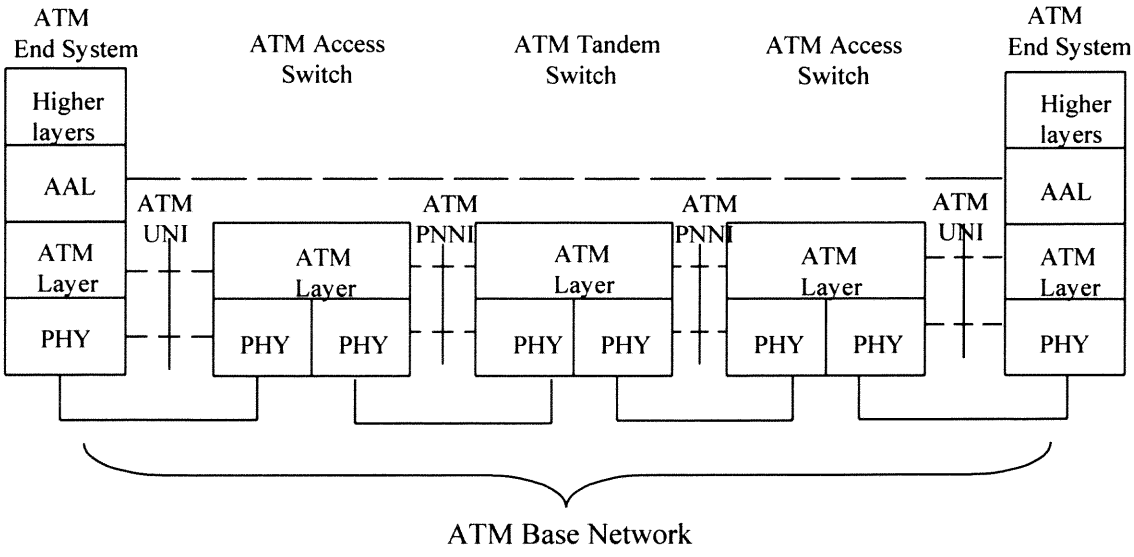

Figure 1.2 Control Plane and User Plane

User plane

The user plane provides the transfer of information via virtual connections set up by the control plane. The user connection uses only the physical layer and ATM layer within the network. The adaptation layer appears only in the ATM end systems. ATM end systems may be user terminals, or may be interfaces (gateways) to other types of networks, such as LANs or PBXs. Figure 1.2 shows the user plane layers in an ATM network.

Management plane

The management handles operation and maintenance (OAM) information flows, and provides coordination between and among all the planes.

1.1.2 Layered Structure

The ATM protocol layers are loosely coordinated with the OSI layered structure. The physical layer is layer 1. Layer 2 is generally considered to consist of the ATM layer and the ATM Adaptation layer. Within the network, layer 3 exists only in the control plane. In end systems, layer 3 exists in both the control and user planes. Refer to Figure 1.2.

ATM cell

In order to properly describe the functions of the various protocol layers, it is necessary to describe the ATM cell, which is shown if Figure 1.3. The cell consists of 5 header bytes and 48 data bytes.

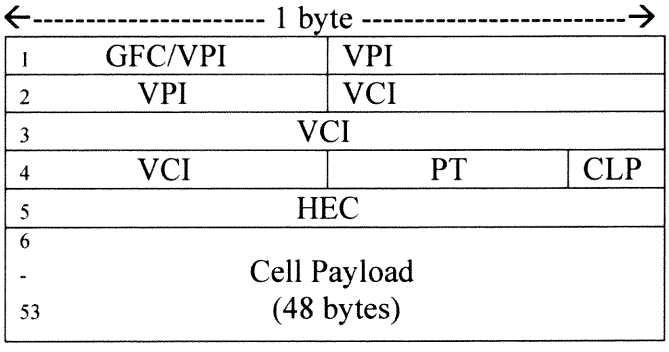

Figure 1.3 ATM Cell
Source: The ATM Forum Technical Committee, *ATM User-Network Interface (UNI) Specification*, Version 3.1, 1995.

At the UNI, the first 4 bits comprise the generic flow control (GFC) field. Within the network there is no GFC field and these 4 bits are used to extend the virtual path indicator (VPI) field. Together the VPI/VCI fields identify the virtual path and virtual channel for a given connection. All cells transferred over that connection contain the same VPI/VCI field contents. The 3-bit payload type (PT) field classifies the contents of the data field as user data, or for other purposes such as management or maintenance. The cell loss priority (CLP) bit indicates whether or not a cell may be discarded when the

network is congested. The 1-byte header error control (HEC) field performs error control on the header. It can correct single bit errors, or detect single and multiple errors (but not both). There is no error control provided for the user data. The defining standard is ITU-T Recommendation I.211.

Layer 3

There are several principle standards, which define the protocols and signal formats for ATM signaling. ITU-T Recommendation Q.2931 defines user to network (UNI) signaling between the user and the network. UNI signaling is non-symmetric since user signaling functions are different than network signaling functions. ATM Forum's af-pnni-0055.000 (PNNI) standard defines protocols to extend the signaling functions across the network from the access switch to the egress switch. PNNI signaling is symmetric, since both sides of the interface are network nodes, and in fact, PNNI signaling is very similar to the network side of UNI signaling.

ATM Adaptation Layer (AAL)

The AAL supports the transfer of signaling messages in the control plane, and user information in the user plane. Since an ATM cell can carry only 48 bytes of user data, a protocol is needed to adapt the protocol data unit (PDU) from the source of the data to the ATM cell. This is the function of the ATM adaptation layer. The AAL consists of two sublayers known as the common part (CP) and the service specific convergence sublayer (SSCS). The CP is further divided into the common part convergence sublayer (CPCS) and the segmentation and reassembly (SAR) sublayer. The CPCS functions include the delineation and transparency of user information, and CPCS PDU error detection. The SAR sublayer handles the segmentation of user information (PDUs) into fixed size

segments for insertion into cells, and the reassembly of cell payloads into CPCS PDUs. The SSCS provides the service specific functions of the particular AAL, and may be null. Several types of AAL protocols are required to handle the different types of user services, whether they are constant bit rate, or variable bit rate. These are described below.

AAL 1

AAL 1 supports constant bit rate (CBR) service such as voice or video circuit emulation where a timing relationship is required between source and sink. It provides a 47-byte payload, with a 1-byte payload header to support timing and sequence integrity. Adaptive clock recovery methods are not subject to standardization, since vendors may use different methods without causing incompatibility. The defining standard is ANSI T1.630.

AAL 2

AAL 2 is not clearly defined.

AAL 3/4

AAL 3/4 supports variable bit rate (VBR) services. It supports both assured and non-assured operations. For non-assured operations, the SSCS may be null. An error discard option allows corrupted PDUs to be delivered to the user. The protocol accepts variable length PDUs up to 65,535 bytes, and segments them into cells. Each cell carries a 44-byte payload, plus a 2-byte header and a 2-byte trailer. The payload header and trailer provide protection against misordering of cells, and a 10-bit cyclic redundancy check (CRC) for cell error detection. AAL 3/4 also provides an optional multiplexing of multiple connections. The defining standard is ANSI T1.629. For assured operations a

SSCS (not included in ANSI T1.629) is required to provide retransmission of erroneous cells.

AAL 5

AAL 5 supports variable bit rate (VBR) services. It supports both assured and non-assured operations. For non-assured operations, the SSCS may be null. An error discard option allows corrupted PDUs to be delivered to the user. The protocol accepts variable length PDUs up to 65,535 bytes, and segments them into cells. No error protection is provided in the cell payload, and each cell carries a 48-byte data payload. A 32-bit CRC is provided in the CPCS PDU for error detection. The defining standard is ITU-T Recommendation I.363. For assured operations a SSCS (not included in ITU-T Recommendation I.363) is required to provide retransmission of erroneous CPCS PDUs.

Signaling AAL (SAAL)

The SAAL conveys signaling information across the UNI and the PNNI. The SSCS is divided into the service specific coordination function (SSCF) and the service specific connection-oriented protocol (SSCOP). The SSCF maps the services of SSCOP to the layer 3 entity. SSCOP provides assured service for the signaling PDUs. SSCOP uses the services of the CP protocol, which in this case is AAL 5. The defining standards are ITU-T Recommendation Q. 2130 for SSCF and ITU-T recommendation Q.2110 for SSCOP.

ATM layer

The ATM layer provides connection-oriented sequence preserving service to the layers above by assigned virtual connection identifiers to each link of a connection when required, and releasing them when no longer needed. Signaling and user information is

carried on separate ATM layer virtual connections. The ATM layer provides its users several types of communications services, such as unidirectional point-to-multipoint communications, and bi-directional asymmetrical point-to-point communications. The ATM layer supports the control, user, and management planes. The defining standard is ATM Forum's af-uni-0010.002, Section 3, except that the approved standard ANSI T1.627 replaces the referenced letter ballot, T1 LB310.

An important function of the ATM layer is Traffic and Congestion Control. The following can cause congestion at the ATM layer: (1) ineffective call admission control, (2) unpredictable fluctuation of traffic flows, and (3) fault conditions within the network. Traffic at the UNI must conform to a traffic contract, which consists of a Connection Traffic Descriptor and a QoS class for each direction of an ATM layer connection. Cells that are non-compliant with the traffic contract may have their CLP bit set to lower priority, and may be subsequently discarded by the network. The defining standard is ITU-T Recommendation I.371 as modified by af-uni-0010.002.

Physical layer

The physical layer provides transmission services to the ATM layer. The physical layer consists of two sublayers, the transmission convergence (TC) sublayer, and the physical media dependent (PMD) sublayer. The TC sublayer performs all functions necessary to transform a flow of cells from the ATM layer to a flow of bits that can be transmitted and received over a physical medium. If the physical medium is synchronous, idle cells may be inserted. Common functions of the TC sublayer include header error checking, cell rate decoupling, and cell delineation. The defining standard is ITU-T Recommendation I.432. The PMD sublayer includes only physical media dependent functions, including

line coding, bit timing, and bit transmission and reception over the physical medium. Many phasic layer standards have been developed to accommodate different bit rates and physical media. Table 1.1 summarizes some of the physical layer standards.

1.1.3 Network Structure

The relationship between the ATM UNI, PNNI, and private and public networks is illustrated in Figure 1.4. The ATM UNI serves as the interface between user terminals and an ATM network, and between ATM networks and the gateway to non-ATM networks, i.e., LANs, IP routers, or PBXs. Also, since a private network switch is viewed as a terminal by a public network, the ATM UNI also serves as the interface between private networks and public networks. PNNI serves not only as the interface between switches within a private network, but also between switches of different private networks.

Table 1.1 Physical Layer Standards Specified by the ATM Forum

	Bit Rate	Media	Standard/Specification
STS-12c	622.08 Mbps	Optical	TC&PMD:af-phy-0046.000
STS-3c	155.52 Mbps	Optical	TC&PMD:af-uni-0010.002
		UTP-5	TC: af-uni-0010.002 PMD:af-phy-0015.000
		UPT-3	TC: af-uni-0010.0002 PMD: af-phy-0047.000
		STP	TC: af-uni-0010.002 PMD: af-phy-0015.000
		MMP	TC: ANSI TI..646 PMD: af-phy-0062.000
Fiber Channel	155.52 Mbps	Optical	TC&PMD:af-uni-0010.002
		STP	TC&PMD: af-uni-0010.002
FDDI	100 Mbps	Optical	TC&PMD: af-uni-0010.002
STS-1	51.84 Mbps	UTP-3	TC&PMD: af-uni-0018.000
DS3	44.736 Mbps	Coax	TC&PMD: af-phy-0054.000
25.6	25.6 Mbps	UTP/STP	TC&PMD: af-phy-0040.000
E-1	2.048 Mpbs	TP/Coax	TC&PMD: af-phy-0064.000
DS-1	1.544 Mbps	TP	TC&PMD: af-phy-0016.000

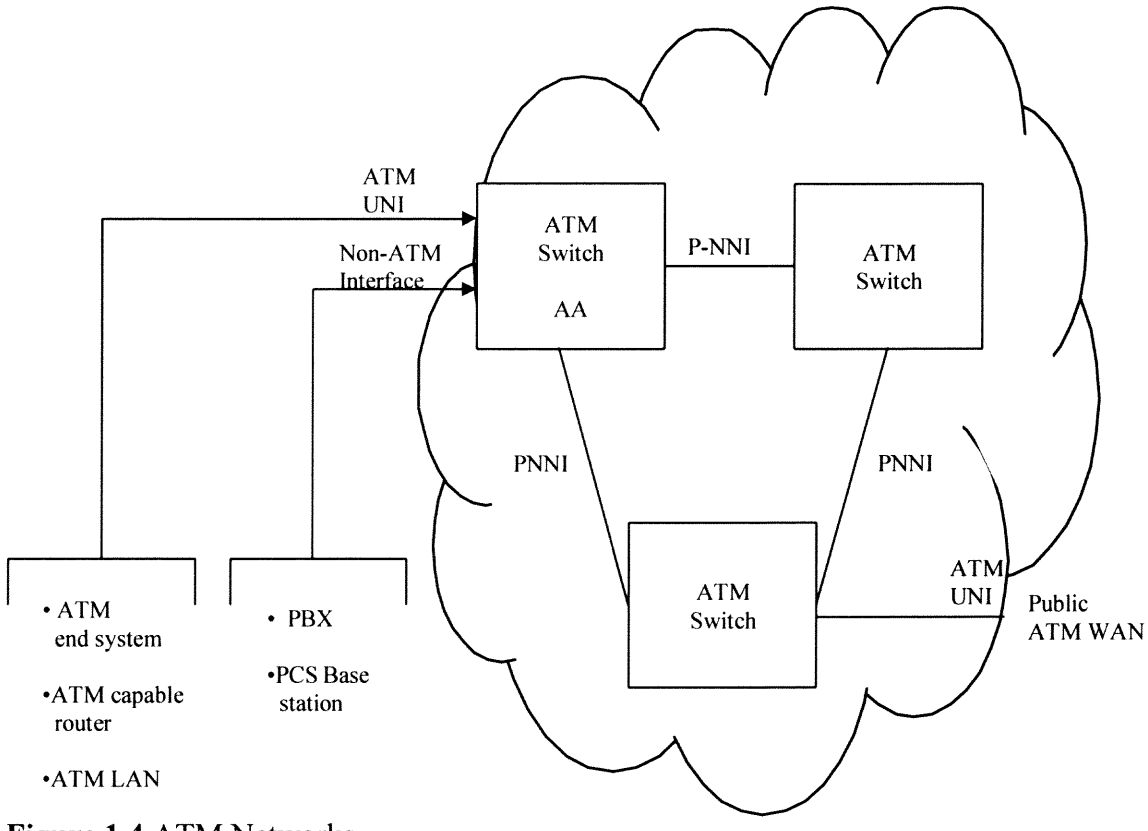

Figure 1.4 ATM Networks

1.2 Routing Techniques

The main goal of routing is to provide a path - an efficient one is desirable -between a source and an intended destination (or set of destinations, in the case of a multicast routing attempt). There is a variety of routing attributes and/or features of a vast number of routing algorithms described in the literature. A list attempting to describe most of those attributes is included below. This is followed by a description of some of the better known routing algorithms that are in use out there either in the telephone networks or commercial packet networks for wired infrastructure (such as the routing protocols in the Internet). It is not the intent of this section to provide an exhaustive list of all types of

routing protocols, but provide the highlights of the ones that are most relevant to the discussion in here.

1.2.1 Routing Attributes

The following provides a list of attributes/factors/issues that are typically used when describing different routing protocols:

Adaptability

It is desirable for routing algorithms to adapt to new conditions in the network. If a particular path does not exist because of failures in the network or link saturation, then it is desirable that the routing algorithm provides an alternative to it. From this perspective, routing algorithms can be classified as:

- Static - they never change their routing decisions (i.e., a simple table provides information as to where the routing effort should be directed to),
- Quasi-Static - they only change their routing decisions based on hard failures, either a node or a link failure, but not in the presence of saturation of a link,
- Dynamic (they attempt to optimize their routing based on the resources available in the links.

Stability

For the static routing algorithms, stability is not an issue (i.e., they are stable by design). However, for the non-static routing algorithms, stability is a desirable feature. Stability is a measure as to how the network approaches saturation. As the load increases in the network and approaches the maximum theoretical value, the routing algorithm should allow for the network to provide service to a constant number of users without

degradation, while using all of the network resources. Among the notorious examples of routing that have not shown stability is the original ARPANET routing approach. In this approach, as the network started reaching capacity, the routing engine started switching from the most-congested routes to the least-congested routes. However, the least-congested routes became the most-congested routes, after an update was received (i.e., the network resources are not used uniformly. This effect, combined with a high frequency of routing updates, created a phenomenon called route flapping (also called a fire-hose effect). Routes were recalculated frequently and the effect on the network was such that half of the network was highly utilized while the other half was not. The amount of capacity used to keep up with routing updates left the network with no more capacity for transporting more packets.

Network overhead

The amount of network overhead, as described in the previous attribute, also becomes an important factor of routing algorithms. Obviously, it is desirable that the overhead required for maintaining a routing protocol is minimum, or at least not significant. However, there is a tradeoff that this thesis will attempt to capture, in general the less overhead a routing protocol requires, the less efficient it becomes. In a less-strict definition, the amount of overhead generated by a protocol is measured by the amount of capacity taken by the messages used by the protocol to exchange routing information. However, in a more strict definition, overhead of a given routing algorithm could also add the amount of extra information sent on a network beyond those required by the most optimum route. A typical example of that is the use of Protocol Independent Multicast – Dense Mode (PIM-DM). In this routing algorithm, the information to maintain the

routing multicast distribution trees is minimized. However, to reach that point, the protocol dictates that all of the information sent to a multicast destination gets flooded out all valid interfaces. This obviously creates a significant amount of packet duplication that should be counted as overhead for a given routing algorithm.

Efficiency

It refers to the amount of network resources utilized to satisfy the requirement of transporting information from source to destination. The most common way of measuring efficiency of a routing protocol is based on the shortest path algorithm. Assuming that one is interested in selecting the minimum-hop route, the most efficient routing algorithm is the one that uses the shortest path between source and destination (on which each link hop has a cost of one). Efficiency can also become more complicated, and one may be interested on minimizing delay between source and destination. Then, the measurement is dependent not only in the number of links associated with the path, but the capacity available on those links to satisfy the demand and the processing power of the nodes associated between the source and the destination.

Memory/processing power

It refers to the amount of memory and processing power consumed by the nodes executing the routing. Memory became an issue in the commercial Internet, prior to address aggregation. That is because the amount of memory required by routers to route to each single destination in the Internet started growing. Updates to the routing techniques that allow address aggregation, which is used in storage of routing tables, provided the necessary relief in Internet type of networks. Similarly, the amount of

processing power needed to determine routes (to include execution of the routing algorithm) can also become an important issue.

Complexity/ease of implementation

Complexity and ease of implementation become an important factor in comparing routing algorithms. In general, the simpler the algorithm is, the more appealing is usage because of its impact on other measures such as: processing power, network overhead. That is because, typically, simpler algorithms require less processing power, less memory, and less network overhead.

Robustness

This attribute is related to the adaptability factor. A robust routing algorithm will find a path, assuming one exists, regardless of how severed a network becomes. For example, the saturation routing algorithm is well known for its robustness property – that is if there is an existent path, it will find it.

Consistency

A consistent routing algorithm is the one that provides a consistent path between source and destination (that is it will not create routing loops because of routing database inconsistencies).

Optimality

Optimality refers to how optimum the path is selected between source and destination. Saturation routing could become an optimum routing algorithm.

Routing Decision Place

This refers to the location of updates for routing occurs. There are two different types of updating routing tables. These are Centralized, Distributed, and Source. In the

centralized case, all of the routing updates are done in one central location and then they are either distributed or accessed by each individual node, as they require the routing information. Whereas in the distributed case, all of the routing updates are triggered by a change in any of the nodes' view of the network. The changes of view by a node, causes routing updates to be performed based on updates received by the other nodes. One of the concerns with distributed routing is the lack of consistency. As updates occur through the network, they could take a finite amount of time to propagate through the network and create routing loops (e.g., the counting to infinity problem referred to in the RIP protocol is a classical example of this problem). In the case of having the decision place done at the source node, the source defines the path to be taken between itself and the intended destination.

Type of routing effort

Under this category, routing efforts have been divided into the following: (1) Routing for a Circuit, (2) Routing for a Virtual Circuit, and (3) Routing for Datagrams. In the two first cases, the routing effort is to establish either a circuit or a virtual circuit. Presumably, the circuit is first routed using the routing protocol. Resources are checked to ensure that the circuit or virtual circuit can be supported. After the circuit is setup an exchange of data occurs between source and destination (and possibly vice-versa), using the circuit established (and therefore not requiring any routing efforts). Whereas, in the Routing for Datagrams case, data to be exchanged between source and destination is broken up into smaller packets (called Datagrams) and are routed individually. In this case, the path between source and destination does not remain constant during the exchange. As a result, each individual datagram requires a routing effort

Source route specifications

In order to avoid routing loops, some routing protocols either dictate or allowed the source node to specify the complete path to be taken by the packets between source and destination. In some cases, the specification may be strict, that is the routing path selected by the source must be followed and has been fully defined (i.e., includes all nodes involved in the routing). In other cases, the specification may be loose (i.e., not all of the nodes involved in the routing are included). An example of this type of routing is the one described in the Private Network to Node Interface (PNNI) for ATM networks. In this case, the source will include as much detail as the source is aware of. The intermediate points between not well defined routes will be defined by those intermediate points that have the full routing knowledge.

1.2.2 Routing Protocols for Telephone Networks

Most of the protocols defined for telephone networks are to select a circuit between source and destination. This circuit (virtual or real) is to be maintained for the whole duration of the conversation between the source and destination. The objective of the routing protocol becomes to select a trunk group to establish communications between the source and destination switches. Most of the telephone networks at their highest level of hierarchy are of the fully connected type (i.e., each switch has a connection with every other switch at that level in the hierarchy). As a result, the first option for most of these routing protocols is to take the direct route (i.e., the trunk group that connects directly the source and destination switches), if it exists and it is available. If that direct route becomes totally congested (i.e., there is no more trunks available for a connection), then that is the time at which a routing decision will have to be made. A description of the

Dynamic Alternate Routing (DAR) algorithm used in the British Telecom public switched Network is included below. Other algorithms such as the Dynamic NonHierarchical Routing (DNHR) used in the US are also classic examples of the routing algorithms used in telephone networks.

DAR algorithm

The DAR algorithm will take the direct path when trunks are available in that link. Let's assume that the 2 nodes labeled as S and D in the Figure 1.5 are to be connected. The most direct path will be selected assuming that is available. Otherwise, a previously selected node (which will be different for each source-destination pair and may change as a function of time) O_1 (called tandem node) will be used to make the connection a 2-hop connection. If the number of trunks available in any of the links associated with the alternate path (i.e., S-O_1 and O_1-D links) is less than a configurable parameter called trunk-reservation, the call will be blocked and the identification of the tandem node will be changed (possibly in a random fashion). Note how flow and congestion control (i.e., the decision of blocking or not blocking a call) becomes intimately related with the routing scheme.

DAR can be extended to multihomed networks on which there are two levels of hierarchy – one is the access network and the other one is the backbone (or also called core) network. In this case, the core is fully connected while the access network has 2 connections to the core (i.e., the primary and the secondary connection) to what is known as connections to parent nodes. In this case, DAR could be modified to always attempt direct connections between the primary parent of the source switch with either the primary or secondary parent of the destination. Or alternatively, between the secondary

parent of the source with either the primary or secondary parent of the destination. In this case, there will be four direct alternatives, prior to invoking the use of the alternate nondirect path by DAR.

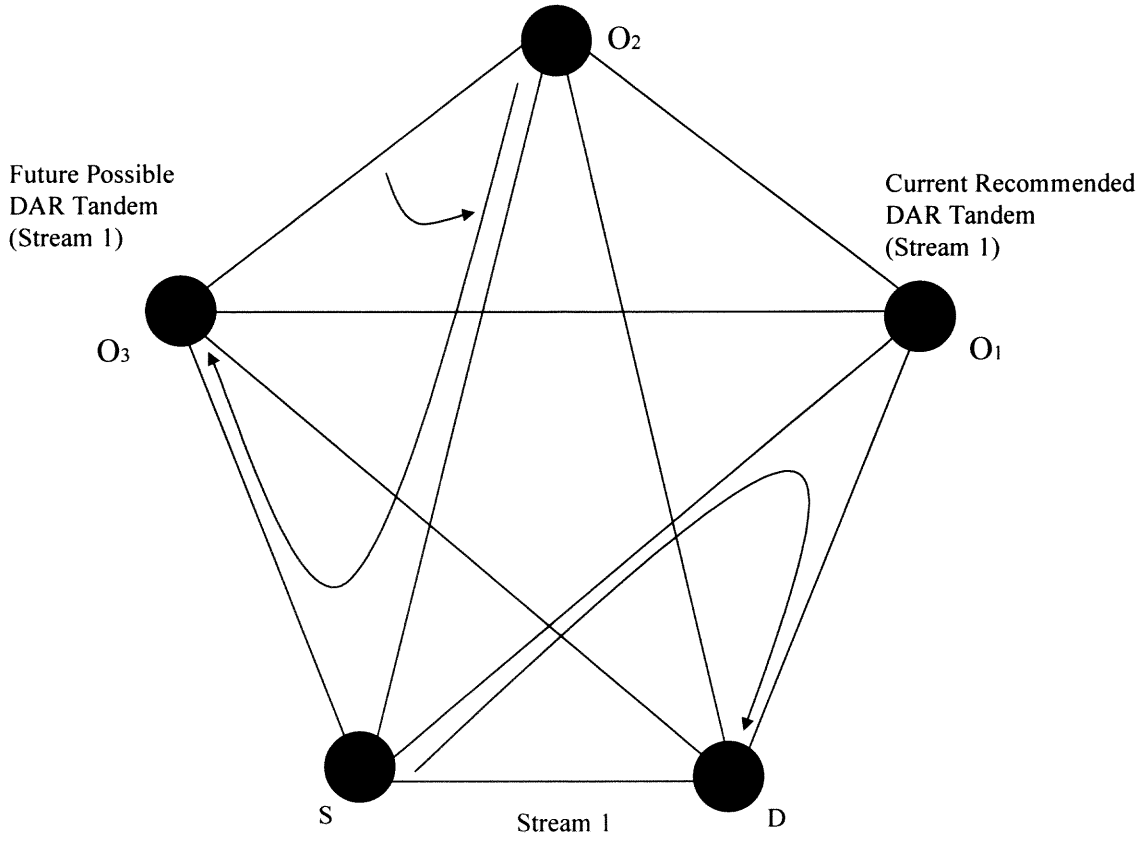

Figure 1.5 DAR Algorithm Illustration
Source: Martha E. Steenstrup, ed., *Routing in Communications Networks*, Englewood Cliffs, NJ: Prentice Hall, 1995.

1.2.3 Routing Protocols for Packet Switched Networks

A large variety of routing protocols are in use in today's networks. The explosion in the Internet has fueled a lot of research associated with the development of packet switched routing algorithms. Packet switching networks can use either Datagram technology or virtual circuit technology. Currently, the Internet uses a Datagram approach for routing between source and destination (i.e., routing decisions are done independently as each

packet arrives at each node). The Internet Protocol (IP) allows for source routing (in a strict or loose sense), but this option is almost never invoked, if at all. On contrast to IP, native ATM uses virtual circuit technology. A review of 3 different routing algorithms will be presented. The Routing Information Protocol (RIP) (this is of the Distance Vector family of routing protocols), the Open Shortest Path First (OSPF) networks (this is of the Link State family of routing protocols). The Private Network to Node Interface (PNNI) for ATM networks (also of the Link State family of routing protocols) will be discussed in the next subsection. This variety of routing protocols will show the different types of routing available for packet networks. Because of its importance to the topic, emphasis will be placed on the description of the PNNI routing algorithm.

Routing Information Protocol (RIP)

RIP is a protocol that was extensively used in the Internet for routing within a Routing domain (i.e., a set of routers that fall under the same network administrator or owner). It consists of a set of routers updating their view of the network (expressed in terms of reachable nodes and the distance required to reach those nodes) based on their local knowledge. In general the RIP routing algorithm could work with an arbitrary cost associated with each link (and that is how is being described in here). However, when used in the Internet that cost is just the number of links.

In essence, each router calculates the distance that it perceives between itself and all of its neighbor routers. This calculation is made and exchanged with all of the neighbors. As the neighbors receive this information, they update "their view" of the network and propagate that back to their neighbors. Because with the information received (assuming that all of the neighbors' information has been received), they now

can not only reach their neighbors, but they can also reach their neighbors' neighbors. This algorithm is recursively repeated by every router in the network, until there is no more updates to be transmitted. Once steady state is reached, routers will refresh information after a refresh period has expired, or if a router's view of the network changes (i.e., if a link or a node fails). It has been shown that the synchronized and the distributed version of the algorithm (whose implementation is often referred to as either the Ford-Fulkerson or the Bellman-Ford algorithm) will converge to the correct path selection. This convergence is in a finite amount of time, assuming proper network behavior (e.g., routers will never stop recomputing paths or receiving updates from its neighbors).

Several issues have been identified and documented throughout the literature with respect to distance vectors. The counting to infinity is a well-documented problem. This is equally true with the proposed solutions such as: the split-horizon and the split-horizon with poisoned reverse.

Open Shortest Path First (OSPF)

The Open Shortest Path First (OSPF) protocol is a protocol of the Link State family. This type of protocol has practically replaced the use of RIP in the Internet. Both of these protocols are executed only within an Autonomous System (AS). The basis for the Link State routing protocols is a database stored and calculated by all routers in the area. This database is a map as to how this router perceives every router in the area is connected and the cost associated with its connection. Every router's database is then exchanged with every other router in the area by a reliable flooding mechanism. Similarly to RIP, routers will update their database based on information received by other router's database.

Eventually, OSPF will converge and steady state can be achieved. Similarly to RIP, information is refreshed in a refresh interval or whenever a change occurs.

It is believed that link state protocols are more efficient, reliable, and contain desirable features (e.g., freedom of loops). Hence, its increase in popularity. ATM has recommended the use of PNNI as its routing algorithm. This routing algorithm is also of the link state vector family. A discussion of this algorithm in detail is included in the next section.

1.3 Current Approach for ATM Routing - PNNI

PNNI Routing is based on the link state routing technique, and supports hierarchical routing to achieve scalability. Nodes are organized into peer groups and hierarchical logical nodes to minimize topological information needed by each node. Peer groups are a collection of logical nodes and are established administratively. Reliable flooding is used for advertising reachability. A topology database is established at each node, which provides all information needed to compute a route from a given node to any address reachable in or through that routing domain. PNNI interoperates with external routing domains, and supports QoS-sensitive path selection (to some extent, efficient QoS-sensitive path selection is still a research issue).

The description in Section 1.3.1 is intended to summarize the workings of PNNI routing. Section 1.3.2 provides more detail, but for a comprehensive view refer to the ATM forum PNNI Specification. Section 1.3.3 provides a description of PNNI routing

packet formats, which are needed to perform analytical calculations of routing overhead, later in Section 2.3.

1.3.1 PNNI Routing Overview

Building the topology database

A physical link is identified by two sets of parameters, one for each direction. Each set consists of the transmitting port ID and the node ID. Physical nodes are lowest level logical nodes. Logical nodes are administratively grouped into peer groups. Neighboring (logical) nodes exchange peer group IDs in Hello packets. If their peer group IDs is the same, they belong to the same peer group. Otherwise they belong to different peer groups, and are border nodes.

A node's local state information is determined by the Hello packets, which provide the status of the link to each neighbor. Information is exchanged on a well-known VCC, the PNNI Routing Control Channel (RCC). Hello packets are sent periodically, and provide the AESA, port ID, node ID, and peer group ID. During link initialization, adjacent nodes within the same peer group synchronize their databases. PTSEs contain topology state parameters and link state parameters. Database synchronization results in the two nodes having identical topology databases. The Hello protocol runs as long as the link is operational, and can therefore act as a link failure detector.

A node's state information is bundled into PTSEs. After database synchronization, PTSEs are reliably flooded throughout the peer group. A node's topology database consists of the collection of all PTSEs received. The topology database provides all the information needed to compute a route to any reachable address

in or through that routing domain. PTSEs are reissued periodically and on an event driven basis. All peer group members maintain an identical topology database.

The nodes of a peer group elect a peer group leader (PGL) in accordance with a leadership protocol. The PGL of the higher level peer group aggregates and distributes information about the child peer group, and floods that information into its own peer group. The functions of the higher level PGL are implemented in the PGL at the lowest level.

The PNNI routing hierarchy is completely described by focusing on the recursive nature of peer groups. The highest level peer group differs only in that it does not need a peer group leader. Logical links between logical nodes in higher level peer groups are usually VPCs.

Path selection during call establishment

Since ATM is a connection-oriented technology, a path selected by PNNI for establishment of a virtual connection will remain in use for as long as that connection remains open. Thus it is critical that PNNI selects paths carefully.

The user specifies QoS and bandwidth parameters that the ATM network must guarantee for that call. PNNI call establishment consists of two parts: 1) selection of a path that appears capable of supporting the QoS and bandwidth requested, and 2) set up of the connection state at each point along the path. The processing of the call at each point along the path confirms that the resources requested are in fact available. If they are not, crankback occurs which causes a new path to be computed, if possible. The final outcome of path selection is either a path that satisfies the request, or refusal of the call.

The routing technique chosen for PNNI path selection is source routing. The source selects the path to the destination based on information available at the node from the topology database. Source routing implies that only the source node is involved in the actual path selection. Therefore, PNNI does not specify any single algorithm for path selection.

Generic connection admission control

Connection Admission Control (CAC) is the process of determining whether or not a node has the resources available to accept the call described in a newly received connection request. It was decided that since only the specific node was involved in this decision, it was necessary to standardize a CAC algorithm. However, a generic CAC was needed as a surrogate for the actual CAC, to in effect predict the outcome of the actual CAC algorithm. The advertised set of topology state parameters must carry information that a generic CAC can use to make this prediction. The actual CAC calculation is performed when the resources are actually being committed to the call.

1.3.2 Detailed Description of PNNI Routing

In PNNI networks, nodes are grouped hierarchically in order to reduce the information required for maintenance by every node in the network. The function of PNNI routing is to build the distributed databases required by PNNI signaling for it to do source routing. Specifically, PNNI signaling uses route calculations derived from the reachability, connectivity, and resource information dynamically maintained by PNNI routing. The sequence of events performed by PNNI routing at initialization is as follows:

- Node configuration.
- Link initialization.
- Topology database synchronization.
- Reliable flooding of PTSEs throughout the peer group.
- Election of peer group leaders.
- Flooding of PTSEs is an ongoing activity to maintain up-to-date technology databases (not part of initialization).

Node configuration

Nodes are configured by assigning a node ID, a port ID for each transmitting port, and a peer group ID, so that each node will know what peer group it is a member of. The PNNI hierarchy begins at the lowest level where lowest level nodes are organized into peer groups. A peer group (PG) is a collection of logical nodes, each of them exchanging information with other members of the group, so that all peer group members have the same view of the group. A logical node at the lowest hierarchical level is a lowest level node, i.e. a switch with a unique node ID.

Link initialization

Logical nodes are connected by logical links. At the lowest level, a logical link is either a physical link or a VPC between two lowest level nodes. Each node determines its local state information, which includes the identity and peer group of the nodes of immediate neighbors, and the status of its links to the neighbors. Link initialization begins with an exchange of information via a well known VCC used a PNNI Routing Control Channel (RCC). Hello packets are sent periodically by each node on the link to exchange ATM End System Address (AESA), peer group ID, node ID, and the port ID for the link. Thus,

link initialization begins when the link becomes operational, and establishes whether the nodes on the two ends of the link belong to the same peer group, or belong to different peer groups. In the presence of certain errors or failures, peer groups can partition, leading to the formation of multiple peer groups with the same peer group ID. Links within a peer group are "horizontal links", whereas links that connect two peer groups are "outside links".

Topology database synchronization

When neighboring nodes conclude that they are in the same peer group, they proceed to synchronize their topology databases. Each node generates a PNNI Topology State Element (PTSE) that describes its own identity and capabilities, information used to elect a peer group leader (PGL), as well as information used in establishing the PNNI hierarchy. The neighboring nodes first exchange PTSE header information. When a node receives PTSE header information that advertises a more recent PTSE version than the one it has, or a PTSE that it does not have, it requests the PTSE and updates its database when it subsequently receives the PTSE. Database synchronization results in link pairs having identical topological databases. When a newly initialized node connects to a peer group, the ensuing database synchronization reduces to a one way topology database transfer.

Reliable flooding of PTSEs throughout the peer group

Reliable flooding of PTSEs throughout a peer group ensures that each node in a peer group maintains an identical topology database. This is the advertising method in PNNI. PTSEs are encapsulated within PNNI topology state packets (PTSPs) for transmission. When a PTSP is received its component PTSEs are examined. Each PTSE is

acknowledged by encapsulating information from its header within an Acknowledgment Packet, which is sent to the sending neighbor. If the PTSE is new or of more recent origin than the receiving node's current copy, it is installed in the topology database and flooded to all its neighbors except the neighbor that originated the PTSE. A PTSE is periodically retransmitted until acknowledged. Only the node that originated a particular PTSE can reoriginate that PTSE. PTSEs contained in a topology database are subject to aging and are removed after a predefined duration unless they are refreshed by new incoming PTSEs. PTSEs are reissued both periodically and on an event driven basis.

Election of peer group leaders

Each peer group elects a peer group leader (PGL). The criterion for election is a node's "leadership priority". The node with the highest leadership priority becomes PGL. The election process is a continuously running protocol. When a node with a higher leadership priority becomes active in a peer group, the election process transfers leadership to that node. When a PGL is removed from the peer group, the election process transfers leadership to the node with the next highest leadership priority. The PGL has no particular function in the internal operation of the peer group. Its function is to represent the peer group to the hierarchically next higher peer group. This function is to aggregate and distribute information for maintaining the PNNI hierarchy.

Peer groups at levels higher in the hierarchy

A higher level peer group (an abstraction) has essentially the same properties as the lowest level peer groups. The peer group members (logical group nodes) of the next higher peer group each represent a peer group at the lowest level. The functions of the logical group node (LGN) of the higher level peer group and the PGL of its child peer

group are closely related. The functions of a LGN include aggregating and summarizing information about its child peer group, and flooding that information into its own peer group (the higher level peer group). A LGN also passes information received from its peer group to the PGL of its child peer group for flooding.

A three-level hierarchical network is illustrated in Figure 1.6. The physical network contains 22 nodes. This is configured into 7 lowest level peer groups, which are labeled as follows: A.1, A.2, A.3, B.1, B.2, B.3, and C. There are 2 second-level peer groups labeled A and B. The highest level peer group has peer group members representing second-level peer groups, A and B, and a lowest level peer group, C. The highest level peer group differs from all other peer groups in that it is not represented by a LGN in a higher peer group. Note that each peer group has a designated (elected) PGL. The functions that define the PGL of peer group A are located in node A.2, which is in turn implemented on the switching system contained in the lowest level node A.2.2.

The notation used here to label nodes and peer groups is representative of the address structure (node Ids and peer group Ids) actually used. An LGN is identified by a node ID. This by default contains the peer group ID of the peer group the node is representing. For example, LGN B.3 contains the peer group ID of peer group B.3, which it is representing. A higher level (or ancestor) peer group has a shorter address than its child peer groups. It is meaningless to directly compare addresses for peer groups where neither is an ancestor of the other. In Figure 1.6, for example, lowest level peer group C has the same level address as second level peer group B. Neither is an ancestor of the other. Similarly, node C.1 and node B.3.3 are both lowest level nodes.

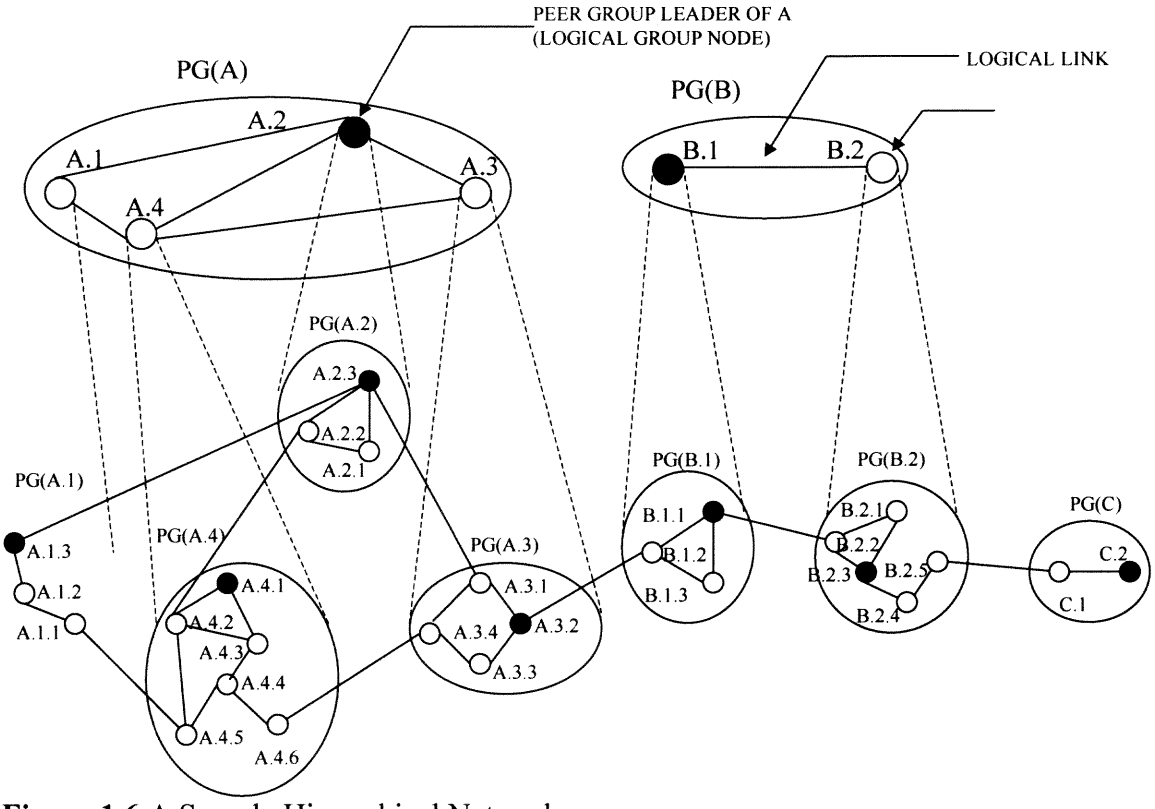

Figure 1.6 A Sample Hierarchical Network
Source: The ATM Forum Technical Committee, *Private Network-Network Interface Specification Version 1.0 (PNNI 1.0)*, March 1996.

PNNI path selection

As stated above, PNNI uses source routing for all connection setup requests. The user issues a connection request, which includes source/destination addresses and QoS/bandwidth requirements. Since PNNI allows multi-level hierarchical routing, the originating switch selects a path to the destination based on the detail of the hierarchy known to it from its topology database. The selected path, encoded as a Designated Transit List (DTL), is explicitly included in the connection request. Such a path is not a fully detailed source route outside the peer group of the originator. Instead, those portions of the path are abstracted as a sequence of LGNs to be transited. The connection request is routed in accordance with the source route, using the VPI=0 signaling channel.

When the connection request arrives at the entry switch of a peer group, that switch is responsible for selecting a lowest level source route across the peer group to reach the "next hop" destination specified by the higher level path.

The implication of source routing, on which only the source node is involved in selecting the path, is that it is not necessary for all nodes to use the same path selection algorithm. Accordingly, PNNI does not specify a path selection algorithm. Each implementation is free to use whatever path selection algorithm it feels is appropriate. The path selection algorithm presented in Appendix H of the ATM Forum PNNI Specification is considered acceptable.

Generic Connection Admission Control (GCAC)

During the connection setup, each switch along the selected path performs Connection Admission Control (CAC) to ensure that the connection can be supported without jeopardizing QoS guarantees to the existing connections. Since this determination involves only the particular node, PNNI does not specify a single CAC. A generic CAC predicts the outcome of the actual CAC. The actual CAC is used at the time that resources are committed to a connection. Each node along the path makes a determination on available resources using the GCAC. A GCAC determines if a node is likely to have sufficient resources to support the call. If sufficient resources are available, the connection request is forwarded to next hop, otherwise the connection request is cranked back till a node is reached that can calculate a new forward route toward the destination.

When a node accepts a connection, its ability to accept a new connection may change. If the change is significant, it will trigger new PTSE instances describing this

node's updated resource capability. Determination of a "significant" change is defined by system configuration parameters. Since PNNI does not specify any single required CAC algorithm for determination of sufficient resources for a node to support a connection, another node could use a different CAC algorithm.

1.3.3 PNNI Routing Packet Formats

PNNI routing packet formats have been designed to provide flexible and expandable encoding to accommodate future versions of PNNI. The format relies on nested type-length-value (TLV) information groups. All type and length fields are each two octets long, in all information groups. All information groups are padded to a multiple of 4 octets, if necessary.

PNNI routing packets types

There are 5 packet types used for PNNI routing, as shown in Table 1.2. Each packet type begins with a common PNNI packet header 8 octets in length. PNNI packets are made up of headers and information groups (IGs). Some IGs may have child IGs. Information groups that may be included in PNNI routing packets are shown in Table 1.3.

Table 1.2 PNNI Routing Packets

PACKET TYPE	PACKET NAME	FUNCTION
1	Hello	Link initialization & maintenance
2	PTSP	Flood PTSEs to peer group
3	PTSE Acknowledgment	Acknowledgment for reliable flooding
4	Database Summary	Exchange PTSE headers (database synchronization)
5	PTSE Request	Request new or updated PTSEs

Source: The ATM Forum Technical Committee, *Private Network-Network Interface Specification Version 1.0 (PNNI 1.0)*, March 1996.

Table 1.3 Information Groups

TYPE	PACKET NAME	MAY CONTAIN IGS
1	Hello	Aggregation token Nodal hierarchy list Uplink information attribute LGN horizontal link extension Outgoing resource availability System capabilities Optional GCAC parameters
2	PTSP	PTSE Nodal state parameters Nodal information group Outgoing resource availability Incoming resource availability Next higher level binding Optional GCAC parameters Internal reachable ATM addresses Exterior reachable ATM addresses Horizontal links Uplinks Transit network ID System capabilities
3	PTSE Ack	Nodal PTSE Ack System capabilities
4	DB Summary	Nodal PTSE summaries System capabilities
5	PTSE Request	Requested PTSE header System capabilities

Source: The ATM Forum Technical Committee, *Private Network-Network Interface Specification Version 1.0 (PNNI 1.0)*, March 1996.

Hello packet

Hello packets are exchanged between neighboring nodes using RCCs.

Hello packets are transmitted by each node over:

- all physical links to immediate neighbor nodes,
- all virtual path connections for which this node is an endpoint, and
- all SVCCs, established for the purpose of exchanging PNNI routing information, for which this node is an endpoint.

Hello packets include the originator's AESA, Node ID, Peer Group ID, and Port ID, and after receiving information from the neighboring node, also includes the remote node ID and remote port ID. It also includes the frequency at which Hello packets are sent. The basic Hello packet is 100 octets long (including PNNI header).

Hello packets sent over outside links include, in addition to the basic Hello packet, the following IGs: Aggregation Token, Nodal Hierarchy List, and Uplink Information Attribute. The Aggregation Token adds 8 octets. The Nodal Hierarchy List adds 12 octets, plus 56 octets for each hierarchical level beginning with the parent level and proceeding until the highest level has been listed. The Uplink Information Attribute adds 8 octets, plus all outgoing Resource Availability Information Groups (32 octets plus an optional 12 octets), plus any additional optional IGs needed to describe the reverse direction of the uplink.

The aggregation token serves, along with the remote node ID, to identify uplinks, which are to be aggregated at the next level of the hierarchy. Hello packets sent over LGNs as part of the LGN Horizontal Link Hello Protocol include the LGN Horizontal

Link Extension IG. This group adds 8 octets, plus 12 octets for each LGN horizontal link.

The network capacity needed to support the Hello protocol is a function of the number of physical and LGNs, the average number of links at a node, the average number of levels of hierarchy for all nodes (including LGNs), and the frequency of transmitting Hellos. The length of the Hello packet depends on the type of link it is to be sent over.

PNNI Topology State Packet (PTSP)

PTSPs are used to distribute (by flooding) information throughout a peer group. They are also used to send PNNI topology information to a neighboring in response to a PTSE Request packet. The PTSP contains one or more PTSEs, all from a single originator. The PTSP header is 44 octets long, which includes the PNNI header, originating node ID, and originating node's peer group ID.

PNNI Topology State Element (PTSE)

PTSEs are the units of information for flooding and re-transmission. The collection of PTSEs constitutes a node's topology database. Each PTSE includes the PTSE header, whose length is 20 octets. The PTSE header indicates which "top level" information groups may appear in the PTSE.

PNNI flooding

PTSEs are encapsulated within a PTSP and flooded to all neighboring nodes within the peer group. When a PTSP is received, its component elements are examined. If a PTSE is new, or more recent than the node's current copy, it is installed in topology database, and flooded to all other neighboring peers. The fact that the PTSEs were sent to these

neighboring peers is remembered, and they will be retransmitted until acknowledged. Each PTSE is acknowledged by sending a PTSE Acknowledgment packet back to the neighboring peer.

PTSE acknowledgment packet

The PTSE acknowledgment packet is used to acknowledge receipt of PTSEs from a neighboring node. The packet consists of the 8-octet PNNI header, and for each set of PTSE acknowledgment about one node, a 28-octet IG. The last field of this IG gives the AckCount, which is the number of acknowledgment for this node. For each acknowledgment (one for each count in AckCount), a 12-octet data structure is included. The packet length is $8 + (28 + 12\ a_i)*n$, where a_i is the AckCount of the ith set of PTSE acknowledgment included, and n is the number of sets.

Database summary packet

Database Summary packets are used during the initial database exchange process between two neighboring peers. The database exchange process involves a sequence of Database Summary packets, which contain the PTSE header information of all PTSEs in a node's topology database. Database Summary packets also contain a sequence number and flags used to negotiate the master/slave relationship necessary to ensure proper functioning of the lock-step protocol. A node sends a Database Summary packet, and the other side responds with its own Database Summary packet, implicitly acknowledging the received packet. At most one outstanding Database summary packet between the two neighboring peers is allowed at any one time. The Database Summary packet consists of a 16-octet header (which includes the PNNI header), and a 44-octet IG for each set of PTSEs in the topology database. In addition, for each PTSE summary, there is a 16-octet

data structure. The packet length is $16 + (44 + 16\ a_i)*n$ where a_i is the number of PTSE summaries in the ith set of PTSEs included in the packet, and n is the number of sets.

PTSE request packet

PTSE request packets are used during database synchronization. When a received Database Summary packet contains a PTSE header that either has not been seen before, or is a more recent version of a PTSE currently in the node's topology database, such PTSEs are requested from the neighboring peer. PTSE Request packets consist of an 8-octet PNNI header, a 28-octet IG for each set of PTSEs requested, and a 4-octet data structure for each PTSE in a set of PTSEs. The packet length is $8 + (28 + 4\ a_i)*n$, where a_i is the number of PTSEs requested for the ith set of PTSEs, and n is the number of sets requested. When a PTSE Request packet is received, the requested PTSEs are bundled into a PTSP for transmission to the neighboring peer.

1.4 Modeling Approach for ATM Networks

The task of obtaining sufficiently accurate analytical approximations that predict network performance has proven to be a very difficult one. This is because of the new sophisticated control mechanisms (in terms of flow and congestion control) and dynamic routing algorithms used by the technology used in the new communication networks. Up to now, the two typical solutions provided are the following:

- an analytical closed form solution (i.e., a solution that relies in a series of assumptions and approximations to make the problem tractable), and
- a modeling and simulation solution (i.e., a solution that basically mimics the environment and traffic conditions under which the network is operating).

However, it should be noted that these solutions are not totally independent. Usually, the analytical closed form solution relies heavily in the modeling and simulation solution to justify that the assumptions and simplifications in the analytical solution are valid. Also, the modeling and simulation solution will usually use a closed form solution to verify: (1) the correct implementation of the model, and (2) that all of the most relevant features of the system (i.e., the ones that affect performance the most) have been properly captured by the model.

The tradeoffs encountered when dealing with these two solutions are simple to state. The former one, the analytical closed form solution, although attractive because of its closed form, usually exhibits an accuracy problem. The analyst using this type of solution is forced to make an extensive set of assumptions to keep the problem within the "tractable" domain. This makes use of the analytical model, for the purpose of analysis of network performance, very impractical. The latter one, the modeling and simulation solution, although attractive because of its accuracy, usually encounters two types of problems: (1) difficulty of validating the model, (2) excessive simulation running, (i.e., execution) time. Interestingly enough, in most cases the more detailed the model is (i.e., the model that has more features implemented at a high level of fidelity), the easier is to validate the model (it can be validated at the algorithm level). However, usually the more detailed the model implementation is, the longer the simulation execution time becomes.

For the analysis of new technologies, the simulation running time problem is bound to at least stay the same. This is because, although machine speed is dramatically improving, so are the traffic loads carried by the systems that need to be analyzed. Among those systems, one can find the Asynchronous Transfer Model (ATM), the Wide

Area Network standard of the Broadband Integrated Services Digital Network, which is supposed to handle billions of information cells. The amount of cells handled make the use of a detailed model almost prohibitive. However, there are no known good analytical models for ATM networks.

1.4.1 High Resolution ATM Model

The high resolution ATM model was built using the "modeling along physical lines" approach. Software was written in separate modules that represented the different entities which needed to be modeled to represent behavior of the ATM switches, the links interconnecting them, and the user (also called subscriber) itself. For a detailed description about ATM refer to [15]. As a result of using this approach, the simulation contains the following models:

- Physical Layer Model (representing links and virtual path service handling)
- ATM Switch Model consisting of:
  - ATM Adaptation Layer Model (representing the layer in charge of assembly/reassembly of data to be delivered to the user network interface)
  - ATM Layer Model (representing the Virtual Path Switching Processor and the Virtual Channel Switching Processor with their corresponding queues). In addition, a network layer using a saturation routing algorithm was implemented here.
- User Network Interface Model (representing negotiation and data exchange for virtual circuit setup as well as information flow from the user to the network and vice-versa). Flow and congestion control techniques were implemented here.

- Subscriber Model (representing generation of variable bit rate, available bit rate, and constant bit rate demands. The rules for priority bit tagging were implemented here.

Emphasis on the model was placed on simulating the delays that a transportation of cells will experience in a given network environment. Therefore, when a cell arrives to an ATM switch, it then experiences queue and processing delays at the virtual path and the virtual channel processors. Cells may also wait at the virtual path transmitter queues before they are serviced. The main objective of the simulation was to capture the inherent extra delays found in an ATM network when traffic loads increase in a very similar way as that expected from a real system implementation. Because of its nature, this ATM model (high and low) is considered appropriate to examine flow control and policing procedures for ATM networks. Also, as it is shown in the result section, the measure used to compare results between two models is cell delay. A table that summarizes delays encountered by cells when transported by an ATM network is included below (Refer to Table 1.4). This table includes areas of delays and whether there were any significant differences in model implementations between the high and low resolution ATM models.

As shown in the table below, the low resolution ATM model differs the most from the high resolution ATM model at the lower layer levels. That is justified by the fact that most of the CPU time used by the simulation for a typical information exchange will happen at these levels because they occur at each ATM switch rather than only at source and destination switches. As a result, the benefits in the terms of gaining simulation speed can be substantial when using the low resolution ATM model.

Table 1.4 Summary of Differences between Low and High Resolution Models

DELAYS	MODELS INVOLVED	DIFFERENCES
Transaction or call setup	All	Minor
Packetizing/Depacketizing	Subscriber Model	None
ATM Adaptation Layer	CS and SAR Layer at the source and destination switches	None
Virtual Path and Virtual Channel Switching queuing and processing	ATM Layer at all switches	Different
Virtual Path transmitter queue/transmission/ propagation/HEC verification	Physical Layer at all interconnections	Different

1.4.2 Low Resolution ATM Model

The user can invoke the use of a low resolution ATM model to obtain better runtime performance when using the ATM simulation. The simulation of the ATM network, when using the option of the low resolution model, first uses both the high resolution and low resolution ATM models. The ATM simulation is able to switch from the high resolution to the low resolution ATM model automatically based on accuracy comparisons.

The description of the ATM low resolution model has been split into three different areas. The first describes the route selection between source and destination subscribers. The objective is to ensure, as much as possible, that the ATM low resolution model selects routes that are comparable to those selected by the ATM high resolution model. Otherwise, it would be overly optimistic to expect that both models provide similar results. The second area deals with the transport of information (i.e., voice, data, and multimedia between subscribers). The ATM low resolution model implementation

attempts to provide results that are a function of traffic loads. The third area provides a detailed implementation description of the Low Resolution ATM model.

Virtual circuit setup procedure

The route selection process (arbitrarily set to saturation routing), in the ATM low resolution model, is identical to that found in the high resolution model. It was designed as such to guarantee that all information collected and stored when using the high resolution model is also collected and stored when using the low resolution model. The only modification encountered will be that the simulated delays encountered for the waiting time due to processing and queuing experienced at the virtual path and virtual channel switching processors will be changed to a fixed average processing plus queuing delay when the low resolution model is operating. This will allow for the simulation, when operating in the low resolution mode, to more accurately predict delays encountered by routing messages when selecting a route between source and destination subscribers.

This delay is necessary because ATM information cells (i.e., cells used to transport end-to-end user information) will not be simulated as sent throughout the network (i.e., delays at the nodes will be determined by different means). It is important to mention that this mode of operation (assuming constant average delays) will temporarily persist even when the mode of operation has been switched to the high level resolution model. Without it, a high to low resolution switch in the simulation would signify that ATM cells would find empty queues at the proceeding nodes making the simulated system more efficient than the real system would perform. However, it is

anticipated that once the ATM low resolution model has been invoked there will be no need to switch back to the high resolution ATM model.

The statistics collected during the high resolution mode of operation to be used in supporting low resolution operation is the mean of the average cell delays.

Data transport

The high resolution ATM model will save the path selected by the flood process (i.e., nodes and links involved in the setup procedure). The ATM low resolution model will access this information, every time a cell is generated at a source subscriber/host. When a cell is generated, a process will inform the queues of each link and node processors (i.e., virtual path switch and virtual channel switch processors) that a cell has a requirement for their services. In return, the link and node processors will provide information regarding their current loading status. An example is included below to further clarify the mode of operation of the ATM low resolution model.

For example, let's say that a virtual circuit has been established between two subscribers (from node 1 to node 4). The path selected (shown in Figure 1.7) traverses node 1,2,3, and 4. Out of the previous nodes, node 2 is an intermediate/relay node of the preestablished virtual path 1-2-3 (i.e., labeled virtual path 11). In the high resolution model, cells generated for this virtual circuit will go through the following delays:

- Virtual channel and virtual path switching processors of node 1 (processing plus queuing delays)
- Transmission plus queuing delay for link 5 in the virtual path 11 domain
- Virtual path switching processor of node 2
- Transmission plus queuing delay for link 3 in the virtual path 11 domain

- Virtual channel and virtual path switching processors of node 3 (processing plus queuing delays)

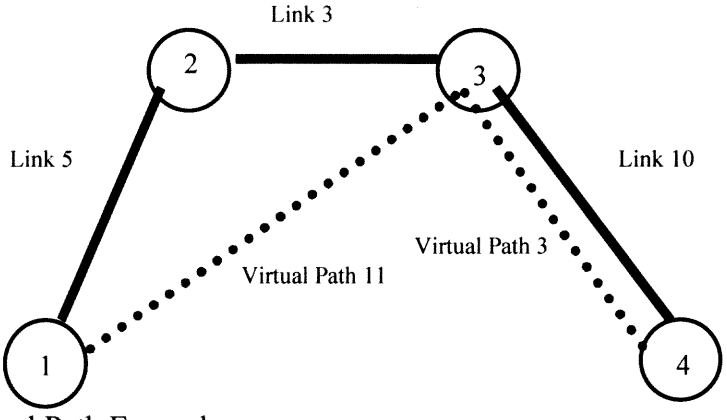

Figure 1.7 Virtual Path Example

- Transmission plus queuing delay for link 10 in the virtual path 3 domain
- Virtual channel and virtual path switching processors of node 4.

As it can be seen, the delays are split into two categories: (1) deterministic (which are processing delays) and (2) statistical (which are queuing delays) and depend on the state of the queues. The queuing delays are correlated to the state of the queue at the time that the cell demanding service arrives. For example, if the queue is empty, the queuing delay will be zero. Accordingly, if the queue contains cells that cumulatively required five milliseconds of processing, the arriving cell will experience that delay. The delays encountered at each node will be a function of time. For example, let's assume that for a particular ATM cell, the following delays (as described in Table 1.5) were experienced at each of the nodes of the previous example. The state of the queues as a function of time is described in the following graphs (Figures 1.8, 1.9, and 1.10).

In the ATM high resolution model, at time equal to zero, (relative time), the cell arrives at node 1 encountering a delay of 100 microseconds (Refer to Figure 1.8). The cell will not experience any delay at node 2 because it does not require a new virtual channel assignment. The cell arrives at node 3 at a time equal to 190 microseconds (Refer to Figure 1.9). At that queue, the cell will experience a delay equal to 50 microseconds before it is processed. Finally, the cell arrives at node 4 at a time equal to 305 microseconds (Refer to Figure 1.10). At that node, it encounters a delay equal to 90 microseconds.

The implementation and description of the low resolution ATM model is simple and is described below. The low resolution model assumes that the load applied in each of the nodes by a single cell will happen simultaneously rather than throughout simulated time for that particular cell, as explained in the example. For instance, the delays experienced by the cell at different nodes will be recorded at relative time equal to zero (i.e., at the time that the cell is generated by the subscriber). As a result, the delays recorded by the low resolution ATM model will be as described in Table 1.6.

Table 1.5 Delays Experienced by an ATM Cell

Node Number	Queuing Delay at the Virtual Channel Switching Processor (Microseconds) at a specific time
1	100
2	No delay
3	50
4	90

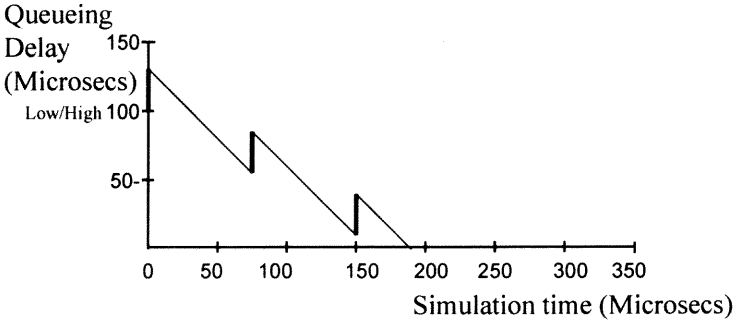

Figure 1.8 Queueing Delay of Node 1

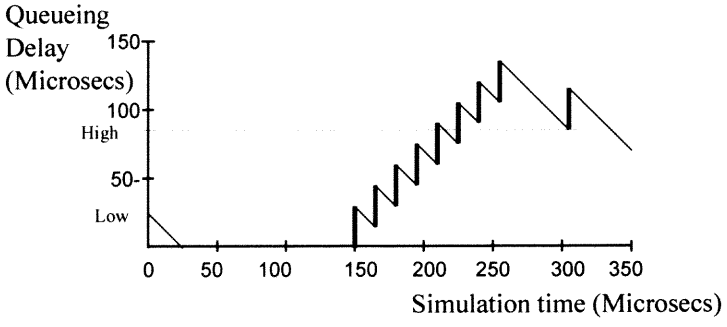

Figure 1.9 Queueing Delay of Node 3

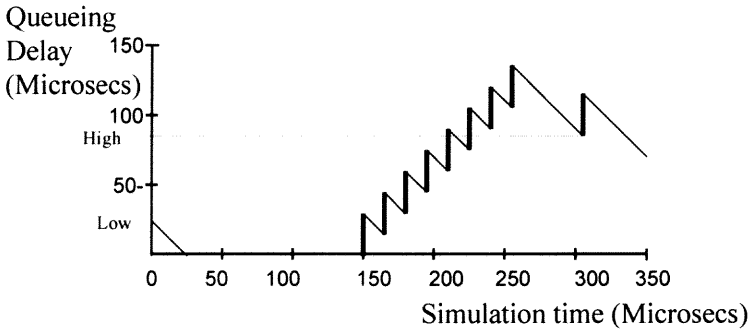

Figure 1.10 Queueing Delay of Node 4

Table 1.6 Delays Simulated in the Low Resolution Model

Node Number	**Queuing Delay at the Virtual Channel Switching Processor (Microseconds) at a specific time**
1	100
2	No delay
3	150
4	25

As one can see, individual cell delays can vary by substantial amounts. However, if one is interested in average cell delays, the low resolution model is believed to track the high resolution model within acceptable limits for analyzing most flow control and policing procedures.

The most significant advantage of the low resolution model is that it does not change significantly the process of selecting a route between source and destination. The model also shows a similar kind of sensitivity to traffic load as the high resolution model does. The only difference lies into the fact that loads applied to each of the nodes are shifted in time. Assuming that there is insignificant correlation among the traffic being generated, the low resolution model provides insightful results. Even in the extreme case,

that there is heavy correlation of traffic generation patterns (which is not common for most cases because of the vast number of cells generated during a session on which a virtual circuit has been established), the low resolution model will still provide meaningful results.

Unfortunately, none of this can be formally or mathematically proven. It is believed that the time resolution needed to model ATM environments with subscribers creating demands that require a tremendous amount of cells transported, can be slightly distorted and compacted without significant differences in results. In order to prove this point via simulation techniques, the ATM simulation has the capability to simultaneously run the same traffic through both the low and high resolution ATM models and compare results. Prior to switching from the high resolution mode to the low resolution mode, the simulation will compare both the actual results obtained using the high resolution model and the predicted results using the low resolution model. If the difference is significant (significant is a term that will be defined by the analyst), the simulation will not change modes. If the difference is not significant, the change will be done automatically. This will provide confidence to the analyst that the use of a low resolution model is justified based on the scenario conditions and desired accuracy rather than assumptions made by modelers.

Detailed implementation of the low resolution ATM model

Whenever a cell is generated, at a user access point, delays will be estimated for each node and each link that constitutes the selected path of virtual circuit. These calculated delays would be a function of the cells present at that node/link when the current cell was generated. To know which cells are present at each node/link at any time, a list is kept

and updated as necessary, containing the most recent cells generated for virtual circuits traversing a node/link. The newly generated cell will become an entry on that node's list which will, in turn, affect the delays associated with cells that are generated in the "near" future and have the same node on its selected path. The time window (impact window) that defines "near" future or "recent" levels of activity is a function of the load being processed by a node. The more traffic a node is handling the larger an impact window a cell transmission will create. This will help preserve the fact that highly congested nodes will experience exponentially increasing delays. The total cell delay time, for simulation purposes, will be a cumulative function of the estimated delays associated with each of the nodes/links of the selected path. The cell will be scheduled for delivery at that cumulative delay. The only exception to that is that a cell will never be delivered prior to one of its predecessors generated at the same circuit. There will be a safeguard check that guarantees that cells will be delivered in order.

To summarize, every time a cell is generated an entry will be entered in a list associated with nodes and links belonging to the selected path of the virtual circuit. The delay impact on all of the nodes and links will be immediate rather than stretched through the delivery time span. The impact window of this cell will be determined by the delay calculated at each node. By adding each of the delays encountered at each link and the node of the path, a total delay will be calculated for the cell. Because the average delay will not be affected by interchanging cell delay times for cells belonging to the same virtual channel, the low resolution ATM model will simply force cells to be delivered in order.

1.4.3 Results

A ten-node network was used to compare results when using the Low and the High resolution ATM models. The first table (Refer to Table 1.7) shows the results obtained with the high resolution model (including ATM cell average delay and Estimated CPU time as a function of the traffic level applied to the simulation). The second table (Refer to Table 1.8) shows the results obtained with the Low Resolution ATM model (including the % Accuracy and Estimated CPU time).

Table 1.7 Results Obtained with the High Resolution ATM Model

Relative Traffic Level	Average Delay (Milliseconds)	Est. CPU time (minutes)
15	5.33	100
17	5.49	113
19	5.60	120
21	5.57	135
23	5.76	148
25	6.39	160
27	6.5	170
29	7.4	181

Table 1.8 Results Obtained with the Low Resolution ATM Model

Average	Est. CPU	%
5.55	20	3.79
5.71	21	3.77
5.73	23	2.39
5.92	24	5.98
6.15	25	6.31
6.13	27	4.26
7.22	29	10.01
7.42	31	0.19

CHAPTER 2

SATURATION ROUTING FOR ATM

This chapter will show that the virtual path selection by a saturation (flooding) technique is a viable adaptive routing solution for substantial data exchanges in an ATM network. Adaptive routing algorithms are the leading techniques used today in Packet Data Networks. Typically, current measurements are used to derive non-congested new routes for future traffic. However, often, current measurements are not reliable enough in deriving effective routing decisions for an ATM network. In fact, it has been reported that in ATM networks current traffic measures could have little correlation to traffic patterns even in the near future [22]. This is because of the highly dynamic nature of not only the traffic but also of the underlying architecture of the network itself. This is even more extreme in military or mobile applications in which the topology of the network changes frequently. In [9], it has been suggested that the only solution for topologies with a high rate of change is flooding. Furthermore, it has been suggested that the trends indicate that the cost of the control of the network, of which routing is a prominent component, has been increasing more rapidly than any other [4]. As a result, increased emphasis should be placed on simple routing techniques for ATM networks.

For a successful implementation of adaptive routing techniques, it is critical that the rate of change in routing demands is smaller than the reaction time of the routing algorithm, the latter always being larger than the time window between measurements. Saturation routing offers a host of well-known advantages, which include accuracy and simplicity. Also, a potential benefit of this type of routing scheme, not exploited as of

now, is its potential for multicast routing. Despite the potential of saturation routing, it has been discounted in the past, because of its perceived large overhead. Here, the amount and effect of the saturation overhead is analyzed, in the transport of substantial data exchanges (e.g., video, multimedia, teleconferencing, etc.); and, in fact, found to be small in amount and insignificant in effect, over a large range of network parameters. In [12], it has been forecasted that future routing protocols may result to "old" switching techniques, such as saturation routing, where transmission efficiency is traded for simplicity. In this paper, it is demonstrated that for a large number of anticipated ATM services for normal data exchanges, the transmission efficiency when using saturation routing is not degraded significantly; and therefore, the tradeoff should be made.

2.1 Description of the Saturation Routing Algorithm

The proposed saturation algorithm, which is illustrated by using a small network shown in Figure 2.1, operates as follows. When a request for a virtual path setup is initiated at the source [Node S] for a given destination [Node D], Node S sends a routing cell (i.e., a cell that contains information about the circuit setup) along its associated links (i.e., those that are connecting Node S with Nodes A1 and A2) [shown as step 1]. Nodes A1 and A2 receive this request and because they are not the intended destination, they forward the routing cell in all of their outgoing links (i.e., all of the links except the one on which the routing cell was received) [shown as step 2].

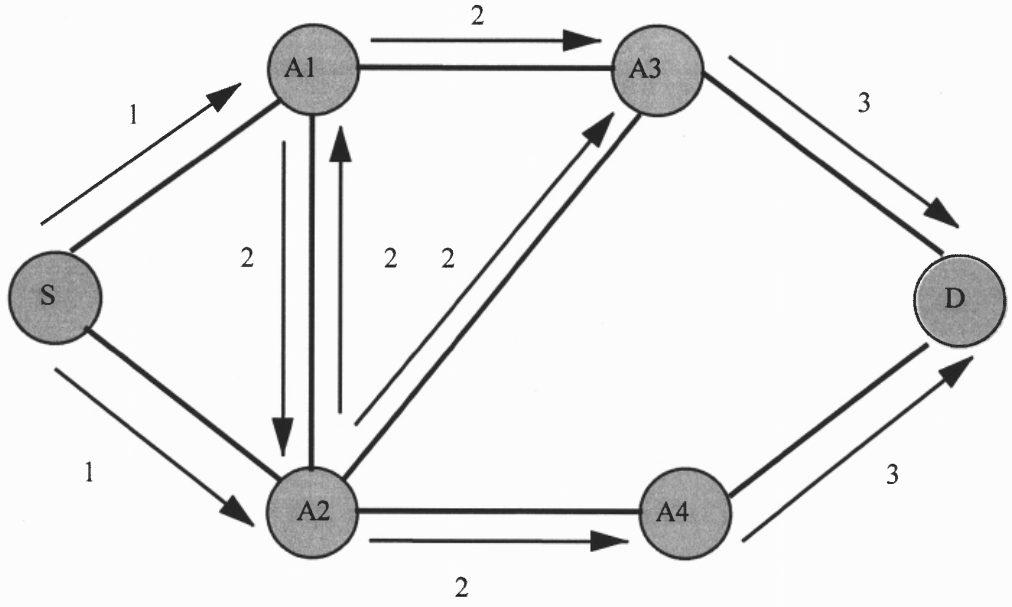

Figure 2.1 Saturation Routing Algorithm Illustration

As a result, Node A2 sends the routing cell to Node A1, while Node A1 sends the routing cell to Node A2 (note that two routing cells for the same path setup attempt is traversing the same link (i.e., link A1-A2), but in opposite direction). Therefore, Node A1 receives a second routing cell for the same attempt (these are the one from Node S and the one from Node A2). Node A1 discards this second cell because it checks its memory and finds a record of a "recent" request for the same attempt has already been processed. At the same time, Node A3 receives requests from both Node A2 and Node A1. Node A3 accepts only the first arriving cell and forwards it to Node D. Node D accepts the earliest routing request it receives (which represents the best route by construction), and sends a cell back to the source following the path selected. This provides the successful new virtual path setup at the source. It should be noted, that a routing cell for the same setup request can not traverse any link more than two times (once in each direction). Therefore, the total number of routing cells generated is upper

bounded (refer to inequality 2-1). This important property of the saturation routing algorithm will be used later on in the analysis.

$$R \leq L_{TOT} \bullet 2 \qquad (2\text{-}1)$$

Where:

R is the total number of routing cells generated for a single routing effort

L_{TOT} is the total number of links in the network

2.2 Analysis of Routing Overhead Incurred by Saturation

To show that the impact of using a saturation technique to establish the route for a virtual path setup is small, it firstly requires to show that the added overhead due to routing is small compared with the amount of information exchanged in the virtual path. As it will be shown later, the impact of using a saturation routing algorithm depends on the saturation routing factor f (defined as the ratio of Routing cells generated due to a data exchange over Information cells generated for that data exchange, $f \equiv \frac{R}{I}$). The proof will be based on an upper bound value for the saturation routing factor, f.

2.2.1 Assumptions

To carry out the analysis, the following assumptions will be made:

- each source in the network is equally likely to generate traffic – a typical assumption made in this type of analysis,

- the amount of overhead needed to convey routing information can fit in one single ATM payload (i.e., 48 Octets) cell – which is elaborated further in the ATM Cell Design Subsection,

- the transmission exchange is at least modest or not small in size by ATM service standards, about 1 million cells, which is approximately 1 minute of exchange at 8 Mbps or more – which is elaborated further in the User Exchanges for which Saturation Routing should be invoked Subsection, and

- a typical large ATM network ranges from about 100 to 1000 nodes – Typical Size of Networks Before Using Hierarchies Subsection

2.2.2 ATM Cell Design

For the saturation routing scheme to work there are several requirements:

- each ATM switch can identify each individual virtual circuit channel setup,

- each ATM can identify the virtual path over which a routing request arrived, and

- each routing cell is labeled accordingly.

The first requirement can be accomplished by using 4 octets (1 for the Virtual Path Identifier (VPI) in the User Network Interface (UNI), 2 for the Virtual Channel Identifier in the UNI, and 1 for a sequence number to differentiate between subsequent requests using the same VPI-VCI combination). The second requirement can be met by including the VPI used in the Network to Network Interface (i.e., 1.5 octets). The third requirement can be met by using the Common Part Indicator of a Type 5 service provided

by ATM (Type 5 requires 8 bytes of overhead). In addition, 20 bytes are needed to specify the ATM End Destination Address. The total routing cell length required is 33.5 octets, leaving enough space (i.e., 14.5 octets) to attempt to perform optimization in the routing selection. Refer to Figure 2.2.

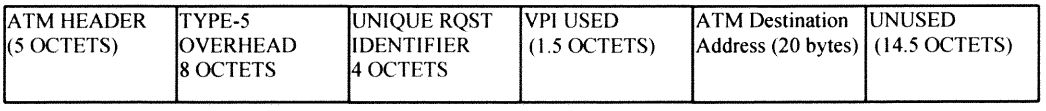

Figure 2.2 ATM Routing Cell Layout

2.2.3 User Exchanges Applicable to Saturation Routing

The user exchanges for which Saturation Routing should be invoked are those which are substantial in size, about 1 million cells, which is approximately 1 minute of exchange at 8 Mbps or more. Reference [11] suggests that there are several types of traffic that meet this requirement (i.e., all services listed below but two). As it can be seen (refer to the Table 2.1 and Figure 2.3 shown below), there are several applications that can benefit from this routing algorithm without being adversely affected from a performance perspective. This is because of their expected transmission exchanges surpasses the calculated value of 1 million ATM cells. The table indicates services (e.g., High Definition TV, High Quality TV, Medical, etc.) which are expected to exchange significant amounts of data. The Figure 2.3 shown below indicates an independent estimation of transmission requirements for high bandwidth services. Also, in here is shown that the High Speed Data and High Quality Video services are anticipated to surpass the 1 million cells (i.e., those services that fall in the right of boundary line 1). In addition, a boundary line 2 is shown with a less restrictive condition of a needed

exchange of only one hundred thousand cells. This lesser requirement can be used to determine the services for which the routing algorithm will have negligible impact. However, it used as its basis typical ATM networks, rather than worst case 1000-node networks (used to determine Boundary Line 1 - Worst Case Analysis).

Table 2.1 Bandwidth Range for Different Services

Service Type	Service Category	Bandwidth Range (Mbps)	CBR/VBR
Voice	CD Quality Voice	1.4	CBR
Data	LAN Interconnection	1.5-100	VBR
Data	Client/Server System	10-100	VBR
Data	Remote Data Base Access	1-10	VBR
Data	Remote Procedure Call	6-60	VBR
Data	Mainframe CAD/CAM	1.5-36	VBR
Video	Video/Image Mail	1-4	VBR
Video	NTSC-Quality TV	15-44	VBR
Video	HDTV-quality TV	150	VBR
Video	Video Browsing	2-40	CBR
Video	Medical X-ray	1.5-10	CBR/VBR
Video	Medical MRI/CAT	10-200	CBR/VBR
Video	High Resolution Graphics	100-1000	VBR

Source: DuBose, K., and Sim, H., "An Effective Bit Rate/Table Lookup Based Admission Control Algorithm for the ATM B-ISDN," *Proceedings, 17th Conference on Local Computer Networks*, September 1992

2.2.4 Typical size of Networks Before Using Hierarchies

Most routing protocols need scaling if the network becomes too large. Therefore, a 1000-node ATM network is considered by most references as a large network. The PNNI Standards addresses this problem by introducing levels of hierarchy. A similar approach will be proposed in the Saturation Routing if required to run in a larger network. For example, Reference [10] recommends that for the Open Shortest Path First protocol (OSPF, a routing protocol used in the Internet), 200 routers (the equivalent of nodes) should be the maximum number in an OSPF Area (the equivalent of a network).

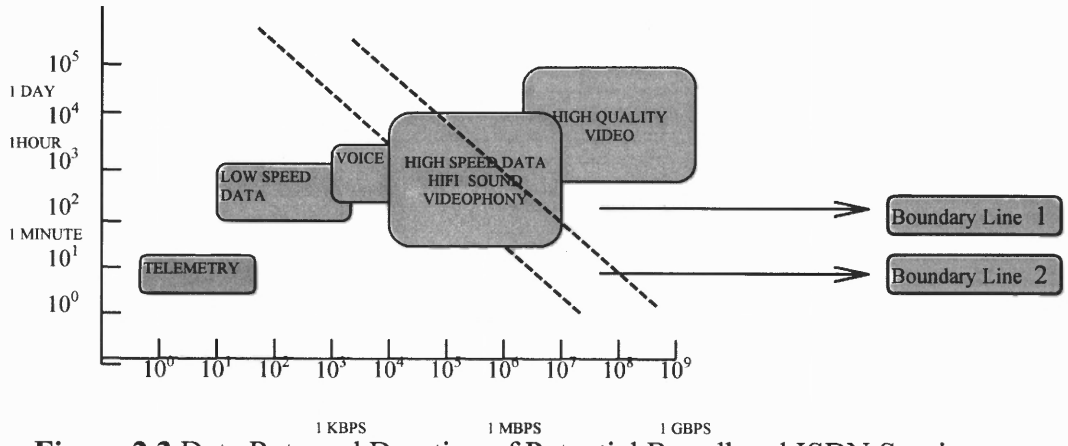

Figure 2.3 Data Rate and Duration of Potential Broadband ISDN Services

2.2.5 Theoretical Analysis of Overhead

To carry out the overhead analysis, routing overhead will be considered as though it only affected the selected path rather than the entire network. In other words, the approach here is to trade the average routing cell load over all network links in space at one instance of time with the average routing cell load over one link in space over a long period of time. A more elaborated discussion of this follows below.

The objective is to prove the following: the average fractional increase in cell arrival rate of the number of cells generated, because of saturation routing, is not greater than f (defined as the saturation routing factor). By examining the load of a particular link as a function of time, one can observe the load due to the transmission of routing cells (see the rectangles in Figure 2.4). Some of those routing cells are for what is defined as successful routing efforts (labeled as S) and some are for what is defined as unsuccessful routing efforts (labeled as U). After each successful routing effort, as it may be expected, user load follows.

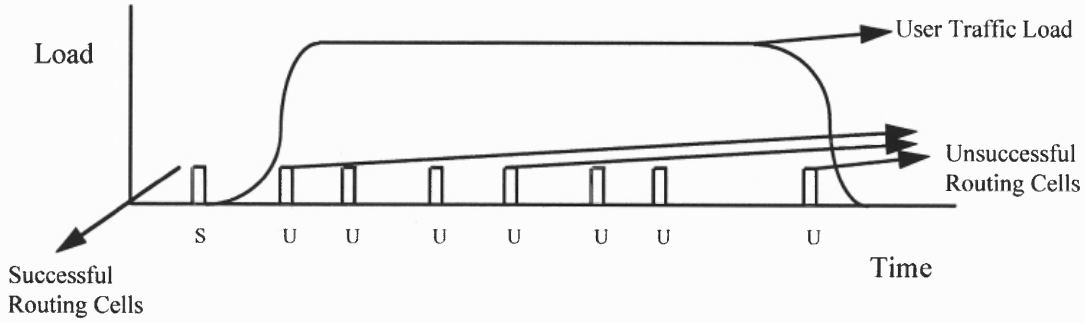

Figure 2.4 Link Load Due to Information and Routing Cells as a Function of Time

The effect of the saturation routing load in a particular link depends on whether the routing is successful or not. Whereby, a link that only transmits routing cells for successful routing efforts will be impacted the least (as the amount of information carried increases) and a link that only transmits routing cells for unsuccessful routing efforts will be impacted the most (as the amount of information carried is zero). For the moment let's make the assumption, that all routing efforts are successful (that indeed there is an acceptable route for every routing request); that is for each routing effort, there is a path that will carry user load. Now, let's assume that there are a large number of successful routing requests, M, made over a long period of time. One can proceed to calculate the average load effect as follows. Add the load of all routing cells generated by the M routing efforts for all links in the entire network (defined as L_{TOT}), and divide that number by the user load generated in all links because of the M routing requests. The latter is equal to M times the average number of cells generated by data exchanges, defined as $\bar{I}$, times the average number of hops, defined as $\bar{S}$. This ratio over long times is a good estimate of the value of the saturation routing factor, f (previously defined as the ratio of R cells over I cells).

$$E[f] = \frac{\sum_{m=1}^{m=M} \sum_{links} routing\_cells}{\sum_{m=1}^{m=M} \sum_{links} user\_cells} \leq \frac{M \cdot 2 \cdot L_{TOT}}{M \cdot \bar{I} \cdot \bar{S}} = \frac{2 \cdot L_{TOT}}{\bar{I} \cdot \bar{S}} \qquad (2\text{-}2)$$

The inequality of the expression (2-2) is as a result of maximizing the numerator and minimizing the denominator. The maximum numerator is as a result of the fact that the maximum number of routing cells generated by a routing effort is equal to twice the number of links (L_{TOT}) (Refer to inequality (2-1)). Since only significant data exchange requirements will invoke the use of saturation routing, then it follows that f is less than or equal to the ratio of a single routing effort over which only the minimum required of information cells and the maximum number of routing cells are exchanged.

Let's examine this ratio for a single routing effort. Because the least number of links that data traverses when there is a successful routing effort is equal to one, then the total number of traffic cells generated in a network for a successful routing request is at least the number of cells generated by the user. It follows that f, previously defined as the ratio of routing cells generated by a single routing effort over the number of information cells generated by the user, is greater than the average factor by which the link is affected by the presence of the routing effort.

$$E[f] \leq \frac{2 \cdot L_{TOT}}{\bar{I} \cdot \bar{S}} \leq \frac{2 \cdot L_{TOT}}{\bar{I}} \equiv f_{\max} \qquad (2\text{-}3)$$

The above expression (2-3) is an average impact for all of the links, if all of the routing efforts are successful. However, some links will be affected more than others. Fortunately, E[f] will be greater for those links that are utilized the least (that is over those links which do not have any successful routing efforts), and E[f] will be lesser for

those links that have the most number of successful routing efforts. Therefore, the use of f_{max} is justified since for the group of links of interest (those with the greatest load), f will be characterized by values less than f_{max} (i.e., this is a worst case assumption). The other assumption to be relaxed is the fact that not all routing efforts will be successful (i.e., a path from source to destination is not possible). This is most likely to happen when the network conditions are near the saturation point. The effect of routing cells on almost saturated links can be controlled by not allowing routing cells to traverse a link, if the link conditions are near saturation, by implementing a method such as described in [10].

Basically, the entire loading produced by a routing effort will be assigned to each of the links of the path selected. Note that there are 9 routing cells generated in the example discussed (as shown by the left portion of Figure 2.5). Two cells are generated at step 1, five cells are generated at step 2, and two cells are generated at step 3. It will assumed that all 9 cells are assigned to each of the links of the path selected (as shown by the right portion of Figure 2.5). This is done to make the analysis tractable. The objective is to demonstrate, in this analysis, that even with this worst condition assumption (i.e., each of the generated routing cells for this routing effort traverse the selected path), the effect of saturation routing cells is small. The only portion to be justified is that some of the links that are affected by the routing effort (e.g., the link between node A1 and node A3, etc.) are assumed unaffected. Here in this analysis, it is assumed a uniform network with respect to links, sources, and destinations (Refer to Assumption 1). Therefore, if there is no preference of one link over any other, then this

assignment of load is averaged out as routing decisions are made over those unaffected links (e.g., the link between node A1 and node A3).

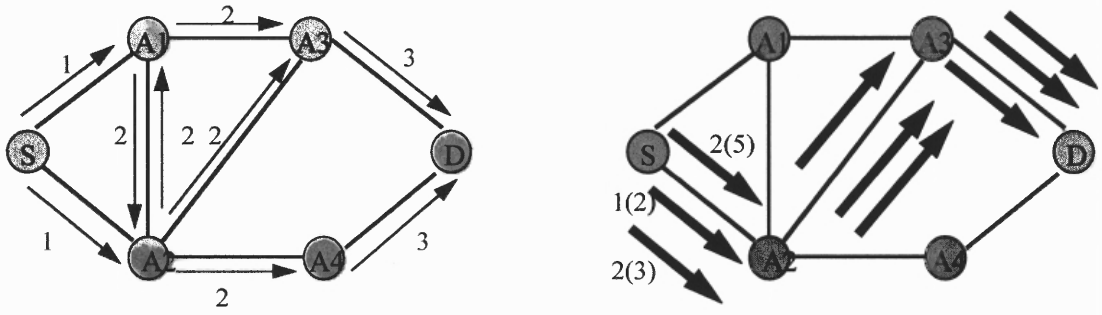

Figure 2.5 Actual Versus Analyzed Routing Effort Loading

By recalling inequality (2-1) (i.e., the number of routing cells generated by a routing effort is at most twice the number of links in the network), the task of calculating the number of routing cells for a virtual circuit setup thus can be converted to the task of computing (or bounding) the number of links in an ATM network. The latter is, in general, a daunting task; however, a reliable estimate will be computed in the subsection below for non-pathological networks (i.e., Number of Links Versus Path Size Subsection).

2.2.6 Number of Links Versus Path Size

The number of links in a given network is topology dependent. However, in general, the more typical large packet data networks employ a relatively small average number of links per node, L, when compared to the number of nodes, N, in the network. For example, the network with the largest number of links is a fully connected network (i.e., one that has N-1 links per each node, where N is the number of nodes) with $N \times (N-1) / 2$ links. Whereas, a star-network will only have (N-1) links. Absolute and relative

connectivity will be defined as parameters, C_A and C_R. They, respectively, describe the average number of links, which characterize a particular network in the absolute sense (i.e., the total number of links, L_{TOT}) and in the relative sense (i.e., the ratio of L_{TOT} to N, providing an average number of links per node, L). However, the more typical large packet data networks show a relatively small average number of links per node, L, when compared to N. As a reference, a table that indicates the total number of links in a network, for different network sizes and topologies is included in Table 2.2.

The number of links in the network needs to be estimated. In previous usage in the literature, the average path length, $\bar{S}$, in a general network was estimated as approximately equal to or smaller than $Log_2 N$. This is as reported in [12] and documented for the Internet Network in 1991 [19], in which a mean Autonomous System distance of 5 is reported, the equivalent of the shortest path, for an Internet with 59 Autonomous Systems, the equivalent of nodes.

Table 2.2 Connectivity, Relative and Absolute, as a Function of Network Topology

Number of Nodes [N]	*Fully Connected Network*		*# Links Per Node,* $L = \sqrt{N}$		*# Links Per Node,* $L = Log_2 N$	
10	45	9	15	3	20	4
100	4,950	99	500	10	350	7
200	9,950	199	1,400	14	800	8
500	124,750	499	5,500	22	2,250	9
1000	499,500	999	15,500	31	5,000	10

2.2.7 Estimates of Overhead Factor Using Simulation

To determine a general relationship between average path length, number of links, and number of nodes is a difficult task because it is strongly topology dependent, and thus analytically challenging. So, here, simulation experimentation was employed to generate random generic network topologies. This is to derive a functional relation numerically over the parameter ranges of interest. A simulation program, the Random Topology Link (RTL) simulation, built for this objective, was used to generate random topologies with a given number of nodes, N, and a given number of Links, L, per node. These results, essentially, represent a statistical type estimation of the relationship of average path length as a function of N and L. Ten thousand runs of the simulation program were carried out for each selection of N and L. The RTL yielded the results shown in Figure 2.6.

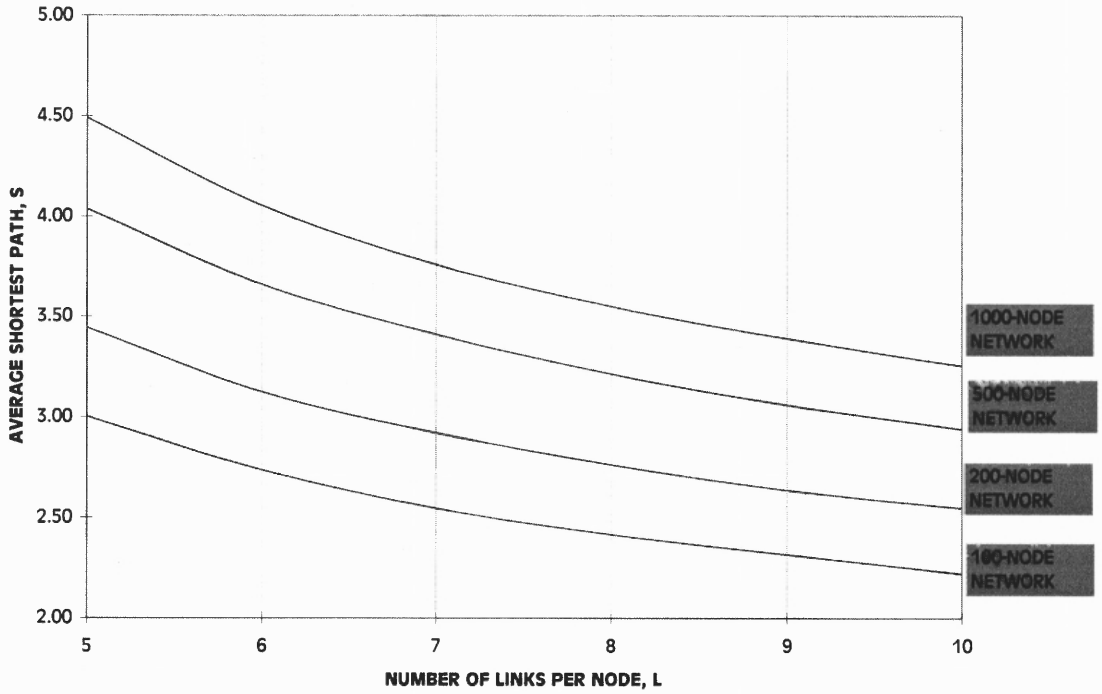

Figure 2.6 Average Shortest Path as a Function of Number of Links Per Node and Network Size

The way that the random topologies were generated is by assigning a number of links, L, from the first node N_0 to a given set of nodes N_1 through N_L. After these assignments were made, each node was assigned its corresponding number of links. These links either connected to already existing nodes at the same level (e.g., N_1 to N_2) or connected to a newly generated node N_{L+1}. This was repeated until the specified number of nodes and links was generated. In addition, a special experiment was conducted for the 1000-node network. The simulation was modified to provide statistically varied topologies, in an attempt to remove the artificial assumption that each node has a fixed number of links. The number of links was assigned from a uniform random distribution

with a lower bound equal to L-2 and an upper bound equal to L+2. Results are shown in Figure 2.7.

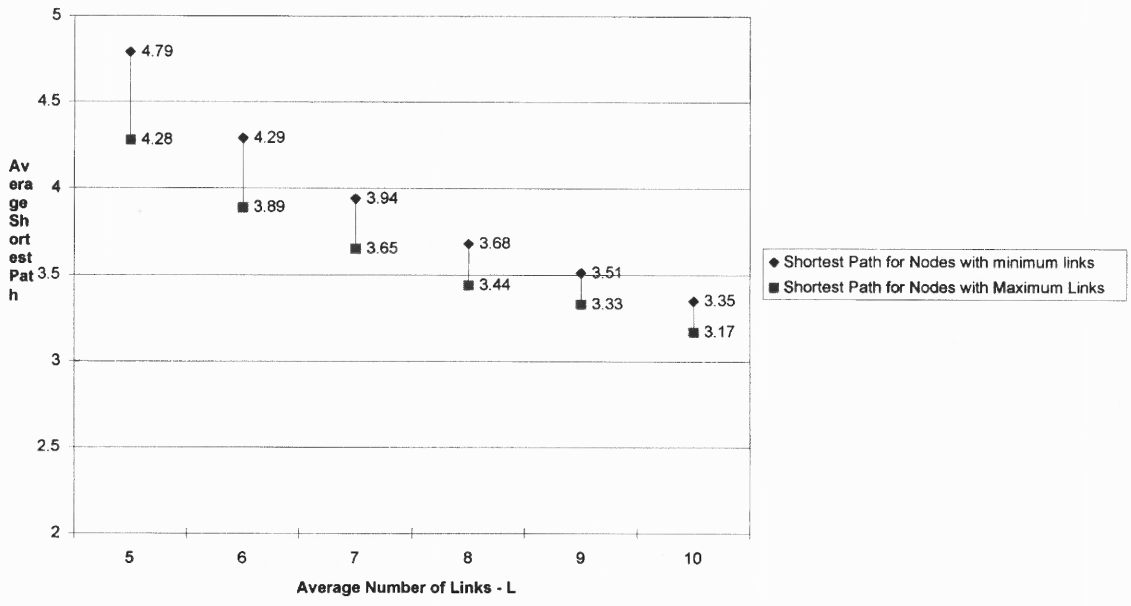

Figure 2.7 Average Shortest Path For a 1000 Node Network Having Nodes with an Average Number of Links – L

2.2.8 Impact of Routing Overhead in an M/D/1 Queue Model

Under some to be determined constraints, the aim is to prove that the added load caused by the saturation algorithm does not significantly degrade the performance of the network. To calculate the performance degradation, the cell information arrival process will be modeled as an independent one with an exponential distribution. The use of this assumption can be justified by the Kleinrock Independence Approximation [14]. To include routing cells in the analysis, the following modeling assumption will be made: the routing cell arrival process is statistically independent of the information cell arrival

process. Although routing cells and information cells are mutually exclusive for a single data exchange, this assumption can be made because cells for all data exchanges are being considered.

Therefore, the total cell arrival process is also an exponential one with a rate equal to the sum of the rate of the information cell process, λ cells per second, and the rate of the routing cell process, ($f\lambda$). Following an M/D/1 queuing analysis [20 and 3], one can conclude that the average waiting time with saturation routing, $E_s(t)$, and without saturation routing, $E_{ns}(t)$, is, respectively:

$$E_s(t) = \frac{(1 - \frac{\rho_s}{2})}{\mu \times (1 - \rho_s)}, \qquad E_{ns}(t) = \frac{(1 - \frac{\rho_{ns}}{2})}{\mu \times (1 - \rho_{ns})} \tag{2-4}$$

where:

μ is the average service rate

ρ_s , ρ_{ns} are the utilization factors with and without saturation, respectively equal, in this case, to $\frac{(\lambda + f \times \lambda)}{\mu}$ and $\frac{\lambda}{\mu}$, respectively

Comparing the average waiting times with and without saturation routing, then one can conclude that the routing cell impact is minimal as long as $E_s(t) \approx E_{ns}(t)$, which reduces to:

$$\rho_s < 1; \tag{2-5a}$$

$$f << 1; \tag{2-5b}$$

$$f << (\mu - \lambda)/\lambda; \tag{2-5c}$$

The first relation, shown in expression (2-5a), requires that some bandwidth has to be reserved for the transmission of routing cells in addition to the bandwidth for data

transmission. The next relation, shown in expression (2-5b), requires that the number of routing cells traversing a link have to be significantly smaller than the number of information cells. This is the most restrictive requirement for utilization values less than 0.9. And, the last relation, shown in expression (2-5c), requires in physical terms that the amount of bandwidth consumed by the routing overhead (i.e., equal to $(\lambda \times f)/\mu$) has to be significantly less than the unallocated bandwidth (i.e., $(\mu - \lambda)/\mu$). This is the most restrictive requirement for utilization values greater than 0.9.

To interpret this result numerically, let's include the requirements on f assuming some utilization factor values (i.e., $\rho = \lambda/\mu$) (refer to Table 2.3). Even at utilization levels as high as 0.9, which is a relatively high utilization for a statistical multiplexer such as an ATM switch, the impact of the saturation algorithm is negligible, as long as f is at most 0.011. Let's recall, that in the example estimates of f, for the worst case network, it was found to be equal to 0.01. This shows that saturation routing is an effective algorithm for substantial data exchanges over an ATM network.

Table 2.3 Requirement 3(c) for f, Where << is Approximated to Mean Smaller by at Least a Factor of 10

ρ	$(\mu-\lambda)/\lambda$	$f\ (\leq)$
0.5	1.0	0.1
0.66	0.5	0.05
0.75	0.33	0.033
0.90	0.11	0.011

However, if higher utilization rates are desired, the saturation routing cells start having a greater impact on the expected delay because of the exponential behavior of the

model at utilization rates close to 1. Furthermore, if one could increase the bandwidth, or virtually allocate part of the bandwidth to routing (as it is done in out-of-band signaling techniques), then the impact is minimum. If one increases the bandwidth by the same factor f (defined as the ratio of saturation routing cells to information cells), one can determine that performance can be enhanced by a factor equal to the inverse of f plus 1.

$$E_{sb}(t) = \frac{\frac{1}{\mu(1+f)} \times (1 - \frac{\lambda(1+f)}{2\mu(1+f)})}{(1 - \frac{\lambda(1+f)}{\mu(1+f)})} \tag{2-6}$$

where $E_{sb}(t)$ is the average waiting time with saturation routing and bandwidth increase

Simplifying,

$$E_{sb}(t) = \frac{\frac{1}{\mu(1+f)} \times (1 - \frac{\lambda}{2\mu})}{(1 - \frac{\lambda}{\mu})} \tag{2-7}$$

from which follows

$$E_{sb}(t) = \frac{E_{ns}(t)}{(1+f)} \tag{2-8}$$

for low values of f, as calculated before (i.e., 0.01), then the bandwidth increased required will be only 1% and the overall effect will be a reduction of 1% of the expected time delay.

2.3 Comparison of Routing Overhead

To compare routing overhead between PNNI and Saturation Routing, a Symmetrical Hexagonal Hierarchical Network (SHHN) will be defined as a sample network. This network has the property that each level looks like the basic level. By using the SHHN, one can proceed to calculate the overhead in a PNNI network as a function of the number of nodes and/or levels. In addition, the benefit obtained by selecting paths with Saturation Routing over the conventional PNNI routing will be calculated.

Basic properties of the SHHN

There are 7 physical level peer groups. Each peer group has 7 nodes, with the center node as a Peer Group Leader (PGL). The PGL has a horizontal link to each of the other 6 nodes. Each of the 6 nodes has 3 horizontal links. Three of these 6 nodes have outside links to adjacent peer groups. The exception is the center peer group, where the 6 nodes each have 1 outside link. The Basic Laydown is depicted in Figure 2.8 (just one level of hierarchy). The next level of hierarchy is depicted in Figure 2.9 (2 levels of hierarchy), the next level of hierarchy is shown in Figure 2.10 (3 levels of hierarchy), and the fourth level of the hierarchy is depicted in Figure 2.11. (i.e., 4 levels of Hierarchy).

One level up

There is one higher level peer group that looks exactly like each of the lowest level peer groups. It has 7 Logical Group Nodes (LGN), each representing a lowest level peer group. Refer to Figure 2.9.

Two levels up

Similar to the one level up, the two levels up consist of a peer group that is identical to the peer groups of level 2.

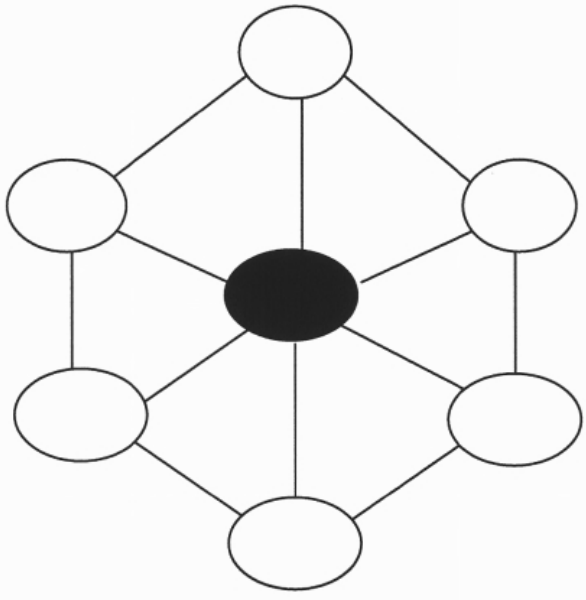

Figure 2.8 SHHN at the Basic Hierarchical Level (i.e., 1 Level of Hierarchy)

Assumptions

1) Network is in steady state, i.e., all topology databases are synchronized.

2) Virtual Path Channels (VPC) have already been established between PGLs of adjacent peer groups.

3) Nodes are static, i.e., there is no reason for a node to change a Private Network-Node Interface (PNNI) Topology State Element (PTSE) it has originated.

4) There are no errors on the links.

The basic overhead associated with the routing protocol is PTSE (embedded in PTSPs), PTSE acks, and Hello PNNI Packets. In general, PTSE flow from one node to the other one, whereas PTSE acks flow on the opposite direction. The effects of PTSE acks will be ignored and PTSE flow in one direction will be assumed. A calculation of the overhead associated with PNNI Hello Packets and PTSPs is included below.

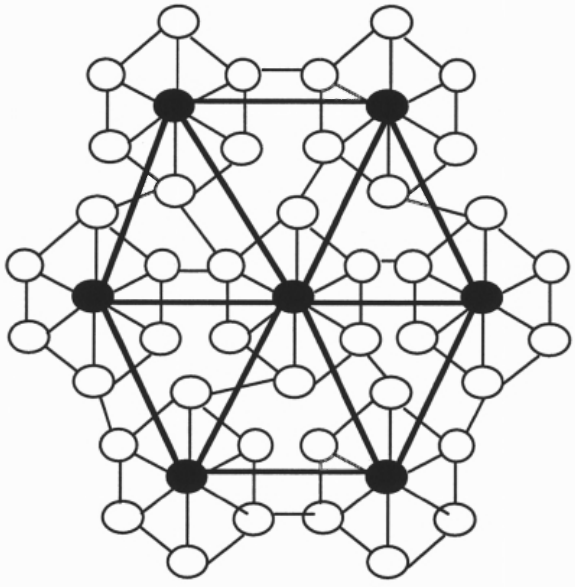

Figure 2.9 SHHN with 2 Levels of Hierarchy

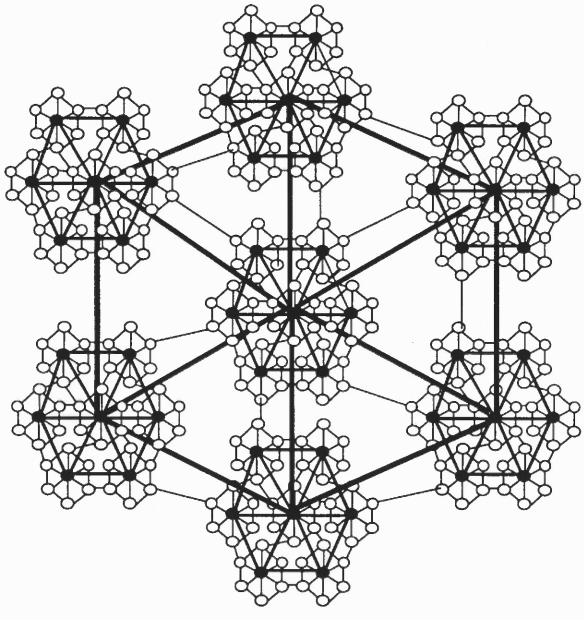

Figure 2.10 SHHN with 3 Levels of Hierarchy

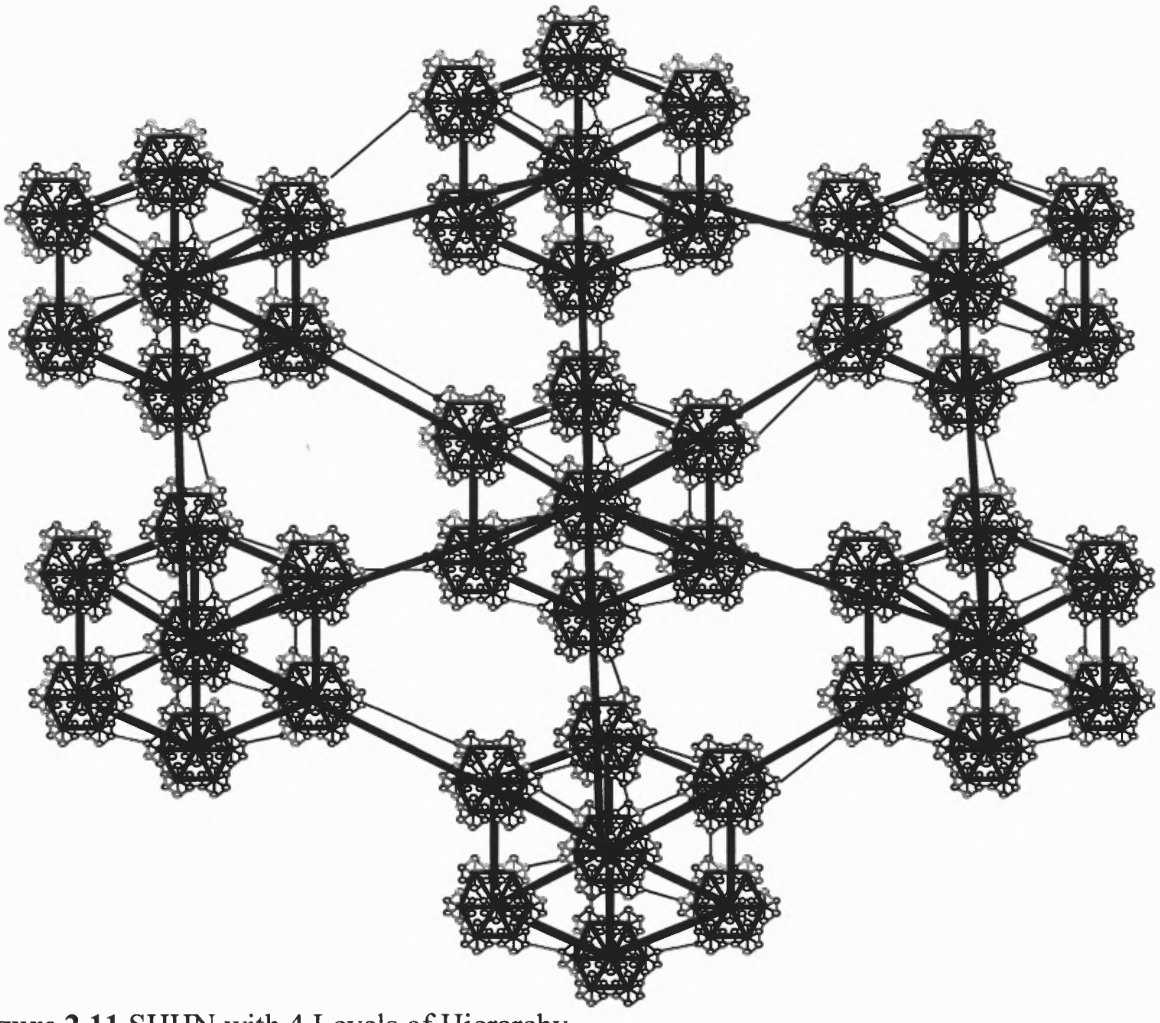

Figure 2.11 SHHN with 4 Levels of Hierarchy

PNNI hello packet overhead

The length of PNNI Hello Packets is as follows:

$$\begin{aligned} Hello\_Length = {} & Hello\_Hdr + Aggregation\_Token\_Ig + \\ & Nodal\_Hierarchy\_List\_Ig + UpLinkIg + \\ & LGN\_Horizontal\_Link\_Extension\_Ig \end{aligned} \tag{2-9}$$

Where Hello_Hdr= 100 bytes, Aggregation_Token_Ig= 8 bytes,

Nodal_Hierarchy_List=12+56*L, where L is the number of Hierarchical Levels above the LGN

UpLinkgIg=8+44*C, where C is the number of service categories (assumed to be 4)

LGN_Horizontal_Link_Extension_Ig =8+12H, where H is the number of Horizontal links between LGNs.

The worst case is the links that are across a Peer Group Boundary, they will see the following Hello Overhead (Refer to Table 2.4)

Table 2.4 Overhead due to Hello Packets (in Bytes)

	HELLO OVERHEAD
Hierarchy Level 1	528
Hierarchy Level 2	120
Hierarchy Level 3	120
Hierarchy Level 4	120
TOTAL	**888**

PTSP overhead

Under these assumptions, there are only two reasons to send a PNNI Topology State Packet (PTSP):

1) re-origination of a new instance of a self-originated PTSE (the default value is 30 minutes),

2) flooding of a received non-self-originated PTSE.

Within a refresh interval

Part 1) All nodes send self-originated PTSEs on each attached physical link. Thus all nodes in a peer group have identical topological databases. At the lowest hierarchical level, one would expect 7 PTSEs (one for the PGL and six for its LGN Peers).

Part 2) Each PGL, representing an LGN in the next higher level peer group, originates a PTSE containing summarized topology information, and floods it to its (LGN) peers. This takes place on VPCs via border nodes. In this way each LGN receives information about child peer groups.

Part 3) Each LGN distributes the summarized topology information it receives via flooding from other LGNs to its child peer group. In this way, each lowest level node gets a view of the higher levels into which it is being aggregated.

Counting PTSPs on a link

Note: Each PTSE will be assumed to be sent in a separate PTSP. Ordinarily with user traffic present, PTSEs may be queued for later delivery. This assumption is made because other traffic assumptions would be needed to estimate queuing, and because bundling of PTSEs is not standardized.

The general formula for PTSP length is:

$$PTSP\_length = PTSP\_hdr + PTSE\_hdr * (\#\_of\_PTSEs) + Total\_Payload\_PTSEs \quad (2\text{-}10)$$

Since only one PTSE in a PTSP is being considered, this formula becomes:

$$PTSP\_length = 44 + 20 + Total\_Payload\_PTSEs \quad (2\text{-}11)$$

where 44 bytes is the PTSP_hdr and 20 bytes is the PTSE_hdr.

The total payload in PTSEs is calculated as follows:

$$Total\_Payload\_PTSEs = Nodal\_State\_Parameter\_IG + Nodal\_IG + IRA + ERA + H\_Links\_IG + Up\_Links\_IG \quad (2\text{-}12)$$

where:

Nodal_State_Parameter=(16+44C)*f(P), where C is the number of Service Categories and f(P) is the number of nodal state parameters. That is f(P)=0 for lowest level and 6 for higher levels for this network.

Nodal_IG= 48+D, where D=84 if there is a higher level of hierarchy, otherwise is 0

IRA=((16+2*44C)+A*(Number_of_Prefixes))*Number_of_IRAs_Igs

ERA=((23+2*44C+N)+B*(Number_of_Prefixes))*Number_of_ERAs_Igs

H_Links=(40+44C)*(#_of_Horizontal_Links)

Up_Links_IG=(72+44C+8+44C)*(#_of_Up_Links)

Assuming the following values,

C=4, A=B=20 (worst case), Number_of_Prefixes=4 (for both ERAs and IRAs), and Number_of_IRA_Igs and Number_of_ERA_Igs=4, #_of_Horizontal_Links=6 for PGL and 3 for the Border Nodes, #_of_Up_Links=0 for PGL and approaches an average of 1 (included induced uplinks) for Border Nodes.

Table 2.5 describes the total overhead due to PTSPs at the different hierarchical levels.

Table 2.5 Overhead due to PTSP Generated at All Levels (in Bytes)

	First-level		Mid-Level		Last Level	
	border node	PGL node	border node	PGL node	border node	PGL node
PTSP Header	44	44	44	44	44	44
PTSE Header	20	20	20	20	20	20
Nodal-State Parameter	0	0	1152	1152	1152	1152
Nodal-IG	132	132	132	132	0	0
IRA-IG	1792	1792	1792	1792	1792	1792
ERA-IG	1900	1900	1900	1900	1900	1900
HOR. LINKS IG	648	1296	648	1296	648	1296
UP. LINKS IG	1656	0	1656	0	1656	0
TOTAL	**6192**	**5184**	**7344**	**6336**	**7212**	**6204**

Assuming that each link will receive the PTSPs from each node at the hierarchy (i.e., 7 nodes at hierarchy level 1, 7 nodes at hierarchy level 2, and so forth). The total number of bytes generated by this network because of PTSPs is 192,612 bytes per refresh period.

As a result, the load applied to the links for this sample case is shown in Table 2.6. In this illustration, over 10,000 bps are used to maintain the routing tables. This is in the absence of any significant change in resources. This 10,000 bps consumed in one direction in each link allow for approximately 25 routing requests (using Saturation

Routing) per second in the entire network. This is just to maintain an equivalent level of overhead between PNNI and Saturation Routing. The next subsection will quantify the benefits of using saturation routing.

Table 2.6 Overhead in PNNI Routing for the SHHN

	PTSP OverHead	HELLO Overhead	TOTAL
Overhead Per Period	192612.00	888.00	
Period	1800.00	15.00	
Bytes Per Second	856.05	473.60	1329.65
Bits Per Second			**10637.23**

Differences in paths selected between PNNI and saturation

Paths were calculated using both routing techniques (i.e., the PNNI and Saturation Routing) for the SHHN. These paths were calculated between every possible source-destination pair in the network. The results for the two-level up hierarchy network and the three level up hierarchy network are presented below.

The difference in path distance between the Saturation Routing and PNNI for the 3 levels of hierarchy network is included in Table 2.7. The difference is shown as a function of the lowest common level between source and destination. As expected, there is no difference between PNNI and Saturation routing whenever the source and destination belong in the same Peer Group. As the common hierarchical level increases, the distance between Saturation and PNNI also increases. This is also expected because PNNI information of the network becomes less as the intended destination is further removed. The overall improvement is higher than 8%.

Table 2.7 Average Distance Difference between Saturation and PNNI for the 3-Level Hierarchy as a Function of Source and Destination Locations

CASES	%-IMPROVEMENT
SAME-LEVEL-3	8.57%
SAME-LEVEL-2	3.56%
SAME-HIERARCHICAL-LEVEL	0.00%
ALL	**8.31%**

In addition, the average difference provided by PNNI and Saturation Routing for the 3-Level Hierarchy as a function of the Path Distance (as measured by the shortest path) is provided. As shown by Figure 2.12, the Saturation Routing algorithm provides the most difference when the nodes are at medium distances. That is the same distance that shows the highest number of paths. Also, Saturation Routing provides in the average paths that are 1.14 links shorter than PNNI.

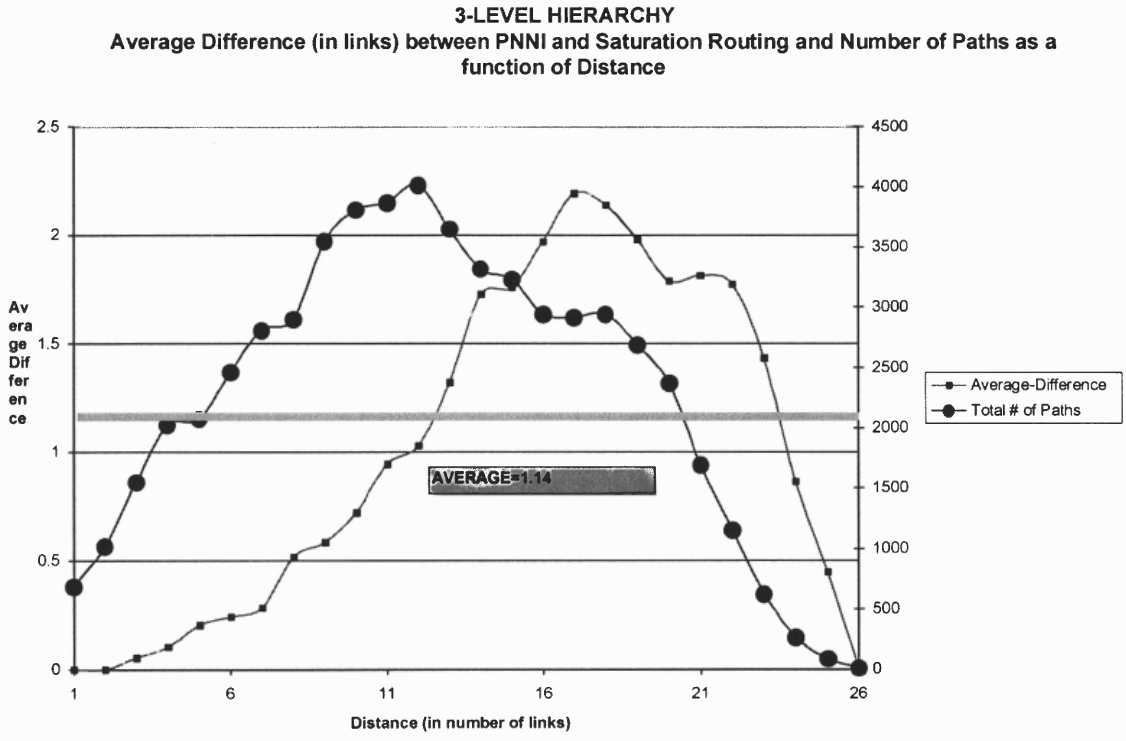

Figure 2.12 Average Difference (in links) between PNNI and Saturation Routing and Number of Paths as a function of Distance for the 3-Level Hierarchy

The difference in path distance between the Saturation Routing and PNNI for the 4 levels of hierarchy network is included in Table 2.8. The difference is shown as a function of the lowest common level between source and destination. As expected, there is no difference between PNNI and Saturation routing whenever the source and destination belong in the same Peer Group. As the common hierarchical level increases, the distance between Saturation and PNNI also increases. This is also expected because PNNI information of the network becomes less as the intended destination is further removed. The overall improvement is higher than 8%.

Table 2.8 Average Distance Difference between Saturation and PNNI for the 4-Level Hierarchy as a Function of Source and Destination Locations

CASES	%-IMPROVEMENT
SAME-LEVEL-4	18.67%
SAME-LEVEL-3	9.60%
SAME-LEVEL-2	3.58%
SAME HIERARCHICAL LEVEL	0.00%
TOTAL	**8.74%**

In addition, the average difference provided by PNNI and Saturation Routing for the 4-Level Hierarchy as a function of the Path Distance (as measured by the shortest path) is provided. As shown in Figure 2.13, the Saturation Routing algorithm provides the most difference when the nodes are at medium distances. That is the same distance that shows the highest number of paths. Also, Saturation Routing provides in the average paths that are 6.4 links shorter than PNNI.

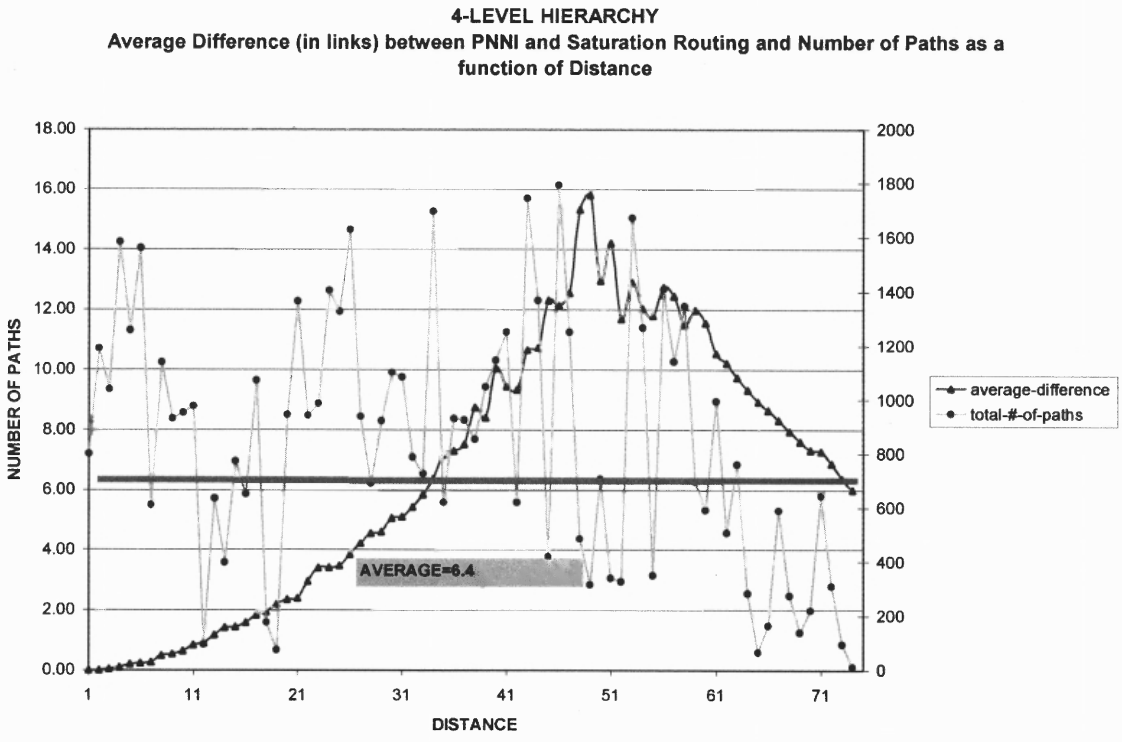

Figure 2.13 Average Difference (in links) between PNNI and Saturation Routing and Number of Paths as a function of Distance for the 4-Level Hierarchy

The level of improvement experienced increases with the number of hierarchical levels. The advantages of using Saturation Routing can be quantified using the SHHN. In the 3-level hierarchy, an average of 1.14-link shorter paths was shown. If one were to allocate a total sum of 155 Mbps (assuming the link connections are OC-3s) distributed over all of the links in the 3-level hierarchical network for Saturation Routing, this will become a more efficient network. In addition, Saturation Routing was demonstrated to be 8% more efficient in path selection. Again, using as an example a typical OC-3 connection, this will allow an allocation of over 7 Mbps for Saturation Routing (approximately 5% of the total maximum link capacity) and still make the overall network more efficient.

In the case of the 4-Level Hierarchical Network, similar conclusions can be drawn. In this network, an average of 6.4-link shorter paths was shown. This will allow an allocation of over 900 Mbps in the entire network for Saturation Routing. The extra capacity is needed for the increased number of links and nodes. The efficiency shown is higher than the 3-level hierarchical network, but it is still around 8%. Similar to the 3-level hierarchical network, this will allow an allocation of over 7 Mbps for Saturation Routing (approximately 5% of the total OC-3 Connection) and still make the overall network more efficient.

2.4 Hybrid Routing Approach

Now that the advantages gained by using the Saturation Routing Algorithm have been illustrated using the SHHN, the concept of a Hybrid Routing Approach will be introduced. This approach combines the benefits of both PNNI and Saturation Routing. The concept consists of a 3-way alternative for each routing effort:

- For small exchanges, the PNNI normal routing effort will be executed.
- For medium exchanges, a hierarchical Saturation Routing effort will be executed.
- For large exchanges, a full Saturation Routing effort will be executed.

The first and last type of routing effort has been described before. A description of the second type of routing effort follows next. The saturation routing process will take place at each level of the hierarchy. In the first level of the hierarchy, all of the nodes will become aware of the routing effort. In the second level of the hierarchy, only the nodes that are Peer Group Leaders (PGLs) will receive information of the routing effort. As the common level between source and destination is efficiently found through the

saturation routing process, the common (for source and destination) lowest level PGL will direct the saturation routing to take effect at those levels only and inhibit the saturation routing to propagate any further from those levels.

The advantages of this proposed approach are the following:

- This hybrid routing approach can be combined with the existing PNNI routing infrastructure.
- If the exchange is small in nature, the setup of the path will not incur any extra overhead.
- If the exchange is large, the potential for substantial path efficiency warrants the expense of searching for the most optimum path. That is routing overhead is spent so that resource savings can be capitalized by using the most efficient paths.
- If the exchange is medium, one is willing to incur the deficiency of PNNI, improved by the fact that one will be attempting every path known by the source PNNI node. That is the extra expense of executing Saturation Routing is minimal as well as the potential for savings.

Boundary Lines 1 and 2 of Figure 2.3 can be used as delineators for small, medium, and large user exchanges.

CHAPTER 3

MEASURES OF PERFORMANCE USED BY THE SATURATION ROUTING ALGORITHM PATH DECISION MECHANISM

The Measures of Performance that being proposed for use in the Saturation Routing Algorithm decision mechanism are the ones, for the most part, used and defined in the User to Network Interface specification. They are defined below for completeness:

- Peak Cell Rate (PCR) – This is the maximum number of cells that will be allowed to be transmitted over a unit of time. The saturation routing will check the overall sum of PCR that a link has committed to – the higher the value relative to the maximum capacity of the link, the less desirable the link becomes.
- Cell Delay Variation Tolerance (CDVT) – the amount of clumping that can be tolerated for PCR. The saturation routing will also check this value. The higher the value CDVT has, the less desirable the link becomes.
- Sustainable Cell Rate (SCR) – a rate that represents the average number of cells exchanged in a session. The saturation routing will also check this value. The higher the value CDVT has, the less desirable the link becomes.
- Maximum Burst Size (MBS) – The number of cells on a virtual circuit that can burst at the PCR.
- Minimum Cell Rate (MCR) – the minimum rate at which a host using the available bit rate service category will always be able to transmit data.

In addition, other Quality of Service (QOS) parameters will be taken in the setup message:

- Cell Loss Ratio (CLR): ratio of lost cells versus attempted cells.
- Cell Transfer Delay (CTD): measure of time required to deliver the cells
- Cell Delay Variation (CDV): measure of how latency varies between the arrival of one cell to another.

Besides the traditional measures of performance, a new measure of performance is recommended for the implementation of the Saturation Routing Algorithm. This is described below.

3.1 Burst Voice Arrival Lag as a Measure of Performance

This section shows that the Burst Voice Arrival Lag (BVAL) can be used as a measure of performance to select or distinguish the level of service provided by systems that delivers voice packets over ATM networks. This measure of performance can be applied for any system that sends voice packets (or cells) and also can be used by a routing algorithm, such saturation routing, to select paths between source and destination. The typical measure of performance used to distinguish the areas for which voice performance is still considered acceptable is packet (or cell) loss ratio. The acceptable range, in terms of packet loss, for a given system depends on the packet length and the voice encoding scheme used in the system. In such systems, the packet losses for this threshold are assumed to occur randomly and evenly distributed. In other words, the errors do not show burstiness. However, it is known that the perceptual quality of burst packet errors is much worse than that of an equal number of packet losses distributed randomly across

the packet stream. To take into account the effects of unevenly distributed errors that may occur, the ITU has imposed some other requirements at the Physical Layer in terms of Degraded Minutes, Severely Errored Seconds, and Errored Seconds. This is an attempt to measure long term average losses as well as to capture bursty errors and completely error free seconds.

In a packet network it is difficult to characterize the performance of the system for voice transport. Packet error rates (or cell error rates) can be measured; but typically the errors have been found not to occur randomly. In fact, when congestion occurs in a network, packets conveying voice information may either be lost, or arrive too late, in bursts. Therefore, there is a need to have a Measure of Performance that takes into account not only of the packet error rate, but that also provides for the burstiness, or lack of, in the system. It will be shown in this paper that the Burst Voice Arrival Lag (BVAL) not only measures the packet error rate, but it also penalizes the packet errors when their nature is bursty.

3.1.1 Description of a Packet Arrival Process In a Packetized Voice System

In a packetized voice system, typically there is a delay variation requirement that originates from the continuous stream of data that this type of application generates. The application at the destination end implements a receiver buffer that has as a purpose to "dejitter" or smooth out the delay variation presented by the network, such that a constant stream with a fixed constant delay is presented to the destination application. For example these playback buffers are used in RTP, a protocol that is used for real-time applications over IP [17]. The size of the buffer at the receiver depends on the maximum

tolerable delay for the given application (approximately 500 milliseconds for round trip voice delay, or a 250-millisecond one-way voice delay, [13]). The "dejitter" buffer is used to store the voice information as it arrives and then it is played back to the destination application at a constant delay. If packets arrive too late (i.e., with a delay greater than the maximum tolerable delay), then they are discarded without being played back.

3.1.2 Burst Voice Arrival Lag (BVAL) Description

In this section, the BVAL Measure of Performance will be defined. After that, the BVAL value for a system with random packet losses (defined as System A) will be calculated. That is followed with a calculation of BVAL for a system that shows burst packet losses with a fixed size N (defined as System B). It will then be demonstrated that the BVAL is higher for System B than it is for System A, although they have the same packet error rate.

BVAL for a system with random packet losses

It will be shown that for a system with a given packet generation rate and random packet losses, BVAL is a measure that depends on the packet loss rate, which is the current measure of performance used to evaluate the systems that carry voice over packet networks. The process will be illustrated by showing packet Arrival at a destination (refer to Figure 3.1). The upper portion of Figure 3.1 represents the transmitter generating packets at a given rate (depicted here by the generation period, defined as r). The lower portion of Figure 3.1 represents data being played back to the destination with a fixed delay of the maximum tolerable delay. Note that 2 cells (the ones generated at 2r

and 5r) are not played back. They could have been lost in the network or delayed by a value greater than the maximum tolerable delay (and therefore discarded by the application at the destination). The lifetime of the packet, L, measures the time from which the last packet to be played back started playing back. This is in essence, the amount of time that the play back queue has starved (i.e., it has no data to play back) plus the playback time of a voice packet. In a perfect system, the L will increase from zero (at the time a packet arrives) to the maximum value r (when the next packet starts playing back). In the Figure 3. 1. illustration, the L reaches the value 2r in two occasions because two nonconsecutive packets were lost. In these examples, the L had a value of 0 when the last packet started playing back (e.g., at time equal to r and at time equal to 4r). L increases its value from zero to 2r, time at which the next cell started playing back (i.e., cells that started playing back at time equal to 3r and 6r). Had the packets been lost consecutively (in a burst of 2-packet errors), L would have increased from zero to its maximum value 3r.

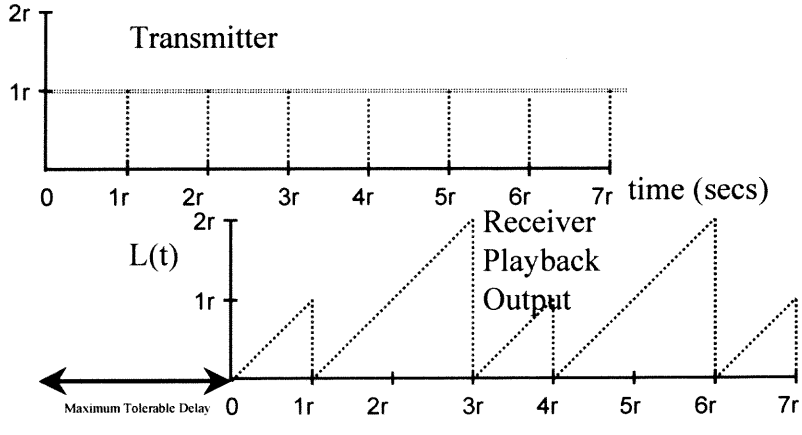

Figure 3.1 BVAL as a Function of Time for a System with Random Packet Losses

The following definitions apply in this calculation, referred to in Figure 3.1.

r - the source packet generation period, a constant for the system under analysis.

p - the probability of delivering a packet correctly from source to destination within the maximum tolerable delay time. For system A, it is assumed that the probability of delivering each packet is independent of whether the previous packet was delivered or not.

L - lifetime of playback buffer starvation plus cell playback time – a random variable that represents the time elapsed since the last packet finished playing back.

BVAL – Represents the area under the curve of L normalized with respect to the number of voice samples played back.

Figure 3.2 depicts the expected difference between BVAL for a system with random losses versus BVAL for a system with burst packet losses. In this figure, the solid line depicts the L for a bursty environment, whereas the bolded dotted line depicts the equivalent L on which the same number of packets are lost nonconsecutively. For example, the packets that were supposed to play back at times r, 2r, and 3r did not arrive or arrived late to be played back. This is a burst of 3 packets. If one compares the area under the curve for these 3 consecutively lost (or late arrival) packets versus 3 packets that are lost or arrived late nonconsecutively, one can determine that the difference in area is proportional to $(N-1)^2$, where N is defined as the burst size. The areas labeled A_1 and B_1 are equal to the areas labeled A_{-1} and B_{-1}, this leaves areas labeled 1, 2, 3, and 4 as the difference between the two arrival schemes in the case of 3 packets. As a result, in the first example in this figure, the difference in area is proportional to $4r^2$. In the second example, packets at time 5r and 6r did not arrive or arrived late. The difference in area, in this case, is proportional to r^2 because the burst size is equal to 2. Because BVAL is a normalized (not an absolute) value of the area, it will be determined that the difference in BVAL between the 2 systems is proportional to (N-1).

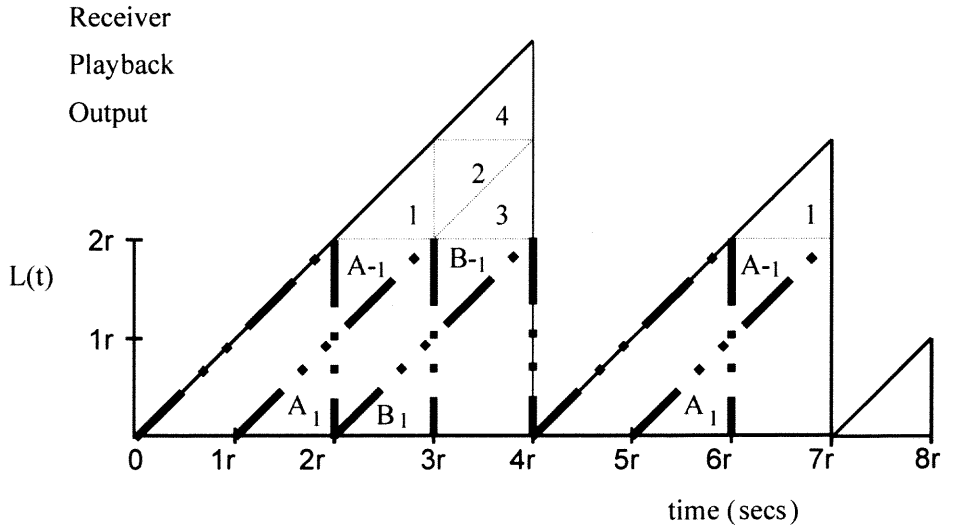

Figure 3.2 Difference in BVAL as a Function of Time for a System with Burst Packet Losses and a System with Random Packet Losses

A geometrical representation of BVAL is the area under the curve divided by the total number of packets played back in the calculation, V. Refer to equations 3-1 and 3-2.

$$\text{BVAL} = \lim_{V \longrightarrow \infty} \sum_{i=1}^{N} \left(\frac{L_i}{2}\right) \cdot L_i \tag{3-1}$$

$$\text{BVAL} = E[L^2] \tag{3-2}$$

Here V represents the number of voice packets received and played back for the measuring period. BVAL represents the sum of the areas of the triangle pieces divided by the number of voice packets, V, received. By taking the limit to infinity, assuming that the limit to infinity exists; which it does as long as $p > 0$. The probability, p, greater than zero, guarantees that at least one packet will be successfully received. This results in equation (3-2). It is also seen that to obtain the BVAL value, one must find the expected values of the packet life squared. The probability of successfully transmitting a

message can be represented as a Bernoulli trial process, whereby the probability of successfully delivering a packet is independent of previous outcomes (i.e., whether previous packets were successfully delivered or not). By using the Bernoulli trial process, one can readily find the expected value of the life of the report squared (See equation 3-3).

$$BVAL = E[L^2] = \sum_{k=1}^{\infty} k^2 \bullet r^2 \bullet p \bullet (1-p)^{k-1} = \frac{(2-p) \bullet r^2}{p^2} \qquad (3\text{-}3)$$

Thus, for a System A, the BVAL is a performance parameter that depends on the following: (1) the probability of successfully delivering a packet within the maximum tolerable delay (i.e., p), and (2) the source packet generation period (i.e., r). This derivation of BVAL is similar to the derivation of the Average Information Staleness (AIS). The AIS measure of performance was introduced in [16].

BVAL for a System with Bursty Packet Losses

In a System B (one that exhibits packet losses in bursts of size equal to N), when one is observing the packet arrival process there are two possible events. The first event that can be observed is that a good packet arrives. This event occurs with a probability of α. The second event that can observed is that a block of packet errors of size N arrives. This event occurs with a probability of β. The relationship of the defined packet completion rate, p, defined in the System A with respect to α and β, will be calculated. This relationship is shown in equations 3-4, 3-5, 3-6, 3-7, and 3-8.

Since,

$$p = \frac{E[good\_cells]}{E[good\_cells] + E[bad\_cells]} \tag{3-4}$$

$$p = \frac{\alpha}{\alpha + N * \beta} \tag{3-5}$$

One finds that,

$$1 = \alpha + \beta \tag{3-6}$$

$$p = \frac{\alpha}{\alpha + N * (1 - \alpha)} \tag{3-7}$$

$$\alpha = \frac{N * p}{1 + p * (N - 1)} \tag{3-8}$$

Again by using a Bernoulli trial process to find the BVAL of this system, one finds that the probability of a successful packet arrival, or the evidence of a burst of packet errors of size N (by not observing their arrival), is independent of the previous event (i.e., whether a packet arrived or a burst of packet errors was observed). To solve the BVAL for system B, the $E[L^2]$ needs to be calculated. Refer to equation 3-10.

$$E[L^2] = \frac{(2-\alpha)\cdot(N\cdot r)^2}{\alpha^2} + r^2(N-1)^2 - \frac{2r^2N\cdot(N-1)}{\alpha} \tag{3-10}$$

BVAL of System B expressed in terms of p is included in Equation 3-11.

$$BVAL = \frac{(2-p)\cdot r^2}{p^2} + r^2\cdot(\frac{1}{p} - 1)\cdot(N-1) \tag{3-11}$$

Now, one needs to determine if the BVAL of System B (defined in equation 3-11) is greater than the BVAL of System A (defined in equation 3-5). Let's write the difference, D, as:

$$D = BVAL_{System\_B} - BVAL_{System\_A} > 0 \tag{3-12}$$

$$D = r^2\cdot(\frac{1}{p} - 1)\cdot(N-1) > 0 \tag{3-13}$$

It is clear that D is greater than zero for N>1 and p<1. In addition, note that the difference on BVAL is proportional to N-1 (the burst packet size minus one) and inversely proportional to the packet completion rate minus one. This relationship provides the desired features. In practice the magnitude of the increase of BVAL in System B over that in System A is important; this defines the utility of BVAL as an easily discernible figure of merit. Refer to equation 3-14 for that magnitude of relative increase.

$$\frac{D}{BVAL_{random}} = \frac{p \bullet (1-p)}{(2-p)} \bullet (N-1) \tag{3-14}$$

Refer to Figure 3.3 for a graph that depicts the relative % increase in BVAL as a function of the burst size, N, and the Packet Completion rate, p. As it can be seen in Figure 3.3, the larger the N (i.e., the burst size) the greater the BVAL becomes (which is a desired property in this measure of performance). As a result of this derivation, one knows that BVAL is greater for a burst cell error rate environment with a fixed size equal to N (as long as N is greater than 1) than for a random cell error rate environment.

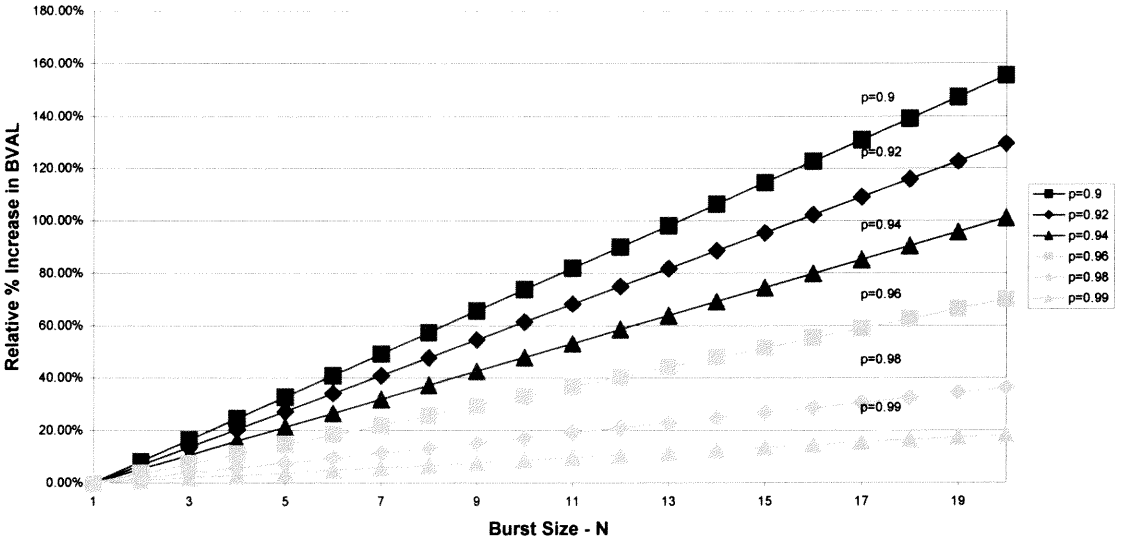

Figure 3.3 Relative % Increase in BVAL as a Function of Burst Size (N) and Packet Completion Rate (p)

The vocoder operates in speech intervals in the range of 16 to 32 milliseconds [15]. These are generally optimized for high coding efficiency (i.e., they are just slightly below the threshold of speech interval above which perceptual click noise appears [24]). A conservative estimate is that the perception threshold is 48 milliseconds, then that is equivalent at 64 kbps (using PCM) to 8 contiguous ATM cells or a smaller number of larger voice IP packets. In other words, a burst packet loss of 8 such cells will be clearly perceptible as a speech disruption. In schemes that use compression (e.g., ADPCM) even smaller burst sizes may be perceptible. Figure 3.4 enlarges the region of smaller values of N in expressing BVAL. Note that the % relative increase even for a 99% Packet Completion Rate and small burst sizes the relative % increase in BVAL is several full percentage points in value and thus clearly measurable.

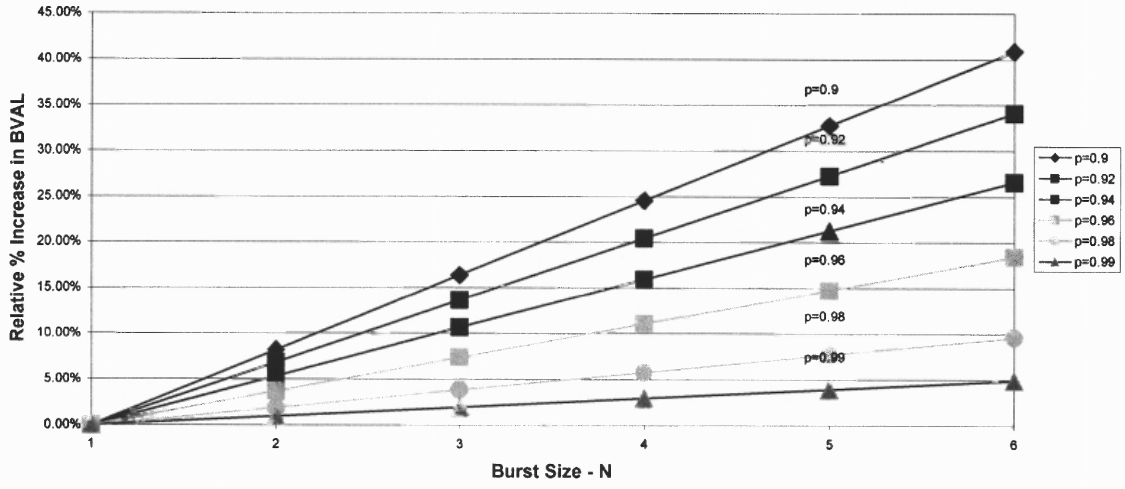

Figure 3.4 Relative % Increase in BVAL as a Function of Burst Size (N) and Packet Completion Rate (p)

Now, there is a need to generalize the results for a burst error rate environment on which the burst size is not necessarily fixed with a size N, but it varies. To prove this aspect, a System B that exhibits packet burst errors of different sizes will be assumed. The observed space will be partitioned in a series of periods $\{A_1, A_2, A_3, \ldots\ldots A_M\}$. Each period will contain packet errors due to a given size burst and separated by one successful packet arrival. Each period will exhibit a different packet completion rate; however, the long term average packet completion rate is still p.

One can state the following:

$$BVAL / A_{a(burst)} > BVAL / A_{1(random)}$$
$$BVAL / A_{2(burst)} > BVAL / A_{2(random)}$$
$$\vdots$$
$$BVAL / A_{M(burst)} > BVAL / A_{M(random)} \quad (3\text{-}15)$$

And because,

$$BVAL_{(random)} = E[L / A_{1(random)}] \cdot P[A_1] + E[L / A_{2(random)}] \cdot P[A_2] + .. + E[L / A_{M(random)}] \cdot P[A_M]$$

It follows that,

$$BVAL_{(burst)} > BVAL_{(random)} \quad (3\text{-}16)$$

Since,

$$E[L / A_{1(burst)}] > E[L / A_{1(random)}] \quad (3\text{-}17)$$

CHAPTER 4

ADVANTAGES OF USING SATURATION ROUTING

4.1 Multicast Routing with Saturation

In addition to its flexibility and ease of implementation, Saturation Routing can also have other advantages over conventional routing schemes. That is assuming that the overhead of saturation routing can be made insignificant, as it has been shown. Among those, one can mention multicast routing. Currently, there is two proposed approaches for providing the support needed. However, these approaches do not possess all of the needed characteristics to effectively support a truly dynamic multicast group. A brief description of the two approaches is included below:

- The Virtual Circuit Mesh Approach– that is each source interested in sending messages to the multicast group establishes a unidirectional point to multipoint VC with the members of the multicast group, or
- The Multicast Server Approach – each source establishes a VC with a common node (called multicast server). The multicast server in turn establishes a point to multipoint unidirectional VC with each intended destination.

There are issues with both approaches. Among those, one can find:(1) scalability issues, and (2) long settling time (i.e., the time needed by the network to adapt to a new member joining the group) Regardless of the approach selected, there is a need for ATM switches to establish effectively and rapidly point to multipoint unidirectional VCs. In addition, it is desirable for a routing algorithm to support both bidirectional point to multipoint VCs as well as to be able to support a feature on which a node other than the

source can add more destinations to the group. The description of the integration of Saturation Routing with multicast will be constrained to the functionality needed for unidirectional point to multipoint services, and not to the more generic bidirectional point to multipoint service.

The general principle behind a Multicast Service is understood to be the transmission of data to be delivered to a group of users. This group of users is dynamic, meaning individual users will subscribe or retire from the multicast service at will. In addition, a user does not need to be a member of a particular multicast group in order to send traffic to that multicast group.

The Internet Engineering Task Force (IETF) has been considered to be in the forefront of research with wide area multicasting. Internet researchers have been using the experimental MBONE network to test some of these multicast routing protocols. Distance Vector Multicast Routing Protocol (DVMRP) has been the routing used in the MBONE to multicast to Internet users the (IETF) conferences.

With respect to ATM, the Request for Comment (RFC) 2022 – "Support for Multicast over UNI 3.0/3.1 based ATM" – describes proposed approaches to solve the problem of multicasting in ATM. These approaches are based upon the fact that the ATM User-Network Interface Specification Version 3.1 provides multicast support. However, this support has the following limitations: (1) Only point to multipoint , unidirectional VCs can be established, and (2) only the source node may add or remove members of the multicast group.

Two alternatives will be provided to implement the Unidirectional Point to Multipoint Service. The first approach applies to the more generic problem on which

multicast members are throughout the network and the network topology is not completely known. This approach is described as follows.

First, the source node issues a saturation routing request message to the multicast group identifier. Second, all active members of the multicast group respond to the request issued by the source node. As each multicast destination is received, they start the process of selecting a path back to the source. This path selection informs the intermediate nodes that they are providing the requested multicast service. In addition, as opposed to the unicast routing case, the destination(s) continue the search for more destinations through its outgoing links. As a result this marked path (constituted by the intermediate nodes involved in the path selection and also called the multicast backbone) can also be utilized by the forthcoming path requests from the other nodes. This is because, once the multicast backbone exists, the nodes do not require to find a path to the source any longer (i.e., a path between the intended destination node and the multicast backbone is sufficient to satisfy the request). Furthermore, as more users join the multicast group and their paths are setup, the multicast backbone grows in size. New members can join by just extending the existing multicast backbone.

A less generic approach can also be used if one knows the topology of the network and the clustering characteristics of members who wish to subscribe to the multicast group. In this case, rather than issuing a general saturation routing request, the source node now issues a request to at least one member of each of the identified clusters – (this member is designated as cluster server – this can be the Peer Group Leader on a PNNI network). The cluster head members are responsible to establish their point to multipoint connections to the rest of the members of the cluster (these members are

designated cluster clients). As a result, a hierarchical backbone (in which the first level of the backbone is between the source and the cluster servers and the second level of the backbone is established between the cluster servers and their clients) has been provided.

4.2 Load Balancing Advantages

An inherent property of using Saturation Routing is the capability of introducing the concept of Load Balancing. That is the desire, in some instances of splitting the requirements of supporting a service across several links (paths) in the network. In some cases, specially found when a request for substantial resources is made, there is no single path that can satisfy the user exchange requirements fully. However, when using more than one path, the total contribution of several individual paths may be able to satisfy the user exchange requirement. It is believed that because Saturation Routing attempts every path in the network, this type of routing algorithm is substantially more suitable to support bifurcation of paths. In essence, every path in the network can be attempted in order to setup the network. As a result, the Saturation Routing algorithm can be modified (with the insertion of a special flag in the routing request) to allows switches to support path bifurcation.

Another important added benefit of Saturation Routing is the concept of cranking back. In PNNI implementations, the source loosely specifies the main path (or node contributors) that will become part of the selected path. Intermediate nodes flesh out the details that are unknown to the source. If for some reason, the loosely selected path can not satisfy user exchange requirements. The path will be crankbacked to intermediate specified points to attempt a route, different than that specified by PNNI. Saturation Routing, because of its feature of trying every possible path, does not require the explicit

crankback capability. It is inherent in the protocol itself. As a result, Saturation Routing can be used effectively to implicitly implement the crankback capability, without the routing overhead penalty associated with it.

CHAPTER 5

CONCLUSIONS AND RECOMMENDATIONS

An efficient method for implementing Saturation Routing has been shown. First, a mathematical queueing model (i.e., the M/D/1 queue model) was used to demonstrate that the impact of using Saturation Routing could be made negligible. For practical networks, it was determined that the impact of Saturation Routing was heavily dependent on the number of links that the network contained. In addition, a typical PNNI routing overhead was calculated for various Symmetrical Hexagonal Hierarchical Network (SHHN) in a stable condition (i.e., with no significant changes). This calculation was accomplished using different number of hierarchical levels and number of nodes. The level of overhead found on these networks will allow one to calculate a number of routing requests using Saturation Routing that will produce an equivalent amount of overhead.

Then, the cost and savings produced by using Saturation Routing was determined. It was shown that by using some of the efficiencies gained by Saturation Routing, one could in essence allow that amount of overhead to execute Saturation Routing. With the savings obtained when selecting shorter paths through Saturation Routing, one could afford bandwidth in the order of around 7 Mbps (assuming 155 Mbps links) for carrying the routing overhead information. Finally, a new measure of performance was introduced. This could be used in the future for ATM voice networks, because it is able to discriminate and penalize a system that produces burst cell errors against a system that produces the same cell error rate, yet random cell errors. The measure of performance is suitable for tracking the performance of either voice or video type of transmissions.

Other advantages of using Saturation Routing were also outlined. The main conclusion is that Saturation Routing can be made to be an efficient Routing Algorithm. This is especially true, as the user exchange requirements grow in size.

Recommendations for further research are as follows:

- Full implementation of Saturation Routing with coexistence with the current recommended PNNI approach.
- Use of graph theory to further develop relationships between network size and the number of links.
- Further exploration of Saturation Routing in the multicast and anycast areas.
- Further elaborate the role of the Burst Voice Arrival Lag (BVAL) as a measure of performance.

REFERENCES

1. The ATM Forum Technical Committee, *Private Network-Network Interface Specification Version 1.0 (PNNI 1.0)*, March 1996.

2. Sang Ho Bae, Sung-Ju Lee, W. Su, and M. Gerla, "The design, implementation, and performance evaluation of the on-demand multicast routing protocol in multihop wireless networks," IEEE Network, vol. 14, Issue 1, pp 70-77, Jan.-Feb. 2000.

3. D. Bertsekas and R. Gallager, *Data Networks*, Prentice Hall, New Jersey, 1992.

4. J. Burg, and D. Dorman, "Broadband ISDN Resource Management: The Role of Virtual Paths," IEEE Communications Magazine, Sep. 1991.

5. Ben-Jye Chang and Ren Hung Hwang, "Hierarchical QoS routing in ATM networks based on MDP cost function," Networks, 2000 (ICON 2000), Proceedings IEEE International Conference, pp 147-151, Sep. 2000.

6. Pao-Yuan Chang, Deng-Jyi Chen, and K. M. Kavl, "Multimedia file allocation on VC networks using multipath routing," IEEE Transactions on Computers, vol. 49 Issue 9, pp 971-977, Sep. 2000.

7. S. Chen, K. Nahrstedt and Y. Shavitt, "A QoS-aware multicast routing protocol", IEEE Journal on Selected Areas in Communications, vol. 18, Issue 12, pp 2580-2592, Dec. 2000.

8. Sok Sien Choy and H.C.-J. Lee, "Efficient broadcast using link state routing in Packet networks," Networks, 2000 (ICON 2000), Proceedings IEEE International Conference, pp 152-159, Sep. 2000.

9. M. S. Corson and A. Ephermides, "A Distributed Routing Algorithm for Mobile Wireless Networks," Wireless Networks 1, pp 61-81, 1995.

10. C. Courcoubetis, G. Kesidis, A. Ridder, J. Walrand, and R. Weber, "Admission Control and Routing in ATM Networks using Inferences from Measured Buffer Occupancy," IEEE Transactions in Communications, Vol 43, No 2/3/4, pp 1778-1784, 1995.

11. DuBose, K. and Sim, H., "An Effective Bit Rate/Table Lookup Based Admission Control Algorithm for the ATM B-ISDN," Proceedings, 17th Conference on Local Computer Networks, Sep. 1992

12. C. Huitema, *Routing in the Internet*, Prentice Hall PTR, New Jersey, 1995.

13. N. Kitawaki, K. Itoh, "Pare Delay Effects on Speech Quality Telecommunications," IEEE Journal of Selected Areas in Communications, vol. 7, No. 5, pp. 632-643, June 1989.

14. L. Kleinrock. *Communications Nets: Stochastic Message Flow and Delay*, McGraw-Hill, New York, 1964.

15. T. Kwok. *ATM: The New Paradigm for Internet, Intranet, and Residential Broadband Services and Applications*, Prentice Hall, New Jersey, 1998.

16. C. Manikopoulos, J. Ucles, "Average Information Staleness (AIS) as a System Measure of Performance," Proceedings of the Third IEEE Symposium on Computers & Communications, pp. 478-482, July 1998.

17. T. Mauffer. *Deploying IP Multicast in the Enterprise*, Prentice Hall, New Jersey 1998.

18. Weijia Jia, Dong Xuan, and Wei Zhao, "Integrated routing algorithms for anycast messages," IEEE Communications Magazine, vol. 38, Issue 1, pp 48-53, Jan. 2000.

19. Y. Rekhter, "BGP Protocol Analysis," RFC-1265, T. J. Watson Research Center, IBM Corp., Oct. 1991.

20. M. Schwartz, *Telecommunications Networks: Protocols, Modeling and Analysis*, Addison-Wesley Publishing Company, Massachusetts, 1987.

21. C.-K. Toh and S. Bonchua, "Performance evaluation of flooding-based and associativity-based mobile multicast routing protocols," Wireless Communications and Networking Conference, 2000. pp 1274-1279, Sep. 2000

22. G. M. Woodruff, R. G. Rogers, and P. S. Richards, " A Congestion Control Framework for High Speed Integrated Packetized Transport," Proceedings of IEEE Globecom '88, 1988.

23. W. T. Zaumen, S. Vutukury, and J.J. Garcia-Luna-Aceves, "Load-balanced anycast routing in computer networks," Proceedings Computers and Communications, 2000, pp 566-574, July 2000.

24. C. J. Wetstein, J. Forgie, "Experience with speech communications in packet networks," IEEE Journal of Selected Areas on Communications vol. SAC-1, No. 6, pp. 963-980, Dec. 1983.