University of Alberta

A Flexible Virtual Path Topology Design Algorithm

by

Rashid Qureshi ©

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of **Master of Science**.

Department of Computing Science

Edmonton, Alberta
Spring 1997

0-612-21202-5

University of Alberta

Library Release Form

Name of Author: Rashid Qureshi

Title of Thesis: A Flexible Virtual Path Topology Design Algorithm

Degree: Master of Science

Year this Degree Granted: 1997

Rashid Qureshi

Rashid Qureshi
Suite 308, 1415 Lakewood Rd. West
Edmonton, Alberta
Canada, T6K 3L2

Date: April 17/97

University of Alberta

Faculty of Graduate Studies and Research

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research for acceptance, a thesis entitled **A Flexible Virtual Path Topology Design Algorithm** submitted by Rashid Qureshi in partial fulfillment of the requirements for the degree of **Master of Science.**

. . J. Harms . . . .
Prof. Janelle Harms (Supervisor)

. . . .
Prof. Pawel Gburzynski (Examiner)

. . J. Conradi . . . .
Prof. Jan Conradi (External)

Prof. Keith W. Smillie (Chair)

Date: April 16/97.

"I pass with relief from the tossing sea of Cause and Theory
to the firm ground of Result and Fact."
by Sir Winston Spencer Churchill

To my parents
and also my sister Farzana

Abstract

Virtual paths in ATM networks can provide a useful facility to the network by reducing connection setup cost and by providing a means of dynamic routing and bandwidth allocation, resilience against network failures and simplified network management tasks. However, the virtual path topology must be designed with certain properties in order to achieve the above mentioned advantages. In this thesis desired properties of a VP overlay are identified and a flexible algorithm to generate VP topologies is designed. The performance of this algorithm is compared with other existing approaches.

Acknowledgements

This thesis is a result of efforts put in by many people. I am greatly indebted to my supervisor, Prof. Janelle Harms and would like to take this opportunity to thank her for providing all the support and guidance throughout the project. I would also like to thank my examining committee Prof. Pawel Gburzynski and Prof. Jan Conradi for taking interest in the thesis and providing invaluable suggestions.

I also like to thank CITR (Canadian Institute for Telecommunication Research) for providing the funding for this project, and also the researchers at the University of Waterloo for providing the ATM call-level simulator.

Lastly I would like to thank my wonderful parents and my sister Farzana for providing me the moral support and the strength to complete my thesis.

Contents

List of Figures

Chapter 1

Introduction

1.1 Introduction

The telecommunication network is largely digital and according to [20], in the United States the data communication on this network is growing at a rate of 20% per year; while the voice traffic is growing at a 3% per year rate. This makes the telecommunication industry want to realize the potential benefit of the data communication traffic and is leading them to develop protocols which could satisfy the needs of voice, video and data communication traffic. Also the processing power of microprocessors will improve at a rate of 60% per year according to [4]. This increase in performance will lead to applications that consume data at a much higher rate. Finally, these applications will also need to communicate with each other (e.g. video conferencing etc.) and will, ultimately, require high speed networks.

In order to handle the future user requirements, the CCITT[1] proposes the B-ISDN (Broadband Integrated Services Digital Network) service as the strategy to handle high volumes of traffic. However, the transmission system for B-ISDN services must be developed to satisfy the varied classes of traffic that it will support. ATM (Asynchronous Transfer Mode) is one such transmission system being actively researched

[1] CCITT (The International Telegraph and Telephone Consultative Committee), known as ITU-TS or ITU (International Telecommunications Union) since 1993, operates as a series of groups standardizing technologies and protocols in networking and communications. However due to the slow process of standardization, ATM Forum plays an active role in promoting and establishing working standards for ATM. ATM Forum includes many companies and government agencies.

and developed. ATM is a connection-oriented, guaranteed quality of service network. The connections are established with a specified quality of service, QoS, and the network guarantees that the QoS of the connections will be met until they are terminated. This provides the capability to support a wide variety of applications including those with real time requirements. Since connections (called virtual channel connections VCCs) must be established before the exchange of data, there is a need for speed in setting up the connections. Also, with a large number of connections with different QoS, it is important to manage them efficiently. To satisfy the above two requirements the concept of a VPC (Virtual Path Connection) is introduced in ATM.

Virtual Topology

Virtual path connections (VPCs) are characterized by two virtual path (VP) terminators nodes, a route using physical links between the terminators and the alloted capacity. These VPCs are then used to route bundles of connections, called virtual channel connections (VCCs). A VPC between the VP terminator nodes can be visualized as a logical link connecting the two nodes. Thus the topology resulting from the assignment of a set of VPs is called a virtual topology or overlay (see Figure 1.1).

A VCC between the end-users can, therefore, be routed through a series of VPCs or it can also be routed using physical links. In routing through a VPC, the call admission control functions (routing, bandwidth allocation, etc.) are performed only at the VP terminators and the transit nodes (through which a VPC is routed) do not perform any additional function. This ability to route a bundle of VCCs together at the transit nodes allows the virtual path concept to reduce the network operations and management costs considerably [21]. In particular better performance can be obtained when similar traffic is grouped together [14]. Also, if a link goes down, the VCCs of a VP can be rerouted as a unit (that is the VP is rerouted). In [5] it

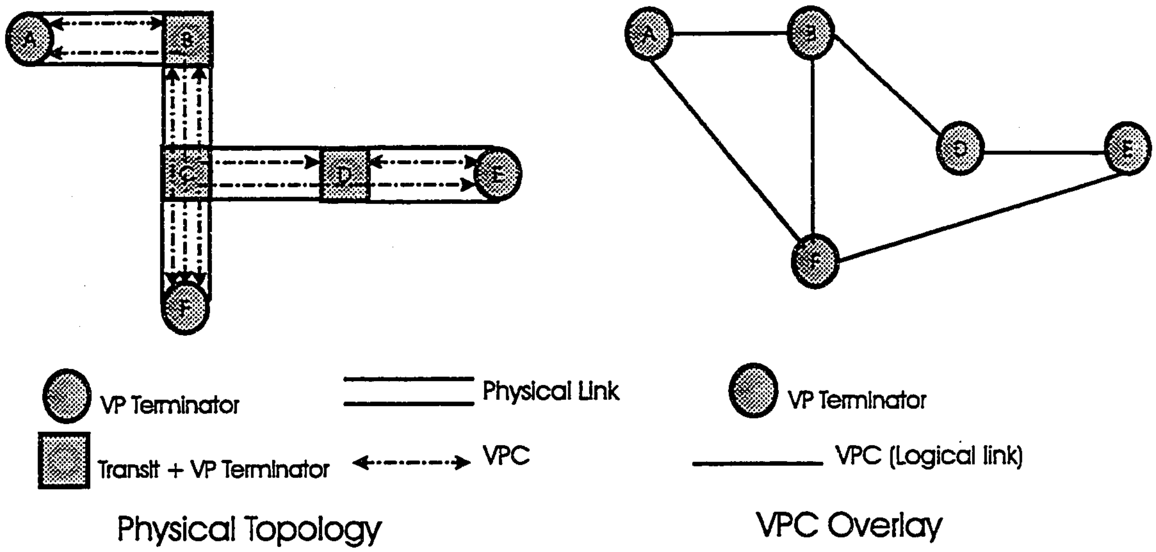


Figure 1.1: VP Overlay on a Physical Topology (From [10])

was shown that more than 90% of the processing time for connection set up can be saved if virtual channels are routed through a virtual path rather than routing them individually. Thus the VP topology should be selected in such a way that the majority of VCCs are routed through VPCs. The VPC topology should ideally also be such that the VCC routed through VPCs should not take a longer physical route than the route selected using only physical links. A survey of virtual path management can be found in [10].

However, since each VP on a physical link would be assigned some initial bandwidth (usually deterministically), this would result in bandwidth fragmentation and consequently could cause higher call blocking rates. The bandwidth fragmentation can be defined as the residual bandwidth available on the physical link after the VPs are assigned their initial bandwidths. Thus if there is a high residual bandwidth on a physical link(due to a smaller number of VPs on the link) the VPs can negotiate more bandwidth (in future) and, therefore, service future calls. On the other hand if the residual bandwidth is low (due to greater number of VPs sharing a link and deterministic multiplexing of VPs) then the VPs might not be able to increase their current bandwidth to service future calls and thus result in increased call blocking.

A goal of gigabit network design is to avoid as much computer processing as possible [20], due to the slower speed of CPUs compared to transmissions over fiber, even if it wastes some bandwidth since the bandwidth on physical links continues to grow. ATM in order to satisfy the above goal is, therefore, connection oriented in which case once a process intensive connection set up operation is performed and the routing tables are set up in the switches, the data simply follows the established path and no process intensive operation is performed. An overlay of VPs, can achieve the above goal, by reducing the connection set up time. However bandwidth fragmentation may still be a concern particularly if the VP overlay is not efficient.

Thus the VP overlay can reduce the network connection setup cost and management cost. However, it may also cause bandwidth fragmentation and, therefore, should be designed to lower the fragmentation as much as possible. A more detailed view of VPCs and VCCs is given later in this chapter.

In this chapter, we take a closer look at the ATM network. The problem statement, thesis objective and thesis outline are also presented in the subsequent sections.

1.2 The ATM Protocol Stack

The ATM protocol stack is shown in Figure 1.2. There are three layers namely the physical layer, the ATM layer and the adaptation layer. The primary purpose of the physical layer is to transmit cells from one end to the other. However, before transmitting, the cells are arranged in a prespecified format. The format used for ATM is SDH-based (Synchronous Digital Hierarchy) which is derived from SONET. The current standard defined by CCITT for ATM, specifies an access speed of 155.52 Mbit/s and 622 Mbit/s for the transmission medium.

The ATM layer is responsible for establishing and maintaining connections. Since

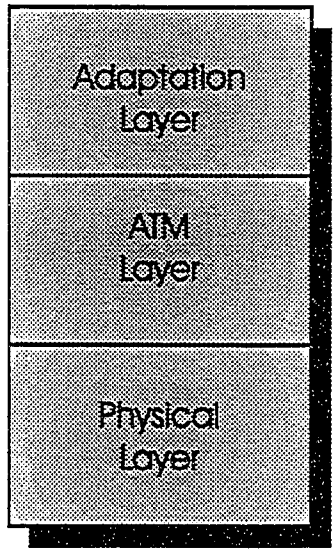


Figure 1.2: Protocol Stack for ATM

the thesis is mostly related to this layer, it is discussed in detail in the next section.

The ATM adaptation layer (AAL) as the name implies is used to adapt the various traffic types into a form suitable for ATM. It can also be looked as the transport layer for ATM. However, none of the protocols developed for this layer offer reliable end-to-end service like the well known TCP for the Internet. Initially CCITT defined four protocols (AAL1, AAL2, AAL3 and AAL4) for this layer. These protocols are developed to serve four categories of traffic namely constant bit rate, variable bit rate, connection oriented and connection-less respectively. The protocols AAL3 and AAL4 have very similar framing formats and minor differences in using the headers. Thus the two protocols were later combined as one protocol AAL 3/4. However, AAL 3/4 has a high overhead and therefore the data communication community proposed and later standardized a Simple and Efficient Adaptation Layer (SEAL) or AAL5 protocol. This protocol, due to its simplicity and low overhead, is said to replace the use of AAL 3/4.

1.3 The Asynchronous Transfer Mode(ATM) Layer in Detail

ATM (Asynchronous Transfer Mode) is a cell relay network that is being developed to provide B-ISDN services. ATM currently can allow speeds of up to 622 megabits/sec speeds, but in the near future gigabit speed will be possible. The cells are short fixed length packets and switching is performed on the basis of control information in each cell.

The current telecommunication network are circuit switched where the connections are established based on the peak rate of traffic. Also, different connections on the channel are identified by their specific time slot as determined during the connection setup phase (time division multiplexing). The connections then send data using their slot synchronously. For the data communication traffic, which is mostly bursty in nature, the above setup would result in waste of bandwidth, as most of the slots would go empty. Thus in ATM instead of reserving slots for the connections, the connections can be identified by the information in each cell header. By doing this the cells may not need to be delivered at a specific interval, but can be delayed or dropped if the need arises; hence the term "Asynchronous Transfer Mode". The advantage gained by the above approach is that connections with similar properties can now be statistically multiplexed. However, if the statistically multiplexed traffic bursts at the same time, some cells must be dropped. In any case, the ATM standard proposed by CCITT requires cells to be delivered in order; therefore, allowing relatively unsophisticated hardware, like telephones, to be attached to the network.

Question: Why don't we use the normal packets (variable length) instead of fixed length cells in ATM? In a packet switched network high priority packets may be delayed due to falling behind a large packet at the router. However, by dividing the packets into smaller fixed size cells the high priority cells can be interleaved between low priority cells at the switch avoiding excessive queuing delay. Also by using fixed length cell sizes the performance of switches is improved [19].

However, it must be noted that the size of cell is dependent upon the following factors:

- Speed of switches.
- Speed of links.
- Type of traffic.

If the switches are slow then larger cell sizes would be preferred as small cells increase the traffic elements to be processed at each switch. On the other hand, if links are slow then keeping the small cell size is preferred because, at the switches, the delay of transmitting a cell would be small and thus the cell delay would be negligible. However, if the links are fast then the cell size does not matter. Finally, the cell size can be selected to better suit the traffic they are carrying. For voice or other such delay sensitive traffic, a smaller cell size is preferred as each cell would take a small amount of time to be filled and then transported. For throughput sensitive traffic such as the transfer of files relatively larger cell size is preferred to minimize the control overhead in the cells.

In any case, the cell network needs to be designed such that the routing functions at the switches are simplified. This is done to increase the throughput of the network and also to minimize the control overhead in each cell. The disadvantage of this simplification is that if a cell gets corrupted in transmission, the receiver cannot uniquely identify the corrupted cell and ask for its retransmission. The whole packet would be required to be retransmitted. The above scenario could degrade the performance of the network; thus a cell network is only feasible for the physical medium which has a very low error probability. Fiber, at the moment, is the medium of choice in the above case.

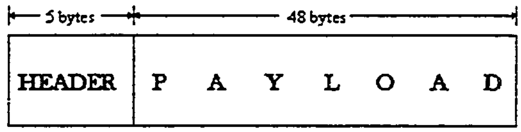


Figure 1.3: ATM Cell Format

1.3.1 ATM Cell Format

The ATM standard defines a cell size of 53 bytes. The first 5 bytes are used for header and the remaining 48 bytes are used to carry the data (see Figure 1.3). The header has two distinct formats, one to be used at the user network interface (UNI) and the other to be used at the network to network interface (NNI) (see Figure 1.4).

UNI and NNI Header Format

The header format for UNI and NNI is shown in Figure 1.5. At the UNI the first four bits are used by the GFC (Generic Flow Control). which provides medium access control functions to the terminals connected to a shared medium. The role of GFC is not finalized yet and some researchers argue against the effectiveness of this field as it takes away some precious bits from the VPI identifier limiting the combined address space of VPI and VCI to only 24 bits. The GFC field is not present at the NNI level.

The VPI (Virtual Path Identifier) and VCI (Virtual Channel Identifier) take 8 and 16 bits respectively in the UNI. In the NNI, the VPI and VCI are 12 and 16 bits respectively. These two fields identify each connection uniquely. Since the combined address space of VPI and VCI is too short to identify connections globally, at each hop, the switch assigns a new number and remembers its association with the connection in a routing table. This process goes on until the destination is reached. Thus the addresses are kept unique locally (only at each hop) rather than globally. This process is described more fully later in this section.

The PTI (Payload Type Identifier) field is 3 bits long. The purpose of this field is to

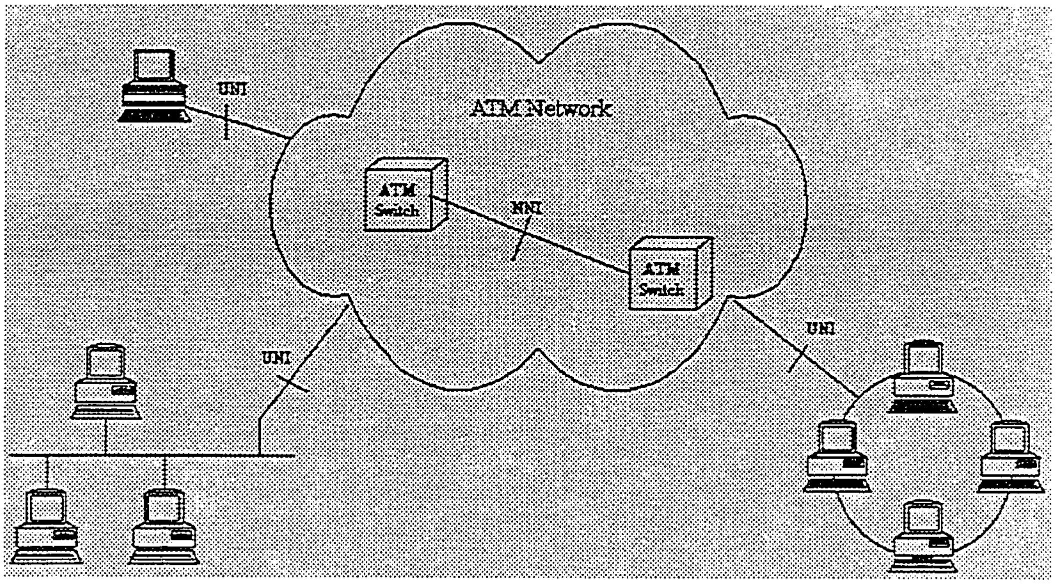


Figure 1.4: User-Network and Network-Network Interfaces in ATM

5 bytes UNI header

Generic Flow Control (GFC)	Virtual Path Identifier (VPI)	Virtual Channel Identifier (VCI)	Payload Type (PTI)	Cell loss Priority (CLP)	Cyclic Redundancy Code (CRC)
4 bits	8 bits	16 bits	3 bits	1 bit	8 bits

(a)

5 bytes NNI header

Virtual Path Identifier (VPI)	Virtual Channel Identifier (VCI)	Payload Type Identifier (PTI)	Cell loss Priority (CLP)	Cyclic Redundancy Code (CRC)
12 bits	16 bits	3 bits	1 bit	8 bits

(b)

Figure 1.5: ATM Header Format: (a) UNI header format (b) NNI header format

identify the data carrying cells from the operations, administration and management traffic. The CLP (Cell Loss Priority) is a single bit field. This field will be set to (CLP=1) in cells which exceed the agreed traffic bandwidth of a connection. Thus if there is a sudden surge in traffic the switches will first drop the cells with the CLP bit set to (CLP=1). Through this mechanism a user can exceed his traffic rate, but the excess traffic can be discarded if the need arises. In this way the waste of bandwidth can be reduced and also this feature gives ATM a mechanism to control the rate of flow of traffic. This mechanism is more suitable than the end-to-end flow control mechanism for high speed networks. The reason is that in high speed networks, by the time the source receives the feedback, either the congestion is removed or the source has already transmitted a large amount of data, worsening the congestion.

The CRC (Cyclic Redundancy Code) field is 8 bits long. The CRC generates a checksum by using the first 32 bits of the header and the generator polynomial $G(x) = x^8 + x^2 + x + 1$. This checksum is stored in the CRC field of the header and will reduce the probability of cells delivered to wrong destination due to error in the header. Since the CRC field in ATM only covers the header it is also called HEC (Header Error Control). The payload field is not included in the CRC, because the checksum would then have to be quite large. The error in the payload field would eventually be detected by the higher layer and the defective data would be retransmitted. The CRC code can correct single bit errors and detect burst errors with a high probability. As the error characteristics of fiber is a mix of single bit and relatively large burst errors [23], the above mentioned properties of CRC would be quite suitable for ATM.

1.3.2 Network Operations in ATM

ATM is a connection-oriented service in which a connection is established before data can be exchanged. ATM defines two basic types of connections called the virtual channel connection (VCC) and the, virtual path connection (VPC). In the following sections the network operations of the ATM layer are discussed in detail.

QoS and Traffic attributes

Associated with each VPC and VCC are some traffic attributes and QoS (Quality of Service) requirements. The traffic attributes specify the behavior of traffic and can be represented as [2]:

- Peak Cell Rate (PCR).
- Cell Delay Variation Tolerance (CDVT).
- Sustainable Cell Rate (SCR).
- Burst Tolerance (BT).
- Minimum Cell Rate (MCR).

The QoS can be specified by the following parameters:

- Cell Delay Variation (CDV).
- Maximum Cell Transfer Delay (Max CTD).
- Mean Cell Transfer Delay (Mean CTD)).
- Cell Loss Rate (CLR).

However, not all traffic types need all of the above listed traffic parameters or QoS parameters. Therefore a set of QoS classes has been constructed, which group the traffic types based on their traffic and QoS parameters. The following is a list of QoS classes defined by ATM Forum:

- **Constant Bit Rate (CBR):** The traffic using the CBR service uses a fixed amount of bandwidth during the connection. This traffic requires a QoS guarantee on the delay and loss rate. The traffic can be specified by using PCR(Peak Cell Rate) and CDVT(Cell Delay Variation Tolerance) traffic parameters. Circuit emulation and some multimedia applications can use this service.

- **Variable Bit Rate (VBR):** The traffic using the VBR service, specifies a peak rate and sustained or average rate for the connection. Some VBR traffic is delay sensitive and some is not. The ATM Forum therefore defined VBR(RT) for real time traffic and VBR(NRT) for non real time traffic. For VBR(RT) the QoS guarantees include delay and loss rates. An example of VBR(RT) traffic may be MPEG video transmission used in video conferencing.

  For VBR(NRT) the QoS guarantee includes only loss rate. The VBR(NRT) can be used by multimedia mail [13] or by Frame Relay Internetworking [2]. Both traffic types can be characterized by SCR(Sustainable Cell Rate) and BT(Burst Tolerance) traffic parameters.

- **Available Bit Rate (ABR):** The available bit rate traffic does not require guaranteed bandwidth, but rather uses whatever bandwidth is available. However it requires QoS guarantees on cell loss rate. In order to implement ABR, a feedback mechanism is needed from the network. That is, the network must inform the sender to slow down when the bandwidth is low and correspondingly the sender slows down in order to satisfy the cell loss rate of ABR service. The ABR service is different than VBR(RT), in the sense that VBR(RT) requires guarantees on the bandwidth. The traffic can be specified by PCR(Peak Cell Rate) and MCR(Minimum Cell Rate) traffic parameters. The network then allows traffic between PCR and MCR depending upon the availability of resources. If MCR is not set to 0 then the network guarantees that the connection can get at least MCR otherwise the connection will not be established. Currently Internet is based on an ABR service or best effort service. Thus this service can be used to carry the current Internet traffic like email, file transfer and so on [13].

- **Unspecified Bit Rate (UBR):** The traffic using UBR service, needs no guarantee on the delay, loss and bandwidth. This service is the simplest to implement and can be used to support LAN traffic. However, ABR service which is under development and, once completed, may be better suited for providing such services.

Raj Jain gave a very interesting analogy of the above types of services to airplane reservation system in [13] as

> CBR is confirmed reservations with no recourse if you do not show up. VBR is like confirmed reservation but you do not pay if you do not show up. ABR is standby. You get to go if seats are available. Having standby service is good for the airline. They can fill their seats that would otherwise would have gone empty. The service is also good for passengers. They can travel cheaply particularly if they don't have to be at their destinations at a certain time. UBR service is not currently offered by the airlines. Passengers traveling on UBR class may be allowed to board a plane but may be asked to leave at connecting airports if seats are not available. ABR users would generally be asked to stay home as much as possible if their routes are congested.

Switching, Signaling and Routing

When the switch receives an incoming cell on a port with particular VCI/VPI value in its header the switching operation in ATM is quite straight forward. The switch indexes the routing table using the port and VCI/VPI value as the index to determine the outgoing port and new VCI/VPI number. This new VCI/VPI number is written in the cell header and the cell is written on the outgoing port (Figure 1.6). The process is repeated until the destination is reached. The switching can be performed only on the VCI label or VPI label or using both VPI and VCI label. In ATM there are three types of switches: (i) **VP switch** can do switching only on the VPI label (ii) **VC switch** can do switching only on the VCI label, (iii) **VP/VC switch** can do switching on both VCI and VPI labels. The routing table in each switch must be set prior to any exchange of data. This is done by using the ATM signaling protocol Q.2931[2] between the sender and receiver across the UNI. The signaling request, however, needs to be routed which is not defined by Q.2931 protocol. However, the ATM Forum has come

[2]This is a modified version of ISDN signaling protocol Q.931

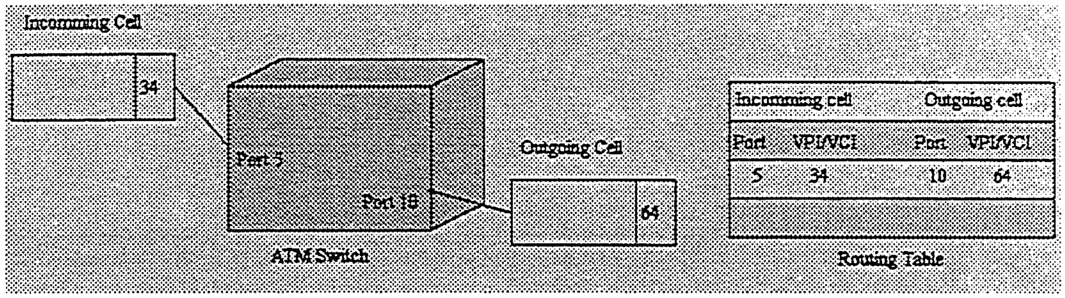


Figure 1.6: Switching operation in ATM

up with an initial signaling and routing protocol to be used at the NNI level. This protocol is collectively referred to as P-NNI.

The routing procedure for connection setup in ATM is quite complex as the objective for a routing protocol is to find a path which would be able to satisfy the (multi-parametric) QoS requirements of the connection. Several approaches have been suggested which attempt to solve the problem. However, most of these approaches generate a feasible path, rather than an optimal path[3]. These approaches can be divided into two types of routing classes *(a) source routing* and *(b) hop-by-hop routing.*

In hop-by-hop routing each switch passes the signaling request to the switch closer to the destination. However, in ATM this process is a bit complex, as the switch must also take into account the current state of the entire network to determine whether the next switch would be able to satisfy the QoS requirements of the call, also the switch performs a local operation to see whether it has enough resources to make the connection (this operation is also termed as CAC (connection admission control)). On the other hand, in source routing the first node(switch) in the connection path determines the whole route based on the QoS requirements of the connection. In order to do this each switch must periodically distribute its local information to all

[3]The optimal solution is found to be NP-Hard.

the other switches in the network. Once the route is determined it is inserted in the signaling request. The intermediate switches in this way only perform local CAC. However, in source routing the processing overhead is increased compared to hop-by-hop routing as the signaling request must now carry the entire routing information. The P-NNI protocol uses source routing in establishing the connections.

In the thesis we will differentiate the connection routing into three classes namely (i) **VP routing** if the connections are routed using only VPs, (ii) **VC routing** if the connections are routed using only physical links and (iii) **VP+VC routing** if the connections are routed using a combination of both physical links and VPs. In order to simplify the routing procedure, the metric used for routing in the thesis is minimum hop count.

Role of VCC and VPC

The VCC establishes an end-to-end connection and the connection request must pass through all the nodes between the sender and the receiver. The delay in setting up a VCC is therefore directly proportional to the number of nodes in the path from sender to receiver. The connection is established for the duration of a call and then torn down when the call is finished. The combined address space of VPI and VCI at the UNI interface is 24 bits. Thus any two end-users can establish at most 2^{24} connections between them. However, ATM provides two levels of connection setup based on the VCI and VPI values. A VCC is identified by the VCI number in the cell. The numbers are unique locally (at each routing table associated with a port in the switch) and at each hop, the numbers are swapped with the next available number in the routing table at a switch. This swapping is transparent to the end-user where the VCC is only identified by the port and VCI number given to it by the network switch at the UNI interface. Since the VCI field is 16 bits, therefore, a total of 65,536 VCCs can be established between two end-users. Thus a physical link between two switches can carry 65,536 VCCs. If the link fails all these connections must be rerouted to alternative routes. The delay in this case could be quite large, resulting in degradation of QoS and possible connection termination.

The role of a VPC is to establish permanent/semi-permanent connections based on the VPI value in the cell header. Once a VPC is established between two node pairs, called **VP terminators,** a VCC between these node pairs can be established by assigning the next available VCI number within the established VPC. Now the VCC (between the VP terminators) is identified by both VPI and VCI values in the cell. The VPI will identify the VPC and VCI will identify the VCC within a particular VPC. At the nodes through which the VPC passes between the VP terminators, called **transit nodes,** the switching is performed only on the VPI field of cells and the VCI values are left unchanged. Thus the VCCs which are associated with a particular VPI value of a VPC (at the VP terminators) will be switched together at the transit nodes. Now at the UNI interface between the two end-users 256 VPCs can be established and each VPC can carry up to 65,536, giving a total of 2^{24} connections between the end-users. This number is the same as the combined address space of VPI and VCI at the UNI interface.

The advantage is that the VCC is established without performing routing functions at the transit nodes between the VP terminators. Thus a VPC can bundle VCCs (Figure 1.7) and route these connections together, reducing the network operations and connection setup costs. Now if 65,536 VCCs are routed through a VPC, these can be identified by a single entry (VPI label) in the routing tables at the transit nodes. A link failure between switches will require only finding an alternative path for a single VPC and all the 65,536 VCC will be rerouted by a single operation. The survivability of the network is improved as, in case of link failures, a smaller number of entities would be required to be managed. Based on the switching capability of a switch, all VPCs must terminate on a VC switch or VP/VC switch. A VP switch can be used only at a transit node; while a VP/VC switch has the capability of both the switches.

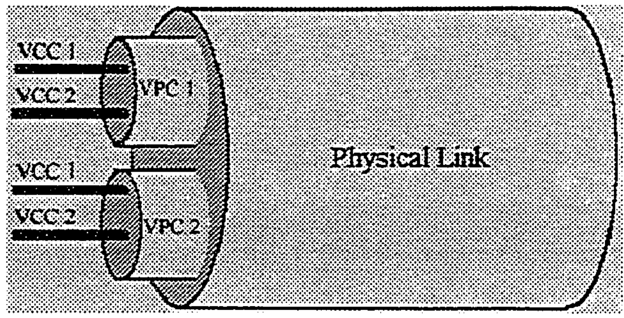


Figure 1.7: Physical Link and the Virtual Links

A VPC has an associated QoS. In order for connection QoS to be met, the QoS of a VPC must satisfy the most stringent requirement of the VCCs that are multiplexed on it. Thus it is desirable to group similar traffic (thus similar QoS) VCCs on a single VPC [14]. If different traffic types exist (CBR, VBR, etc) in the network then it is feasible to construct separate VPCs for each traffic type. In summary the VPCs can provide lower management cost by grouping similar VCCs and also the connection setup cost can be reduced since VCCs using a VPC only need to perform routing operations only at the VP terminators.

A VCC can now be visualized as a concatenation of VPCs and physical links (see Figure 1.8). The VPC is characterized by the VP terminators, assigned capacity and the physical route between the VP terminators. The routing for VPCs is different than the connection setup routing pointed out earlier. In the case of VPC routing only physical links can be utilized. A **VP topology or layout** is a set of VPCs established over the physical topology. The capacity of the virtual path can be deterministic or statistical. In the case of deterministic capacity, each VPC is multiplexed on a physical link based on their peak bandwidth requirements, whereas in statistical, the VPCs are multiplexed based on its computed effective bandwidth. Since deterministic multiplexing of VPCs will reserve the peak bandwidth for all the VPCs sharing a

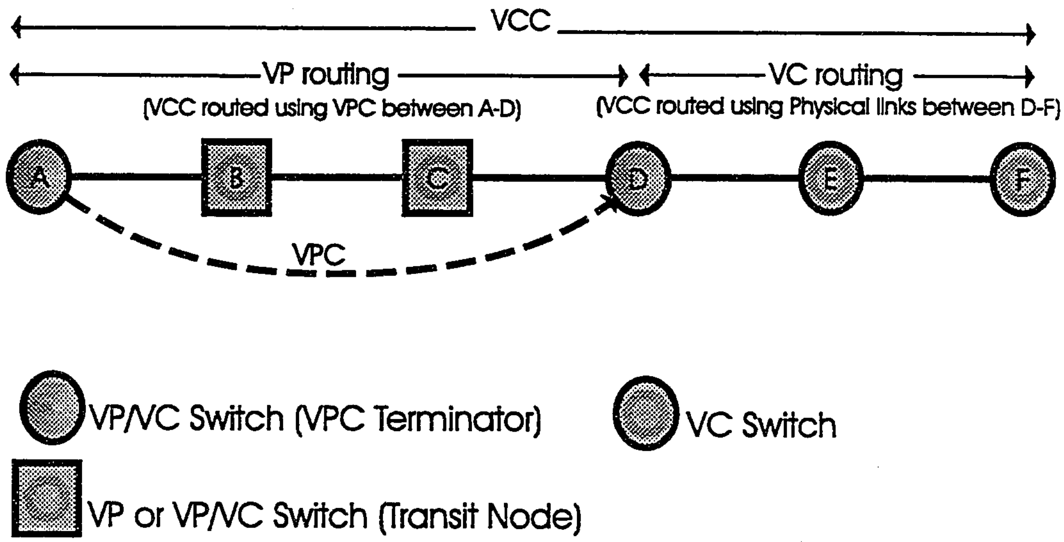


Figure 1.8: VC Connection using VP+VC routing (From [10])

link, if some VPCs are idle on a physical link their assigned capacity cannot be taken away by other VPCs on the same physical link. This could result in poor bandwidth utilization and bandwidth fragmentation; and as a result, possibly, higher call blocking rate. However, this can be improved by either increasing the capacity on the physical link or by keeping the number of VPCs sharing a physical link small. If the VPCs are multiplexed statistically, the VPCs can share their capacity with other VPCs on the link. This may give lower bandwidth fragmentation, but statistical multiplexing of VPCs complicates the process of maintaining QoS for VPCs and the VCCs within. Thus, due to the complexity involved with statistical multiplexing, VPCs are generally deterministically multiplexed. Similarly VCCs within a VPC can either be multiplexed deterministically or statistically depending on the traffic type. Also the QoS of a VPC must meet the most stringent QoS requirement of the VCCs. Thus, if the network has more than one traffic class, for each traffic class a separate virtual path topology should be established.

It follows from the above that the VP topology can lower the management, operation and connection setup costs of the network. In order to achieve the benefits of VPCs, it is clear that a VP topology must be generated before calls arrive. However, for a given network, the VP topology must be determined with certain properties such that it can be useful to the network. One important property is low bandwidth fragmentation in order to provide lower call blocking rates. Also the generated VP topology should require minimal topological changes over a long period of time. A change in VP topology could require rerouting of existing connections, which may create delays that would adversely affect the QoS of a connection. Thus the motivation is to design a general VP topology which could efficiently satisfy the traffic requirements by minor VP capacity adjustments rather than topological adjustments. This can be achieved by establishing a VP topology which is not tuned to a specific traffic pattern but a general topology which is effective for all traffic scenarios. The generated topology would be assigned some initial capacity and as the need arises the capacity of each VP can be dynamically adjusted by using a dynamic bandwidth allocation scheme as in [3]. Other properties are described in Chapter 3.

1.4 Problem Statement

In this chapter an attempt is made to highlight the key features in an ATM network. ATM is definitely going to play a major role in future gigabit network environment, as it has all the elements that the future networking environment will need. The problems encountered with respect to flow control, signaling, routing and QoS have been actively researched and solutions are being developed to make ATM a reality. Also there is a need to develop a long term VP topology layout which could provide facilities to the network to help it reduce network operation, administration and management costs (OAM).

One very simple implementation of a VP topology would be to establish a VPC between every node pair in the network. This topology would result in a minimal average connection cost topology, but due to the limited size of VPI address space it is only suitable for a very small setup. This is because in a network of N nodes there can be $N(N-1)$ VPCs or possibly much more (in order to provide alternative routes, VPCs for each QoS and traffic type between the nodepairs). These VPCs, would result in an average of $N-1$ entries in the routing table of the switches. However, in the worst case, some centrally located switches could have $O(N^2)$ entries [12] and since $O(N^2) \leq 2^{12}$, the maximum value of $N = 64$. Also the addition and deletion of a node would cause excessive processing of establishing and removing of VPs. Moreover it also results in bandwidth fragmentation, as the number of VPs sharing a link would be large, and thus is not suitable for real time network. Finally due to the large number of VPs sharing a link, the network would be less resilient to link failures and could result in unacceptable levels of loss of flow.

The other approach is to provide a sufficient number of VPs in the ATM network assigned with certain properties such that it would provide some desirable facilities. In the literature several papers have been published emphasizing the importance of a particular property and developing a VP topology which would optimize that property. However, it is yet to be seen which property is more important and whether by selecting a particular property it would provide an effective *general* VP topology.

For large networks, having an effective VP topology is even more important. The ATM network can cover a vast geographical region and, in the future, possibly covering the entire globe. Thus an effective VP topology can play a vital role in such a scenario by reducing the management, connection setup and failure costs.

1.5 Thesis Contribution

The aim of this thesis is to investigate the properties useful in a general VP topology and to develop a method that creates a VP topology that satisfies these properties. The generated VP overlay should result in minimal topology reconfiguration over a long period of time. The algorithm developed should also be able to efficiently generate VP topologies for large networks. We also need to look into the physical properties of the network and determine how they will affect the generated VP topology. In developing our approach we have not considered any *a priori* traffic pattern, as the goal is to develop a general VP topology which could provide effective service in all cases.

The method developed in this thesis will be based on an "assignment model" (of operations research) where by minimizing a cost function (which reflects the important properties) the method chooses the VP terminators of a VP overlay. The cost function and the structure of the algorithm will allow some important properties to be incorporated into the generated VP overlay, discussed in chapter 3. The routing of generated VPCs is then performed in such a manner that the maximum link load is small.

The existing approaches for generating a VP topology are also investigated and compared with our approach.

1.6 Thesis Outline

This thesis begins with an introduction to the VP topology problem and the goals to be achieved by this thesis. Chapter 2 gives an overview of VP topology problem and discusses some of the existing approaches in detail. In Chapter 3 we develop the properties which are important in a general VP topology by analyzing some of the existing approaches and present our approach for generating an effective VP topology. In Chapter 4 the properties of the VP topology generated by our approach is analyzed and compared against some other approaches. Chapter 5 provides the analysis of simulation results and provides some insight into the usefulness of a VP topology. Finally we conclude in Chapter 6 with some suggestions for future work.

Chapter 2

Background

In this chapter the current research in VP topology is studied and analyzed in order to identify properties suitable for a general VP topology. Also, since we are interested in developing a long term general VP topology we need to see that the selected properties would result in an efficient utilization of resources. This is important as a general VP topology that results in high bandwidth fragmentation or a large number of VPs sharing a link will compromise the survivability of the network.

2.1 Introduction

In [10], several issues related with the management tasks of virtual paths in ATM networks are discussed and the definition of an effective VP topology is given. An effective VP layout should provide low setup and switching costs, as well as robustness to handle unexpected traffic conditions and component failures. The setup costs can be characterized by the following scenarios

- A VPC to the destination is available and the call is accepted immediately, by performing decisions at the VP terminators only.
- The VP to the destination is available but the VCC has to traverse through several VPCs to reach the destination. Thus due to an increased number of hops the setup cost increases.

- There exists a VP to the destination, but not enough bandwidth. To avoid call blocking the additional bandwidth can be made available by some capacity reallocation activities, but the setup cost in this case would be higher than in the previous case.

- There is no VP available to the destination. In this scenario a VP needs to be setup and also some bandwidth needs to allocated to the VP. Thus in this scenario the setup cost would be higher than all the previous cases.

Thus, the effectiveness of a VP overlay can be determined by the lower setup costs, which would in turn guarantee lower blocking probabilities and also greater chances of the first two of the above-mentioned scenarios.

In the following sections, literature related to our research is being reviewed with a viewpoint to understand the important properties that are required for designing an effective VP topology. The literature reviewed in this section attempts to solve the VP generation problem in a number of ways. However, we can broadly categorize these methods into two main categories:

- Traffic dependent.

- Traffic independent.

The *traffic dependent* approaches need a traffic matrix between the source pairs in the network in order to generate a VP topology optimized to facilitate the current traffic flow. However, in high speed networks, the traffic flow can be highly volatile [17] and, in the long run, the generated VP topology becomes inefficient and a burden on the network resources. The topology may need to be changed when the traffic changes. If the topology is changed, existing connections may have to be rerouted. This will result in delays and possibly problems with meeting QoS constraints. Alternatively, the old and new topology can coexist allowing the existing connections to use the old topology until they are finished while the new connections are established using the new topology. However, this may provide a strain on resources resulting in more blocked calls.

On the other hand, the *traffic independent* approaches do not need any *a priori* traffic information. The VP topology developed in this case is supposed to give satisfactory performance (though not optimal) even with changing traffic scenarios and require less topology reconfiguration. The generated topology is therefore general and provides certain facilities to the network to improve its efficiency/performance.

In the following sections, the methods in each of the above mentioned categories are discussed and analyzed. By the end of this chapter, we will have an idea of the important properties that are needed to generate an effective general VP topology. The next chapter will then develop these properties and provide justification of their importance.

2.1.1 Traffic Dependent VPC Overlays

In this section, several representative methods which require a priori information about the traffic pattern are discussed. Each of the methods discussed attempt to optimize certain properties in a VP overlay.

- In [1], a heuristic approach is developed to generate a VP topology to provide k alternative VPs between the source pairs which have a high volume of traffic. The number of alternative paths k is a function of the amount of traffic between the source pairs and the physical link capacity. The capacity on the VPs between the source pairs is assigned according to the total traffic between them. However, the capacity is divided between the k paths in such a way that the primary (first) path gets more capacity and the secondary and k-ary paths gets lesser and lesser capacity. Next the routing is done for each of the k paths. The terminating points of each path are selected as VP terminators. The assignment of VPs in this way may result in the total number of VPs exceeding the switching constraint at some switches. The problem is solved by selecting a switch where the switching constraint is violated and then combining a selected number of VPs at that switch. The combining of VPs, lead to alteration of routing tables at the intermediate switches. Also the VPs that are combined must share common VP terminators and intermediate nodes.

The performance analysis of the above approach indicates that selecting 2 or 3 multiple paths between the source destination results in lower blocking rate. However, the above approach results in a VP topology which is tuned to a particular traffic pattern. The topology is generated off-line and cannot adapt to changes in traffic pattern which may take place subsequently. For every major traffic pattern change, a new topology would be required which would result in interruptions. Thus in a real network the above approach might be too restrictive and unsuitable.

- In [7] approximate solutions are presented to optimize the use of virtual paths as it was shown that this problem is NP-Hard. The proposed method is based on a decomposition of the problem into separate stages. First, the VP terminator pairs are found using the approach based on *clustering.* The algorithm selects VPs (as cluster centers) which are farthest apart. The intuition here is that by selecting cluster centers (VPs) which are farthest apart, the overlay would cover the whole network. Then the nodes pairs which are close to the cluster centers are grouped into clusters. The cluster centers (VPs) are therefore to be selected in such a manner that they would result in an increased multiplexing gain and reduced connection setup cost. At each iteration a VP is selected which is found to be farthest apart from other selected VPs by using the formula:

$$D(x,y) = \lambda_{ij}(d(i,s) + \alpha d(s,t) + d(t,j)).$$

where $D(x,y)$ is the cost between the node pair $x = (i,j)$ from a selected VP, $y = (s,t)$, λ_{ij} is the traffic between the node pair $x = (i,j)$ (number of VCCs), $\alpha = 0.1$ and $d(...)$ is the minimum physical hop count between the node-pairs. The worst case time complexity of this algorithm is $O(n^4)$, where n is the number of nodes. The algorithm initially arbitrarily selects a node-pair and assigns a VP to it. After that the above equation is used to find a node pair which results in maximum cost $D(x, y)$ by routing its traffic through VP y. A new VP is then assigned to node-pair x which has the maximum cost $D(x, y)$. Now, in the subsequent iterations a new VP is assigned to such a node-pair whose cost $D(x, y)$ is maximum from all the selected VPs y. This process is repeated until all the VPs are generated.

Second, the VP routes are generated such that the maximum link load (number of VPs per link) is minimized. The algorithm used in this phase will randomly select a shortest path (from a list of available shortest paths) to route the VP between the node pairs. It is shown that such a scheme results in link distribution closer to the optimal link load distribution.

One clear advantage of clustering is that since it selects cluster centers (VPs) which are farthest apart it will cover the entire network efficiently and thus provides better connectivity. However, the selection of cluster centers does not take into account the overall effect (such as bandwidth fragmentation) of a particular selection of cluster center (VP) on the entire VP topology configuration. Also the time complexity of the algorithm may not be suitable for large networks. The performance of this scheme will be investigated further in Chapter 4.

- In [17], a dynamic reconfigurable VP overlay design algorithm is presented such that it produces reconfigurable VP overlay that can adapt to various traffic changes during the day. Efficiency is achieved as the VP topology is changed during the day in order to utilize resources according to the demand of traffic.

  The algorithm considers that the virtual circuits of a traffic class (similar traffic), are statistically multiplexed in a VP and different VPs are deterministically multiplexed. Bandwidth estimation of each QoS for different traffic pairs at different load periods during the day is estimated by using a procedure which takes as its input the number of connections in a load period and the traffic descriptor parameters. The next step is to solve the dynamic VP routing and network sizing problem. The algorithm starts by generating a few candidate VPs between each traffic pair for each traffic class and at each traffic load period. The problem is then solved by using optimization techniques.

The computational cost of algorithm, although quite reasonable, increase as the number of nodes in the network increases and the author uses only two traffic classes to base the performance improvement on. It is unclear if the same level of improvement can be obtained with a large number of traffic classes and how the computation cost would grow for very large networks. Also, the implementation details of switching over to a new overlay have not been discussed.

- In [18], the VP topology is generated so that it can provide a minimum loss of flow in cases of link failures. In ATM, where the data rates can be in gigabits per second, it is crucial to quickly recover from a network failure. This is necessary as even a small interruption could result in a significant lost of flow and degradation of service.

  The algorithm is developed to find a virtual path routing and bandwidth assignment that minimizes the expected amount of lost flow upon restoration from network failures. In ATM networks the success of a restoration process would largely depend on the traffic conditions and spare capacity distribution when the failure occurs.

  Therefore the following process needs to be provided in order to supplement the restoration process.

  - Dynamic VP routing/configuration: Given a traffic demand and available network resources, a flow assignment is determined. The spare bandwidth is allocated over the network in a way to facilitate the restoration process.

In other words, this process makes the flow assignment and distributes spare bandwidth in such a manner that it ensures survivability of network at a particular level. Accordingly if a link failure occurs the restoration process is guaranteed to succeed, because it would find the necessary spare bandwidth and restoration routes.

  - Network synthesis with survivability constraints: Given the projected traffic demand the link capacity should be placed optimally. Without an appropriate link capacity design the VP reconfiguration process cannot maintain survivability at the desired level.

The above two survivable network management mechanisms together with a fast restoration scheme, are expected to offer a highly reliable ATM network. Thus the general VP topology should be developed to increase the spare capacity on the links or, in other words, the number of VPs sharing a link should be small.

Other traffic dependent solutions are discussed in a survey paper[10] and in [8, 6].

2.1.2 Traffic Independent VPC Overlays

In this section methods are discussed which do not require a priori information about the traffic pattern.

- In [9], an understanding of an effective VP overlay structure is provided. In the paper the role of VPs are analyzed from the viewpoint of setting up a virtual private network (VPN). By allowing virtual subnetworking, the traffic can be subdivided into more homogeneous and manageable groups. The virtual subnetworking can be achieved by using VPs and since each virtual subnetwork can be privately owned and managed, it is termed as a VPN. The VPN can be constructed in two ways

  - End-to-End Virtual Private Network(EEVPN): Uses direct VPs between sites.

  - Broadband Virtual private Network(BVPN): Uses a concatenation of VPs between sites.

  The EEVPN can only support a small number of sites and causes bandwidth fragmentation. The BVPN has the advantage that a VP can be utilized by several connections en-route to their destinations. This increases the multiplexing gain. However, some ATM switches would now also need to do VP/VC switching.

  If the link capacities are high, the BVPN schemes favor topologies that result in greater statistical multiplexing. This can be achieved by having a low VP fan-in/fan-out at each node, as this would result in lesser bandwidth fragmentation. Thus the optimal topology for BVPN (from the viewpoint of lesser bandwidth fragmentation) would be the ring topology. On the other hand, in the EEVPN scheme, as there is no multiplexing within the VPs, topologies with shortest distances between source and destination are favored. Thus the optimal topology for EEVPN is a superposition of shortest trees.

  If the link capacities are smaller, then in the case of BVPN, topologies other than rings would result. The reason is that the VP layout may not find enough capacity on a single outgoing link from a node and, therefore, it may need several VPs from a node to satisfy the bandwidth requirements. Similarly, in the case of EEVPN, the topology may include paths other than the shortest path, as it may not be able to find the required bandwidth on the shortest path.

  Thus the BVPN scheme is more suitable as it provides a VP overlay which can result in low bandwidth fragmentation and therefore better utilization of resources.

- In [12] a mathematical model of a virtual path layout in ATM networks is developed using the concepts of graph theory. In the graph the vertices represent the nodes of the network and the edges represent the VPs. An algorithm is developed in which the VP layout is determined in a manner that allows the use of small VP routing tables as a trade off to the call setup performance.

The layout makes the following assumptions:

- Linear connection structure: it considers that VPs and VCs are setup in pairs for bidirectional communication.
- Full switching capability: it considers that every switch in the network can do both VP and VC switching.
- Pure routing: the layout does not consider multiple routes from the source to the destination.

The paper describes the construction of a $VPPL^{1-m}$ (one-to-many Virtual Path Pair Layout) over a tree network. This structure is then used for constructing a $VPPL^{m-m}$ (many-to-many Virtual Path Pair Layout) for a general network. Several other physical topologies (meshes, rings, bounded width trees) are discussed and the possible application of VPPL to these topologies are analyzed. In all the cases the algorithm provides a mechanism to adjust the load on routing table as a trade off for the setup cost. That is, for a given maximum VP hopcount value h, the algorithm generates a set of VPs, such that the link load is small and the maximum number of VP hops between any source and destination should be $\leq h$. The performance of this scheme will be investigated further in Chapter 4.

- In [26], the VPs are assigned to node pairs which are furthest apart. As the node pairs which are furthest apart would have the highest connection setup cost, assigning a VP between them will reduce the connection setup cost of the distant nodes.

  In this way the method reduces the diameter of the network. However, the number of VPCs sharing a link and therefore possible bandwidth fragmentation is not explicitly considered in the construction of the VP overlay. The performance of this scheme will be investigated further in Chapter 4.

2.2 Conclusions

In this chapter various methods are discussed and classified on the basis of whether they are dependent or independent of the traffic pattern in generating a VP topology. Each approach provides some properties to the generated VP topology and, based on these properties, the performance varies. In view of this discussion, we identify a set of useful properties in the next chapter and develop a method to incorporate these properties into a VP topology. In Chapter 4 and 5 the performance of our method will be compared with the algorithm [12] which will be referred to as *graph*, [26] which will be referred as *diameter* and [7] which will be referred as *clustering.*

Chapter 3

A Flexible VP Topology Design Algorithm: Assignment Method

3.1 Introduction

From the previous chapters an understanding has been gained about ATM networks and the problems involved in operating such a complex network. The virtual path is a concept used to reduce this complexity and make the network cost effective. However, not all VP topologies for a given network would be able to provide such a facility. Thus, an effort is made to understand what properties are desirable in a VP topology. In this chapter we identify several such properties and discuss their merits. Next, an algorithm is proposed which incorporates these properties in the generated VP overlay. The proposed algorithm utilizes the concepts used in operations research to solve assignment problems. Thus we have given our approach the name "assignment method". Since, we also need to find a physical route for each VP, a routing algorithm is proposed. The complexity analysis of the proposed algorithms are also presented.

3.2 Important Properties in a VP Topology

The approaches *clustering, graph* and *diameter* described in the last chapter have some distinct advantages over each other. The *clustering* approach may result in a better connectivity by assigning VPs which are spread throughout the network while, on the other hand, *diameter* might result in a lower connection setup cost topology as it establishes VPs between farthest apart node pairs. As compared to *clustering* and *diameter*, *graph* provides a more controlled mechanism of establishing a VP topology and it also attempts to incorporate useful properties (such as lower link load, bounded maximum VP hop) in the generated VP topology. However, it does not give the option to generate any prespecified number of VPs. This capability might be useful in cases where the physical topology changes and the algorithm can then be used to generate a prespecified number of VPs to lower the cost to a specific value. Instead of regenerating the entire VP topology each time there is a physical topology change, the algorithm should be capable of assigning VPs in a step-wise fashion.

From the above discussion, we have some understanding about the important properties of an effective VP topology. Our approach attempts to combine the advantages of all the above mentioned approaches and is therefore based on the following properties:

1 **Connection setup cost:** The VP topology should result in a reduced connection setup cost. This property can be achieved by establishing VPs to node terminators which are farthest apart, as all the intermediate nodes would be by-passed in establishing the connection. In other words this property attempts to reduce the number of hops in order to establish a connection.

2 **Physical links utilization:** This property requires the use of small physical links in routing a call. This can be achieved by establishing shorter VPs; this way the chances of routing a call which goes through many physical links is reduced. However, satisfying this property results in a tradeoff with property 1. For example if, in satisfying the above property, VPs are assigned to nodes which are farthest apart, then the node pairs which are adjacent or a few physical link hops apart are less likely to be connected by VP. Thus these node pairs may route connections using convoluted VP paths resulting in longer physical

routes and poorer physical link utilization. The above point is made with the assumption that only VP routing[1] is allowed. However, if VC routing[2] is also allowed then close node pairs can directly use physical links to route connections on shorter paths; while distant node pairs can still use VPs resulting in better physical link utilization and, consequently, better connection setup cost.

3 **Survivability:** This property can be achieved by keeping the link load[3] low. Thus, in case of a link failure only a small number of VPs would be rerouted using alternative links. Having a smaller number of VPs per link would improve the chances of finding the required bandwidth on alternative routes as well. This will correspondingly reduce the loss of flow due to readjustment of the VPs and will improve the resilience of the network.

4 **Resource utilization:** This property requires efficient utilization of resources by:

i) Keeping the link load small (in order to lower bandwidth fragmentation in the network).

ii) Using smaller VPs, so that the resources committed to each VP are small.

It can be seen that satisfying (ii) also satisfies (i) and vise-versa. Thus, this property is related to property 2 and 3 and satisfying this property also supports the above two properties.

5 **Availability of Alternative Paths:** This property requires that the VP topology should be well-connected. This property is important as the node pairs should have alternative paths available to establish connections and thus possibly lower blocking rate.

[1] Remember that by VP routing we mean that calls are routed using VPs only.
[2] Remember that by VC routing we mean that calls are routed using physical links only.
[3] Link load means the number of VPs sharing a physical link.

6 **Minimal reconfiguration:** The generated VP topology should require minimal reconfiguration over a long period of time. To incorporate such a property, some heuristics are used and in our case the VP terminators are selected in such a way that they are spread regularly throughout the network. This way the VP overlay would map the entire network efficiently and a large number of sources can economically route their traffic using the general VP topology. Hence the topology can remain effective for a long period of time.

7 **Flexibility:** This property suggests that VPs can be assigned in any prespecified number and the above mentioned properties would still apply to the generated VP overlay. An example of such a scenario would be changes in the physical topology of the network. Thus instead of regenerating all the VPs for the changed configuration, the algorithm should assign only a prespecified number of VPs to lower the cost to a specific level. The overlay generated this way should still follow the laid down properties of the algorithm. Also, the algorithm should be capable of easily handling the switching constraints[4] in an ATM network.

3.3 Problem Formulation

As was mentioned earlier, a VP is characterized by terminator nodes, the physical route and the allocated capacity. Our approach therefore considers the VP topology assignment as follows:

1 Selection of VP terminators to satisfy the previously mentioned properties.

2 Routing for the VPC, such that the maximum link load is reduced and the survivability of the network is improved.

[4]Remember that there are three types of switches (a)VP (b)VC and (c)VP/VC. The VP switch is a transit switch and no VP can terminate on it, all VPs must terminate on a VC switch. The VP/VC switch has the capability of both the switches

3 Capacity allocation to the VPC, such that the blocking probability is reduced for a given VP topology.

In this thesis, the first two steps are developed and the third step is left to future research.

3.3.1 Step 1: Selecting VP Terminators

The VP terminators are selected in such a manner that the previously mentioned properties can be satisfied. This results in a VP overlay which would lower the network operation, administration and management costs.

Let the network be represented by a graph $G = (V, E)$, where V is the set of nodes, E is the set of edges and $N = \mid V \mid$ is the number of nodes in the network. The problem of finding a suitable set of VP terminators can be solved by reducing the problem of finding VP terminators to the assignment problem. In the assignment method [25, 15], given an $N \times N$ matrix, where N is the number of nodes in the network, a subset of N elements in the matrix is determined such that the sum of the weights of these elements is maximal or minimal depending upon whether the objective function is a maximization or minimization. Thus if the number of required VPs M is equal to N, the assignment model can be used directly to generate N VPs with $2 \times N$ VP terminators. First consider the case where $M = N$.

The problem can now be formulated as:

Let

$$x_{ij} = \begin{cases} 1 & \text{if the node pair (i,j) is assigned a VP} \\ 0 & \text{if the pair (i,j) is not assigned a VP} \end{cases}$$

$$minimize \;\; z = \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij}$$

subject to

$$\sum_{i=1}^{n} x_{ij} = 1, \quad i = 1, 2, \ldots, n$$

$$\sum_{j=1}^{n} x_{ij} = 1, \quad j = 1, 2, \ldots, n$$

In order to solve the assignment model, the techniques for solving the transportation model [25, 15] can be used, but this approach would not be suitable as the assignment model results in degeneracy. The other more suitable approach is the Hungarian method [16, 15]. The resulting approach can optimally solve the $N \times N$ assignment model in a reasonable amount of time (see the complexity analysis below).

c_{ij} is the cost of establishing a VP between the terminators (i, j). The cost function reflects the required properties. For example property 1(connection setup cost) can be satisfied by selecting VP terminators which are farthest apart. This will result in reduced connection setup cost for the traffic routed between these terminators. To incorporate this property the cost function can be represented as

$$c_{ij} = 1/d_{ij}. \tag{3.1}$$

The parameter d_{ij} is the distance (minimum physical hop count) between the node pairs (i, j). By using Eq 3.1 as the cost function, the assignment model would generate a VP topology with VP terminators that are farthest apart (reducing connection setup cost (property 1)). However, by assigning long VPs we would also be committing a large number of resources to each VP and consequently it would also result in greater bandwidth fragmentation and higher link loads. Also, when only VP routing is allowed, optimizing property 1 may increase the average connection cost as the closer node pairs would have to use convoluted VP routes in establishing the connection. The cost function should therefore also support property 2 (physical link utilization) and property 4 (resource utilization). But, as mentioned earlier, satisfying property 4 would also support property 2 and property 3(survivability). Thus to include property 4, the cost function is now constructed as

$$c_{ij} = 1/d_{ij} + f(d_{ij})^{\beta}. \tag{3.2}$$

The term $f(d_{ij})$ in Eq 3.2 is a function of d_{ij} and represents the resources that would be committed if a VP is assigned between node pair (i, j); the term β is the weighting factor. By including the second term the assignment model selects VPs which would use the resources efficiently (see property 4 (resource utilization)). The weighting factor β can be used to adjust the relative importance of the resources as compared to the length of the VPs. That is, the higher the value of β, the smaller the average VP length and, consequently, lower resource (bandwidth) fragmentation and link load. In our algorithm we used Eq 3.2 as the cost function.

The physical topology is normally designed to provide alternative routes between the node pairs in case of link failure. Thus, to incorporate property 5 (availability of alternative routes) into the VP overlay, bidirectional VPs are established on each link of the physical topology and $c_{ij} = \infty$ is assigned between the adjacent node pairs (i, j) so that the algorithm will not place another VP there. Now the value of $M = TotalVPs - 2 \times TotalPhysicalLinks$. Thus the minimum value of $TotalVPs = 2 \times TotalPhysicalLinks$. We will refer to this initial overlay as the *BasicOverlay*.

We now state a general algorithm which considers the cases $M = N$ and $M \neq N$. In one iteration the algorithm will generate N VPs, where N is the number of nodes in the network. If the number of VPs $M < N$, a subset of M VPs would be selected from the generated N VPs, such that the cost value associated with the node pairs is minimum. If $M > N$ then the algorithm iterates $\lceil M/N \rceil$ times to generate the total of M VPs. Thus instead of assigning M VPs optimally, the problem is decomposed into assigning N VPs optimally (under the given constraints) at a time.

The following is the algorithm for our approach.

Algorithm I

1 Initialize $Gen = \phi$

2 $Gen = Gen \cup \{VPs\ in\ BasicOverlay\}$

3 Initialize the $N \times N$ assignment model matrix A, by assigning the c_{ij} values.

4 $M = TotalVPs - 2 \times TotalPhysicalLinks$

5 Determine the number of iteration as $I = \lceil M/N \rceil$

6 for i := 1 to I do

begin

initialize $S = \phi$ and $\acute{S} = \phi$

generate S as a set of VP terminator pairs (i, j) using the assignment model.

if $(M < N)$

select $\acute{S} = \{(i,j) | \acute{S} \subseteq S \ \ and \ \sum_{i=1}^{m} \sum_{j=1}^{m} c_{ij} \ \ is \ \ minimum\}.$

else

$\acute{S} = S.$

$M = M - N$

endif

$Gen = Gen \cup \acute{S}.$

assign $c_{ij} = \infty$ in A, if $(i, j) \in \acute{S}$.

end

In any single iteration of the assignment model, due to the constraints, each node will be a terminator in at most two distinct VPs. The constraints imposed on the nodes, therefore, allow the VPs to be spread throughout the network (see property 6-minimal reconfiguration). Once the VP terminators (or VPs) are generated, we would need to route the VPs. The following section presents the algorithm for routing the generated VPs.

3.3.2 Step 2: Selecting Routes for VPs

This section provides an implementation of the routing algorithm[5] for VP paths. A routing for the set *Gen* should be generated such that the maximum link load is

[5]There are four types of routing appeared in this thesis. For connection setup (as mentioned in Chapter 1) we will use the term VP routing, VC routing and VP+VC routing, to describe which medium is used in setting up the connection. For routing of VPs itself no special name is used and will simply be referred as routing of VPs. The algorithm discussed here is for routing of VPs

minimal. However, as is shown in [7], this problem is NP-complete. We, therefore, propose the following algorithms for finding a routing solution.

Algorithm IIa (below) is based on Dijkstra's shortest path algorithm and is adapted from [23, 22, 24]. The algorithm can choose paths based on shortest path, least loaded shortest path and least loaded path routing. The algorithm takes a node pair (s, d) and starting with s as the source node it generates a path to d such that the cost function D_d in minimum. The cost function D_d is computed by including the link load along the path and the number of hops taken from the source node s to d. If both the terms are included then the generated path is the least loaded shortest path, otherwise, it can be either least loaded path or the shortest path, depending upon which term is included. In the algorithm, W_{ij} reflects the load (the number of VPs that are using the physical link) on the link (i, j). However, we normalize this weight before including it in the cost function. Thus, W_{ij}^n is the normalized weight and varies from $(0, 1]$. The normalization factor (the number with which the weights are divided by) can be chosen as a very large integer or as $TotalVPs$, but since in the future the total number of VPs can be increased, it is safe to use a sufficiently large integer preferably $N(N-1)$. The above number is chosen because the $TotalVPs$ would always be less than or equal to $N(N-1)$. Thus $W_{ij}^n = \frac{W_{ij}}{N(N-1)}$. Initially all the links are assigned a weight $W_{ij} = 1$. Thus the W_{ij}^n for each link is computed as:

$$W_{ij}^n = \begin{cases} \frac{W_{ij}}{N(N-1)} & \text{if (i,j) is a link} \\ \infty & \text{if there is no link } (i, j) \end{cases}$$

H_{ij} is the value of the (i, j) element in the adjacency matrix and is initialized as:

$$H_{ij} = \begin{cases} 1 & \text{if (i,j) are adjacent} \\ 0 & \text{if } i = j \\ \infty & \text{for all other cases} \end{cases}$$

We now formally state the algorithm IIa. In the algorithm the source node s is assumed to be given number 0 as the identifier.

Algorithm IIa

1. Initialize
   $Path = \{s\}$
   for l := 1 to N-1 do
   begin
   - $D_l^H = (SP)H_{sl}$
   - $D_l^W = (LL)W_{sl}^n$
   - $D_l = D_l^H + D_l^W$
   - if $(H_{sl} = 1)$
     - parent(l)=s
   - endif

   end

2. Find $k \notin Path$, such that $D_k = \min_{j \notin Path} D_j$
   Add k to $Path$

3. Update cost
   if (SP) and (not LL)
   - for l := 1 to N-1 do
   - begin
     - $D_l^H = min\{D_l^H, D_k^H + H_{kl}\}$
     - if (D_l^H is changed)
       - parent(l)=k
     - endif
   - end

   endif
   if (LL) and (SP)
   - for l := 1 to N-1 do
   - begin
     - $D_l^H = min\{D_l^H, D_k^H + H_{kl}\}$
     - if $(D_l^H \geq D_k^H + H_{kl})$
       - $D_l^W = min\{D_l^W, max\{D_k^W, W_{kl}^n\}\}$
     - endif
     - $D_l = D_l^H + D_l^W$
     - if (D_l is changed)
       - parent(l)=k
     - endif
   - end

   endif
   if (LL) and (not SP)
   - for l := 1 to N-1 do
   - begin
     - $D_l^W = min\{D_l^W, max\{D_k^W, W_{kl}^n\}\}$

if (D_l^W is changed)
    parent(l)=k
endif
end
endif

4. if $|Path| = N$
    finish
   else
    goto step 2
   endif

The above algorithm works by generating the spanning tree such that the path found from s to d has the minimum D_d. The *parent(l)=k* simply assigns the node k as the parent of node l. LL and SP are boolean variables and by setting $LL = 0$, Algorithm IIa will find the shortest path; whereas by setting $SP = 0$, the least loaded path will be found between a source and destination. If both LL and SP are set to 1 then the algorithm generates a least loaded shortest path.

Algorithm IIb is now used to generate all the VP routes. The set *Gen* (see Algorithm I) holds the set of all the VP terminators. In this algorithm the path between VP terminators $(s, d) \epsilon Gen$ is represented as a set P, where $P = \{e_1, e_2, \ldots, e_n\}$ and $e_i \epsilon E$.

Algorithm IIb

Let M be the number of VPs in *Gen*.

for i := 1 to M

begin

Select a node pair (s, d) from *Gen*.

Using **Algorithm IIa** find the path P for (s, d) with a minimum D_d.

Assign P to (s, d) and increment the weights for $e_i \epsilon P$ by $\frac{1}{N(N-1)}$.

Remove the node pair (s, d) from *Gen*.

end

3.4 Complexity analysis

In this section, the complexity analysis of the algorithms of this chapter are presented.

3.4.1 Algorithm I Analysis

It is shown in [1, 7] that the generation of VP terminators, such that the resulting VP topology incurs maximum multiplexing gain and reduced connection cost, is a mixed integer programming problem and therefore NP-hard. However, in our approach instead of finding a globally optimal solution, the problem is decomposed into assigning N VPs optimally under the given constraints at a time. The assignment model, in each iteration of **Algorithm I**, takes a finite number of steps to satisfy the constraints and minimizes the objective function. We have used the Hungarian method to solve the assignment model. The run time complexity of Algorithm I can be given as:

$$O(N^2 i \lceil M/N \rceil)$$

The value i is the number of iterations the assignment model takes to find an optimal solution for generating N VPs (N is the number of nodes in the network) and $\lceil M/N \rceil$ is the number of iterations Algorithm I takes to establish M VPs. Although theoretically the value of i can be quite large, in testing the algorithm for large networks (N=1000) the value of i remains within a reasonable bound.

3.4.2 Algorithm IIb Analysis

The Algorithm IIb used Algorithm IIa which is based on Dijkstra's method. Thus for generating M shortest paths, where M is the number of VPs the run time complexity of Algorithm IIb can be given as

$$O(N^2 M)$$

Where N is the number of nodes in the network.

3.5 Conclusions

In this Chapter important properties of a VP overlay have been identified and we have presented an algorithm for designing a VP topology. The cost function and the structure of our algorithm incorporates the above mentioned properties. The first term in Eq 3.2 satisfies property 1(connection setup cost). However, due to the importance of property 4(resource utilization) the second term in Eq 3.2 is included. As mentioned earlier, satisfying property 4 would also support properties 2(physical link utilization) and 3(survivability). We will demonstrate this further in the experimental results. By establishing bidirectional VPs for each link in the physical topology, the VP overlay will inherit the connectivity property of the physical topology, thereby improving the availability of alternative routes between node pairs and satisfying property 5(availability of alternative routes). The special structure of our algorithm results in regular distribution of VP terminators throughout the network and would consequently result in satisfying property 6(minimal reconfiguration). The algorithm can also be easily adapted to various switching constraints in the physical topology by selecting appropriate cost values in the initialization of the cost matrix. Also, whenever there are physical topology changes, (for example, some nodes are added into the topology), the VP topology does not need to be regenerated. The algorithm can work from the already established set of VPs and assign only a prespecified number of VPs to lower the cost to a specific level. The entire set of VPs generated would still satisfy the laid down properties. Thus our algorithm also satisfies property 7(flexibility).

The routing algorithm presented is quite simple and yet by modifying its parameters its behavior can be altered thus making it more flexible. In the chapters to follow, we analyze the effectiveness of the properties that are incorporated into the VP topology generated by our approach and compare the results to some other approaches.

Chapter 4

Experimental Results

4.1 Introduction

In the previous chapter we have stated the important properties in a VP topology and have developed our algorithm to incorporate such properties. In this chapter we analyze the properties of VP overlays generated by *assignment* and other algorithms. The effect of routing with respect to *assignment* is studied. Also the physical topology is studied with a view to obtaining an understanding of how to efficiently set the parameters of our algorithm. The next chapter will give the performance evaluation of VP overlay by using a call-level simulator of an ATM network.

4.2 Analysis of Properties

In this section the analysis of VP overlay properties is given. The algorithms included for analysis are:

- *Diameter* based [26]: this algorithm establishes VPs between node pairs which are farthest apart in order to reduce the diameter of the network. We will refer to this method as *diameter.*

- *Clustering* based [7]: this algorithm creates cluster centers as VP which are farthest apart. The traffic between nodes is included in the calculation of distance. We will refer to this method as *clustering.*

- *Graph* based [12]: this algorithm generates a VP topology such that the maximum link load is low for a given maximum VP hop value. We will refer to this method as *graph.*

- *Assignment* model: this algorithm, as explained in the previous chapter, reduces the problem of VP assignment to the assignment model and produces a flexible VP topology. We will refer to this method as *assignment.*

The *graph* algorithm takes the maximum VP hop value and generates a VP topology. The other methods, including *assignment*, take the number of VPs as the input argument. Thus, once the VP topology is generated using *graph*, the same number of VPs are then generated using the other methods. In the experiments the maximum VP hop value is 2.

For *assignment* the experiments were performed based on β values 1 and 3. $f(d_{ij}) = d_{ij}$ in the cost function (Eq 3.2). The least loaded shortest path routing algorithm, as mentioned in step 2 of our method, is used for all the algorithms. Later, we study the *assignment* model for different VP routing algorithms. In order to compare the effectiveness of a VP overlay (based on certain properties) the following measures of interest are used:

- Average connection cost: The average connection cost gives a measure of efficiency of a network in establishing VCCs between the node pairs. It is defined later.

- Maximum link load: This is the measure of the maximum number of VPs sharing a common physical link. For an efficient VP overlay this measurement should be small.

- Average link load: This is the average VP load on the physical links and should also be small for an efficient VP overlay.

- Average hop count: This is used to give a measure of the delay in connection setup. For a good VP overlay this measure should be small.

- Average physical length per node pairs: This is used to give a measure of how efficiently the VCCs are routed.

- Average physical length per VP: This is used to give a measure of the characteristics of VPs (i.e. whether the VPs are long or small).

The connection cost between node pairs using only physical links (VC routing) is computed as:

$$C_{ij} = d_{ij}.$$

where d_{ij} is the minimum physical link hop count, between the node pair (i, j). In the case of only VP routing connection cost is computed as:

$$C_{ij} = \sum_{(l,m) \epsilon VP_{ij}^{path}} \gamma + \alpha d_{lm}$$

Where VP_{ij}^{path} is the minimum VP hop count path between node pair (i, j), l and m are the VP terminators of a VP in VP_{ij}^{path}. The value of γ is chosen to be 0.9 and α as 0.1. α is chosen due to the observation in [5], where it is shown that the use of VP reduces the processing cost by 90%. The γ is chosen to ensure that routing a call on a VP between adjacent nodes does not lower the connection cost as there are no transit nodes avoided in the connection setup. The average connection cost is computed as:

$$\bar{C} = \frac{\sum_{i=1}^{n}\sum_{j=1}^{n} \lambda_{ij} C_{ij}}{\sum_{i=1}^{n}\sum_{j=1}^{n} \lambda_{ij}}$$

If only VP routing is allowed, C_{ij} is the cost of routing a VCC on VPs only. If VP+VC routing are allowed, C_{ij} is the cost of routing a VCC using both VPs and physical links, whichever combination results in a minimum hop path to the destination.

In this section, the traffic is considered to be the number of VCCs between the node pairs. The experiments were performed both under uniform and non-uniform traffic pattern. In uniform traffic, the traffic pattern is considered as a normalized distribution of VCCs between all the node pairs, therefore, $\lambda_{ij} = 1$. In the case of non-uniform traffic pattern, λ_{ij} is chosen as random value between $(0-2)$ such that $\sum_{i=1}^{n}\sum_{j=1}^{n}\lambda_{ij}$ for non-uniform traffic is equal to the uniform traffic for a given network.

The following physical topologies are used:

Name	*Nodes*	*Links*	*Average degree*
EN-1	10	14	3.3
EN-2	18	27	3.278
EN-3	23	33	3.087
ARPA2	21	26	2.714
US	28	45	3.393
A-1	50	75	3
A-2	80	120	3
A-3	100	198	3.96

The network topologies with the prefix "A-" are artificial networks. The real network topologies are shown in Appendix A.

4.2.1 Performance Evaluation of Experiments Under Uniform Traffic

Once a VPC overlay is set up there is an option of either routing strictly on the VP overlay (using VP routing) or doing VP+VC routing. The *clustering* and *diameter* algorithms have been designed for the latter; while *assignment* and *graph* overlays are designed more for VP routing, although VP+VC routing can also be used. Experiments are run for both assumptions. We will consider the performance of each algorithm in turn. We also consider the case of using no VP overlay (this is VC routing only). To improve the readability of the graphs we have used lines instead of bars, therefore, a line joining two points should not be considered as an interpolation between the points.

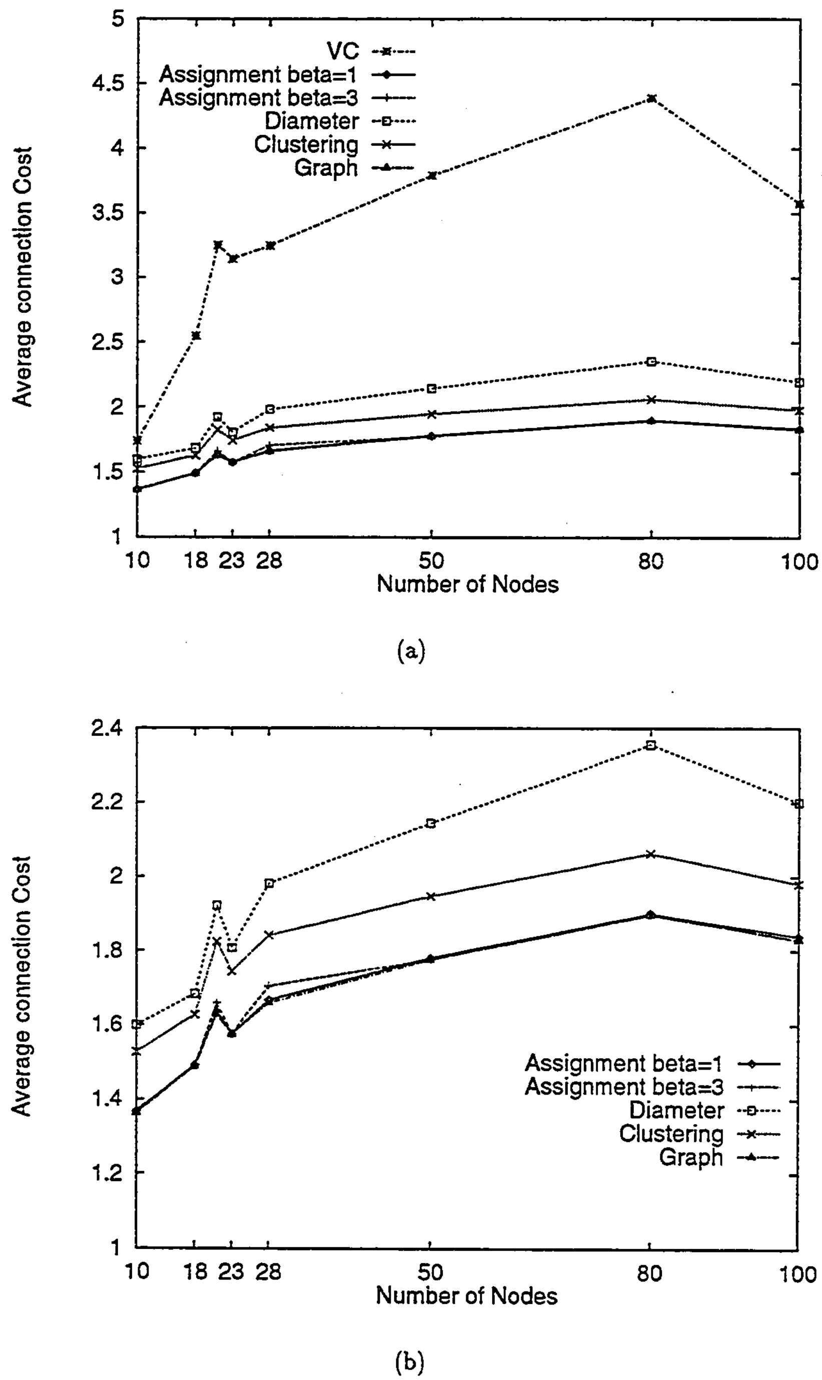
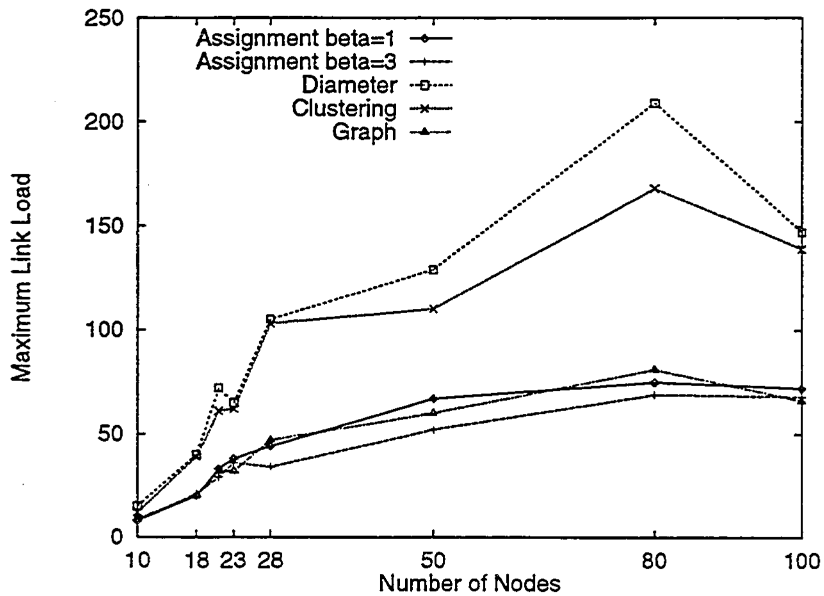
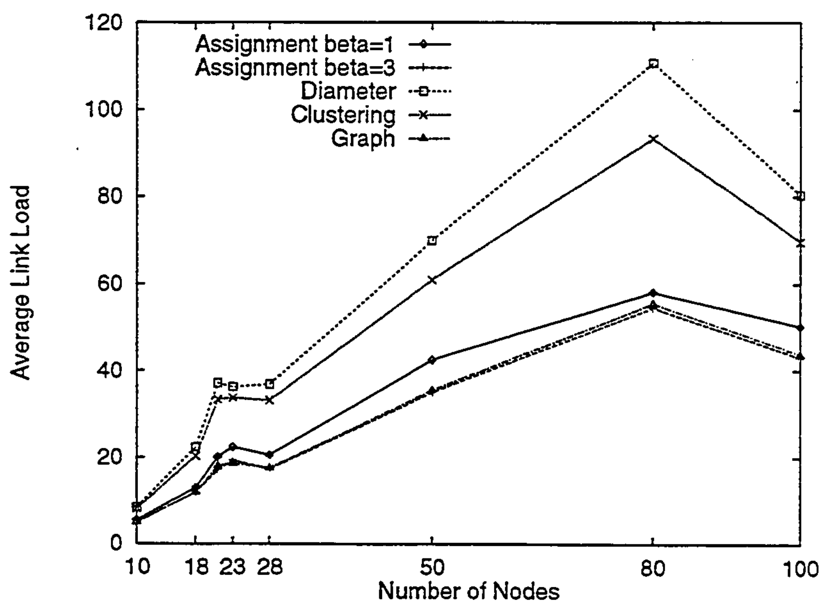


Figure 4.1: Average Connection Cost Analysis Under Uniform Traffic and VP routing: (a) Comparison with VC Routing (b) Using VP Routing Only

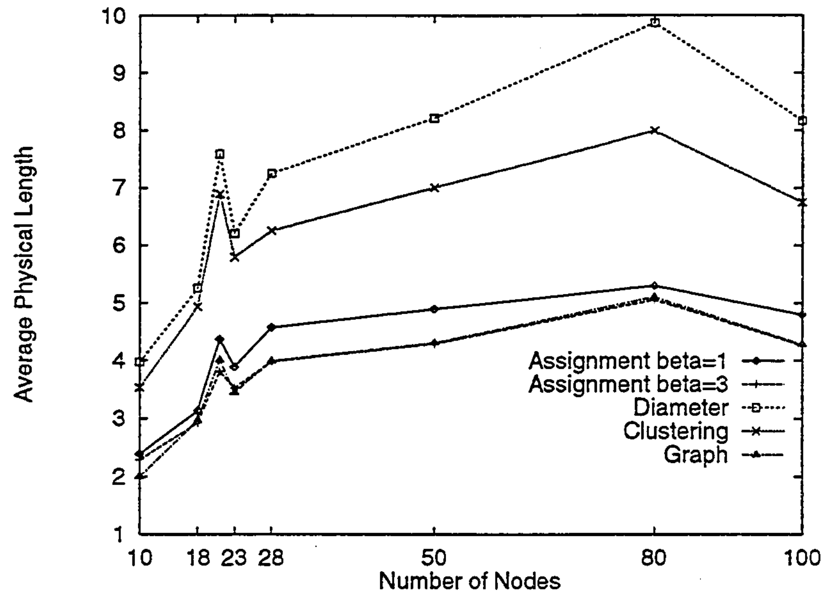


(a)

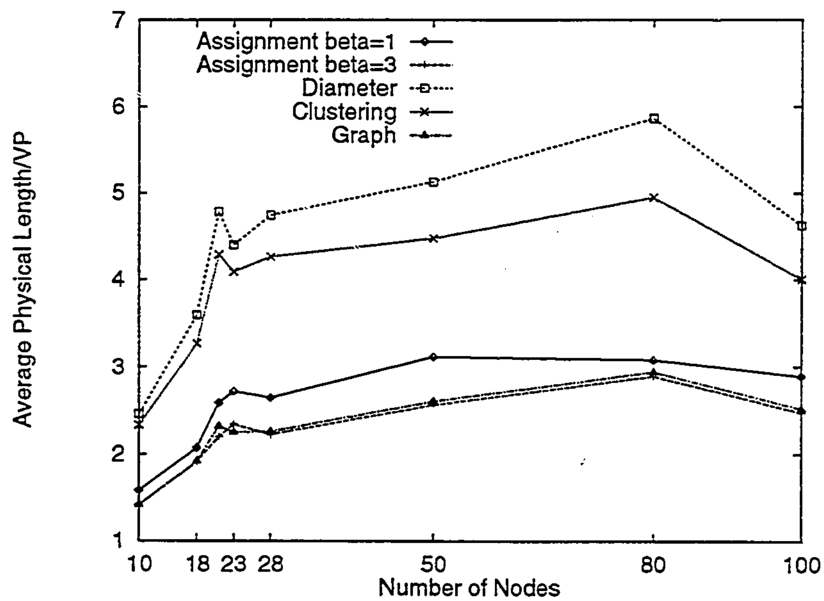


(b)

Figure 4.2: Load Analysis Under Uniform Traffic and Using VP Routing: (a) Maximum Number of VPs Sharing a Link (b) Average Number of VPs Sharing a Link



(a)



(b)

Figure 4.3: Physical Length Analysis Under Uniform Traffic and Using VP Routing: (a) Average Physical Length Between Node Pairs (b) Average Physical Length for Each VP

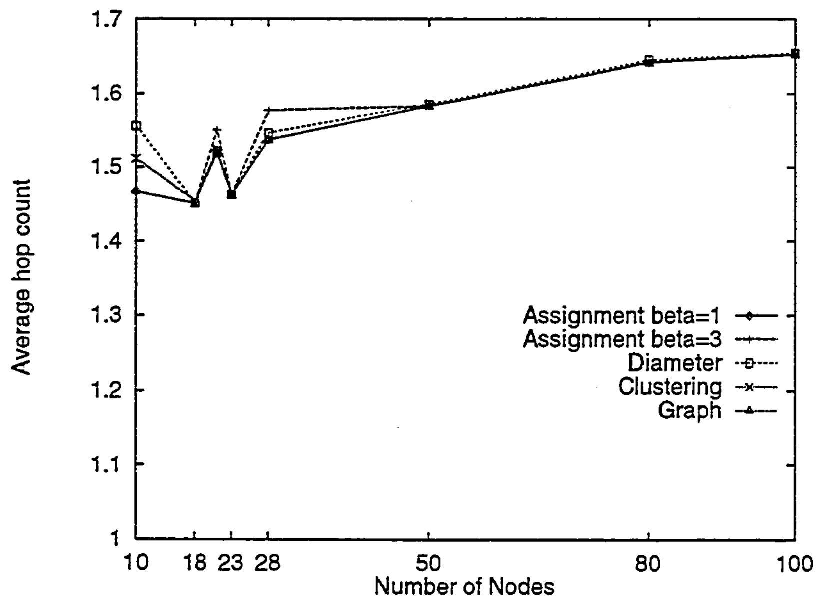
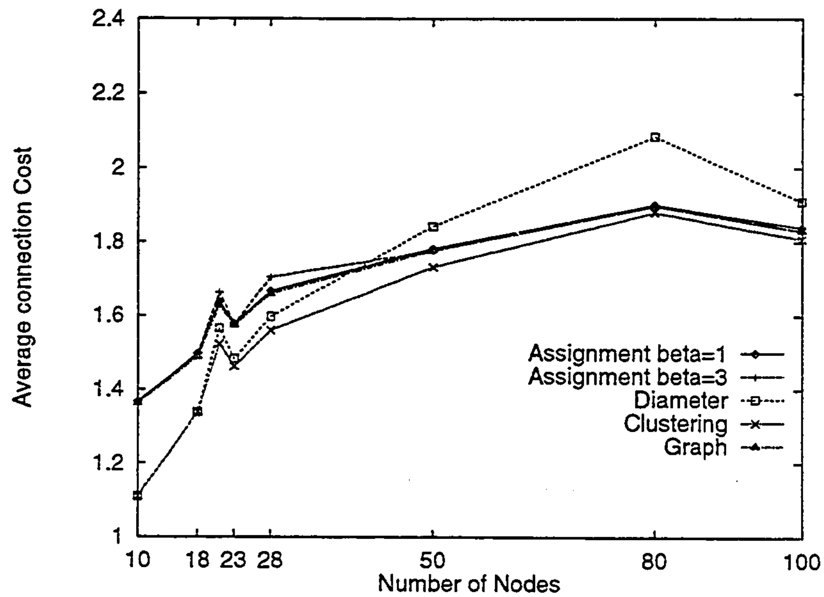


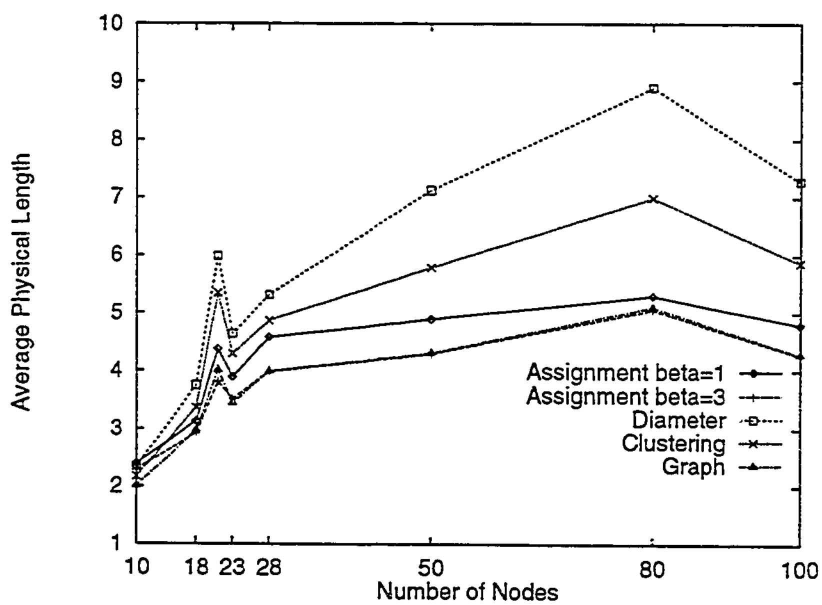
Figure 4.4: Average Hops Between Nodepairs Under Uniform Traffic and Using VP Routing

In Figure 4.1, it can be seen that all the methods succeed in reducing the average connection cost compared to VC routing. However, *diameter* and *clustering* result in higher maximum link load and average link load (see Figure 4.2). The high values for the maximum and average link load are due to the fact that *diameter* and *clustering* tend to select VP terminators that create long VPs and correspondingly increase the load on physical links. But *diameter*, since it optimizes property 1, always selects the longest VPs and therefore the load on physical links is greater than for *clustering.* However, the selection of a large number of long VPs by *clustering* and *diameter* adversely affects properties 2(physical link utilization),3(survivability) and 4(resource utilization). Since VP routing is used in this experiment and property 2 is not well satisfied, the average connection cost is also slightly higher. This is because some of the connections might take a longer physical route compared to the route they would take using only VC routing, consequently, increasing the average connection cost. This argument is further supported by the graphs of average physical length and average physical length per VP (see Figure 4.3). The average number of hops are similar for all methods (see Figure 4.4).

However, if VC+VP routing is allowed, then in the case of *diameter and clustering*, we would expect that the average connection cost, average physical length between node pairs and also the average hops to improve (see Figure 4.5-4.6). By allowing VC+VP routing the connections between node pairs which are close would be handled without using convoluted VP routes and nodes which are at a greater distance can still be reached in a single hop by using the long VPs. The average physical length between node pairs in Figure 4.5 is still high because, when a physical path(a set of physical links) and a VP path (a set of VPs or logical links) have the same minimum hop count, the routing algorithm could select the VP route which may have greater physical length than the physical route. Due to the above reason we can expect the average physical length to be high. Also, since *diameter* only selects the longest VP in order to reduce the diameter of the network, in VC+VP routing the chances of

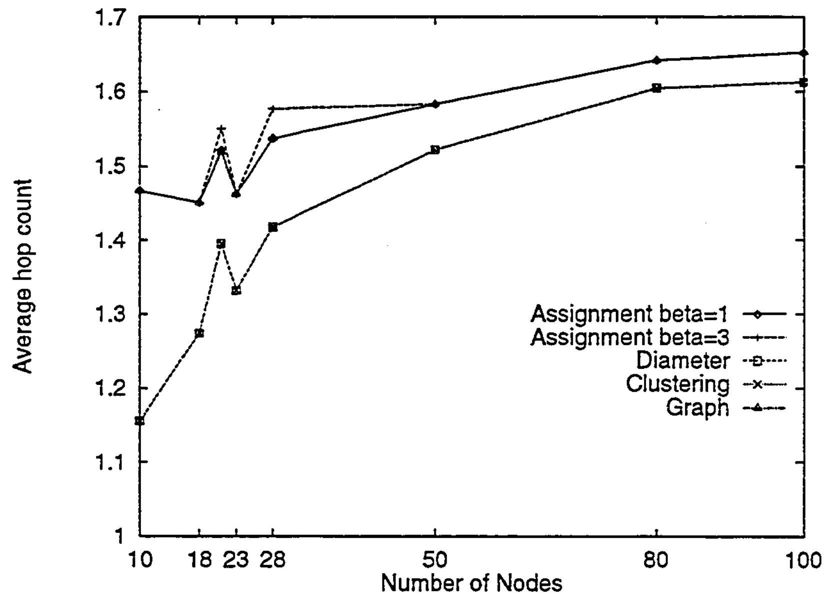


(a)



(b)

Figure 4.5: Analysis Performed Under Uniform Traffic and Using VP+VC Routing: (a) Average Connection Cost (b) Average Physical Length Between Node Pairs



(a)

Figure 4.6: Analysis Performed Under Uniform Traffic and Using VP+VC Routing: (a) Average Hops Between Node Pairs

finding a minimum VP hop path similar to physical link hop path is greater. Thus there is a greater chance for *diameter* to use longer physical paths, increasing the average connection cost. This effect will be more evident for larger networks and we see that the average connection cost for *diameter* for the larger networks is high under VP+VC routing as well (see Figure 4.5).

From the above discussion, it is clear that for *diameter* and *clustering*, which attempts to optimize property 1(connection setup cost), it is necessary that both VP+VC routing is allowed in order for them to be effective in terms of reducing average connection cost and average hops. The measurements of maximum link load, average link load and average physical length per VP remain the same under VP+VC routing as under VP routing, as they are related only to the VP topology irrespective of the route chosen for a VCC. These values can be improved by reducing the number of VPs established. However, this might adversely affect average connection cost and average number of hops. The *clustering* algorithm uses a processing-intensive algorithm to generate the VP overlay, and would have to regenerate the overlay if the physical topology changes. The *diameter* algorithm, however, has the potential to make successive VP overlay changes so that it can handle the physical topology changes.

The *graph* algorithm, as mentioned earlier, establishes a VP topology in a manner to minimize the link load for a given value of maximum VP hop. Thus *graph* optimizes property 4(resource utilization) for a given value of maximum VP hop. The optimization of property 4 would also lead to satisfying property 2(physical link utilization) and 3(survivability) as well. Hence, the maximum link load and the average link load (see Figure 4.2-4.3) are lower. Since this method does not attempt to optimize property 1(connection setup cost), VPs are distributed more uniformly between different node pairs (i.e. node pairs which are closer together would also get VPs). Thus for VP routing only, the node pairs which are close do not have to use convoluted VP routes to establish connections and therefore, the average connection cost is low (see Figure 4.1). Also by not optimizing property 1 the average connection cost and the average hops are not lowered any further when both VP and VC routing is allowed, as in the case of *diameter* and *clustering*. Due to its properties, this method is suitable when VP routing is allowed as no advantages can be gained if both VP+VC routing is allowed. The VP overlay generated by *graph* is closely tied to the existing physical topology, if there is a change in the physical topology the VP overlay would have to be regenerated. Although an appropriate number of VPs could be assigned to cover the new nodes and therefore delay regeneration of VPs [11], this would lead to a less ideal solution (in terms of maximum link load, average connection cost) and eventually requires the VP overlay to be regenerated. The *graph* algorithm also assumes that all the switches in the ATM network are VP/VC.

In the formulation of the *assignment* algorithm, property 4(resource utilization) is taken into consideration and therefore a second term is included in our cost function (Eq 3.2). If β is set to 0, *assignment* would result in optimizing property 1(connection setup cost). However, higher values would lead to satisfy property 4 more and consequently would also support property 2(physical link utilization) and property 3(survivability). For VP routing when $\beta = 1$, the established VPs are long and therefore in Figures 4.2 and 4.3 the link load and physical length measurements are higher than *graph*. However, by including the second term, the measurements in Figures 4.2 and 4.3 are lower than *clustering*. Also, due to the establishment of bidirectional VPs on each physical link, the average connection cost in the VP routing case does not increase even when $\beta = 1$ (see Figure 4.1). By increasing β to 3, the average physical length and the average physical length per VP decreases and, consequently, the maximum link load goes down even further. Thus, the second term controls the degree of satisfying properties 2-4 in the topology and should be kept high in order to get better link load and resource utilization. Since *assignment* establishes bidirectional VPs on each physical link (thereby improving property 5 (the overall connectivity of the VP overlay)) no change in performance is observed when VP+VC routing is allowed. The behavior of *assignment* can be easily changed by properly initializing the cost matrix. This makes it possible for *assignment* to adapt to different switching constraints and also due to the step-wise nature of *assignment*, the VP overlay need not be completely regenerated when there is a change in the physical topology. Thus due to the special structure of *assignment* it can satisfy property 7 (flexibility).

4.2.2 Summary of Results

In the above section the main characteristics of each algorithm are identified and based on that we can draw some conclusions about the effectiveness of an overlay generated by such methods. In the case where only VP routing is allowed, the *assignment* and *graph* algorithms gives better average connection cost. However, when VP+VC routing is allowed, the *clustering* algorithm gives slightly better average connection cost for all physical topologies compared to the other methods. The *diameter* algorithm, under VP+VC routing, gives comparable performance to *clustering* for smaller networks only and higher average connection for large networks. The improvement in average connection cost for *clustering* and *diameter* under VC+VP routing is due to the availability of physical links for closer node pairs to establish connections instead of using the convoluted VP routes when only VP routing is allowed. The average connection cost for both *assignment* and *graph* is lower under VP routing compared to *clustering* and *diameter*, because they include other important properties in the overlay and as a result distribute VPs more regularly so that the closer node pairs also find efficient VP routes.

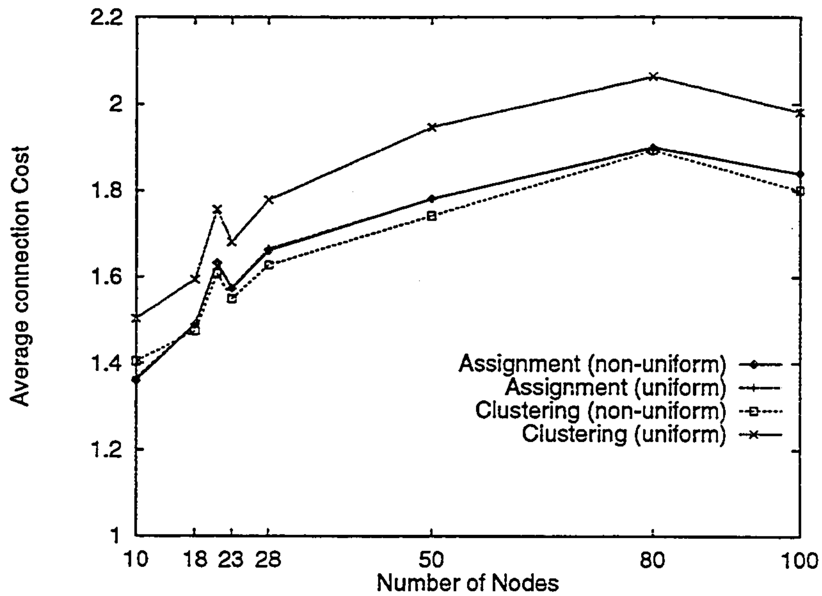
Under both VP and VP+VC routing, due to the properties of *assignment* and *graph* the maximum and average link load is significantly better than *clustering* and *diameter*. This makes *assignment* and *graph* more suitable for a long term VP overlay. Also, only *assignment* and *graph* guarantee the connectivity of all the node pairs using only the VP overlay. This is important when only VP routing is allowed.

4.2.3 Performance Evaluation of Experiments Under Non-uniform Traffic

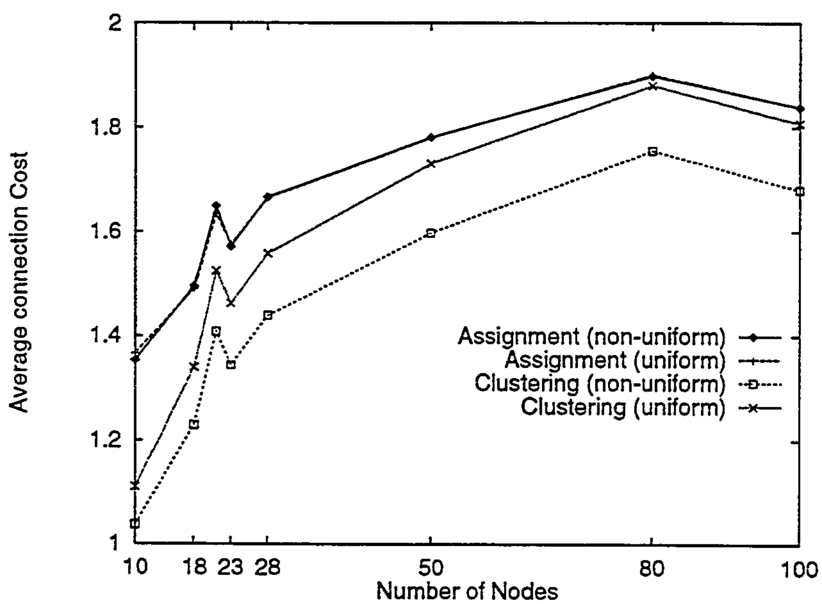
In this section, experiments are conducted under non-uniform traffic pattern. Multiple experiments are run with random non-uniform traffic patterns and 95% confidence intervals are found. The confidence intervals are small and, therefore, not shown to improve readability. In the experiments, after generating the topology for a non-uniform traffic the average connection cost is computed and then without regenerating the topology the traffic matrix is replaced by a uniform traffic matrix and the average connection is computed again. This is done to observe the affect of changes in traffic on the performance of the VP topology. All the methods, except *clustering*, are traffic independent. Thus the VP topology generated by the other methods would not be affected by changing the traffic pattern (i.e. the topology is the same for uniform and non-uniform traffic) while the *clustering* algorithm generates a VP topology which is tuned to a given traffic pattern (in this experiment, the non-uniform traffic).

In Figure 4.7, it can be seen that the average connection cost of *assignment* remains almost the same regardless of the traffic pattern; while *clustering* provides noticeably different average connection cost. When VP+VC routing is allowed the average connection cost for both uniform and non-uniform traffic for *clustering* remains below the average connection cost of *assignment*, but the maximum and average link load for *clustering* are high (similar to the characteristics observed in the previous section). Also, the fluctuation in performance would be undesirable in a VP overlay due to the performance sensitive nature of the traffic serviced by ATM.

The consistency in performance of traffic independent methods is due to the fact that they will distribute VPs more regularly throughout the network and thus the resulting topology can be deployed for a longer period of time satisfying property 6(minimal reconfiguration). For example, if the traffic is concentrated in one area, then in the case of traffic dependent methods, more VPs will cover that area; a change in traffic to less concentrated areas will result in very poor performance. This



(a)



(b)

Figure 4.7: Average Cost Analysis Under Non-Uniform Traffic: (a) Average Connection Cost Using VP (b) Average Connection Cost Using VP+VC

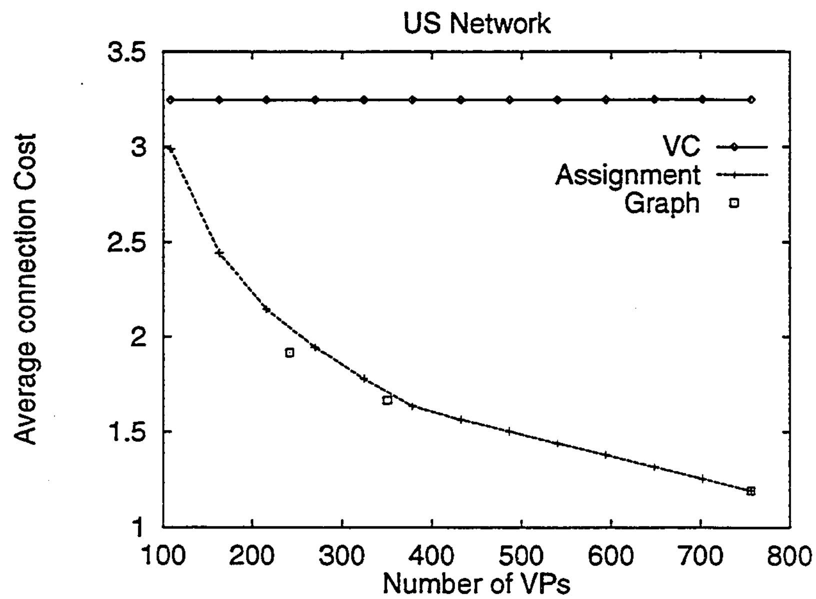
situation is avoided under traffic independent methods and therefore gives consistent average connection cost under varied traffic patterns.

Since *clustering* constructs a VP topology based on a specific traffic pattern, a change in the initial traffic pattern may cause a change in performance of the VP topology. This would lead to the need for reconstruction of the VP overlay based on a newer traffic pattern possibly causing interruptions. The *assignment* algorithm distributes VPs more regularly throughout the network and thus the resulting topology can be deployed for a longer period of time satisfying property 6(minimal reconfiguration). The average connection costs using *graph* and *diameter* also remains almost the same under non-uniform traffic and are therefore not shown here.

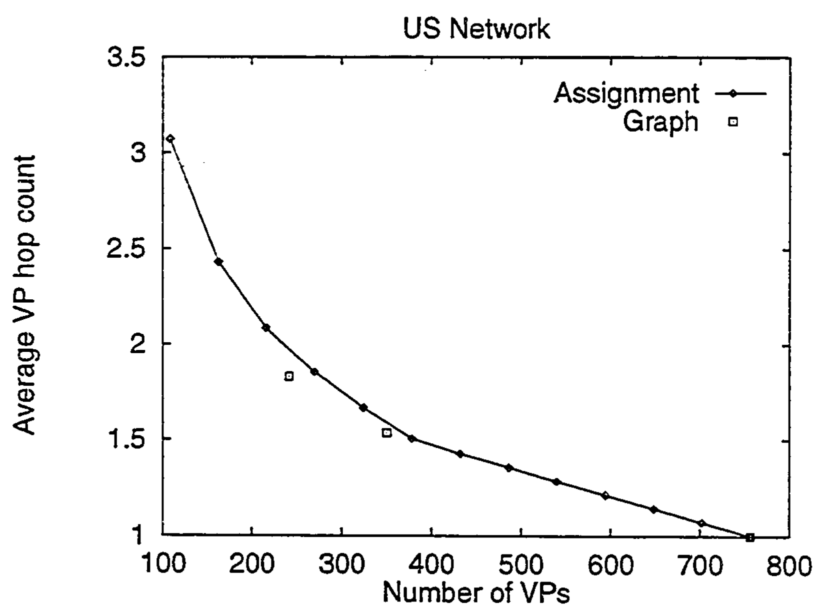
The graphs of the other measures have similar characteristics compared to the uniform traffic experiments and are also not shown here.

4.3 Analysis of the Number of VPs

In the previous section, the number of VPs used in *assignment* is the same as the number of VPs generated by using the *graph* method for maximum VP hop value $h = 2$. However, here we highlight the flexibility of *assignment* in selecting any prespecified number of VPs to bring the cost to a specific level. We have used the US (28 node) physical topology to perform this analysis. For *graph*, the points in Figures 4.8-4.9 represent the number of VPs established for maximum VP hop counts $h = 1, 2$ and 3. A higher number of VPs correspond to lower values of h and, as can be seen, there is a significant difference in the number of VPs between each value of h. The *assignment* algorithm can be used to establish VPs between the values produced for *graph* with $h = 1$ and 2 to further bring the cost down, without significantly increasing the maximum and average link loads(see Figure 4.8-4.9). With *graph* this flexibility is not available. The lowest connection cost achievable using *graph* is given by $h = 2$. For $h = 1$, it establishes a fully connected VP topology which is not feasible for large network. Also, it can be seen that *graph* produces lower maximum

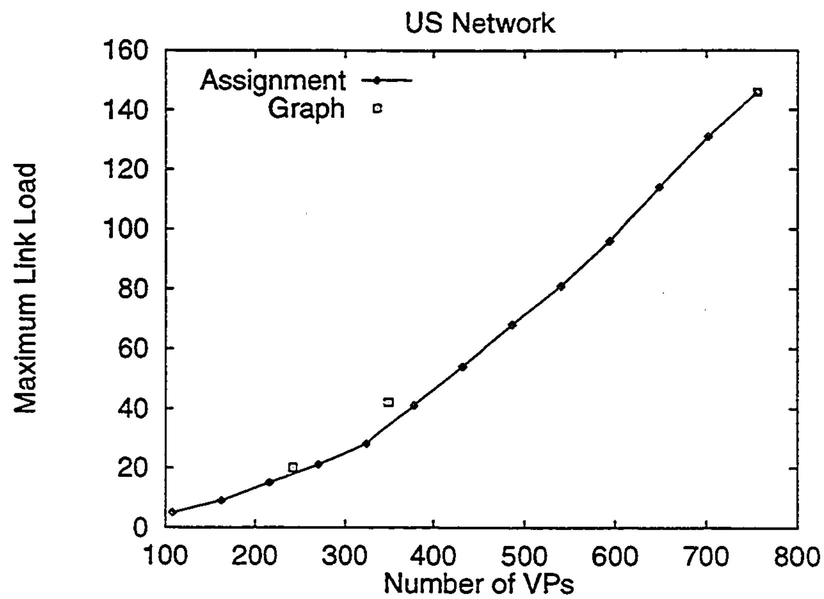


(a)

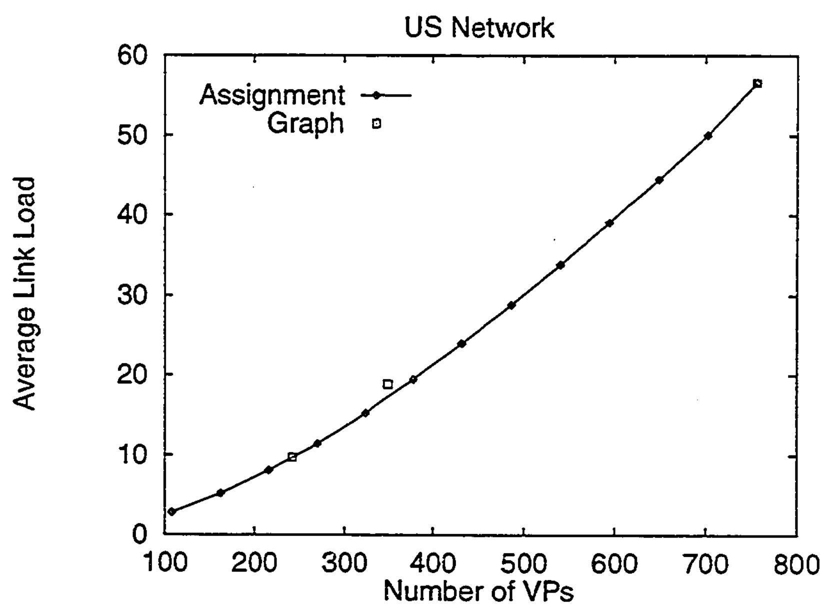


(b)

Figure 4.8: Analysis of the Number of VPs: (a) Average Connection Cost (b) Average VP Hop



(a)



(b)

Figure 4.9: Analysis of the Number of VPs: (a) Maximum Link Load (b) Average Link Load

and average link load for $h = 3$ at the cost of higher average connection cost. Again in this case *assignment* can be used to establish VPs between $h = 2$ and 3 to precisely control how much connection cost can be increased in order to lower the link load. Thus *assignment* provides better tradeoff control than *graph*.

4.4 Analysis of Physical Topology

In this section a procedure is developed, taking into account the physical topology, for giving the number of VPs which should be generated using *assignment* in order to bring the average VP hop to a specific level. The procedure uses exponential and inverse-exponential functions to approximate the cost function for *assignment* and uses greedy heuristics to approximate the assignment procedure. The procedure then takes the required average VP hop performance and iterates (emulating *assignment*) until the desired performance is achieved. It then outputs the number of VPs that are needed to obtain the required average VP hop count.

4.4.1 Approximating the Cost Function

The cost function of *assignment* assigns weights between node pairs making them more or less likely to get a VP. When β is high[1] the node pairs which are closer together are made more likely to be selected by *assignment*. On the other hand, when $\beta = 0$, the node pairs which are farthest apart are more likely to be selected by the *assignment*. A similar affect of β is demonstrated in Section 3.2.

To see how the weights are distributed for different distances (physical hop count) by *assignment*, we construct the normalized weight for each distance using the following equation:

$$W_{d_{ij}} = \frac{\min\limits_{\forall i,j | d_{ij} \neq 1} \{c_{ij}\}}{c_{ij}}.$$

[1]A value of 4 is sufficiently high for all cases

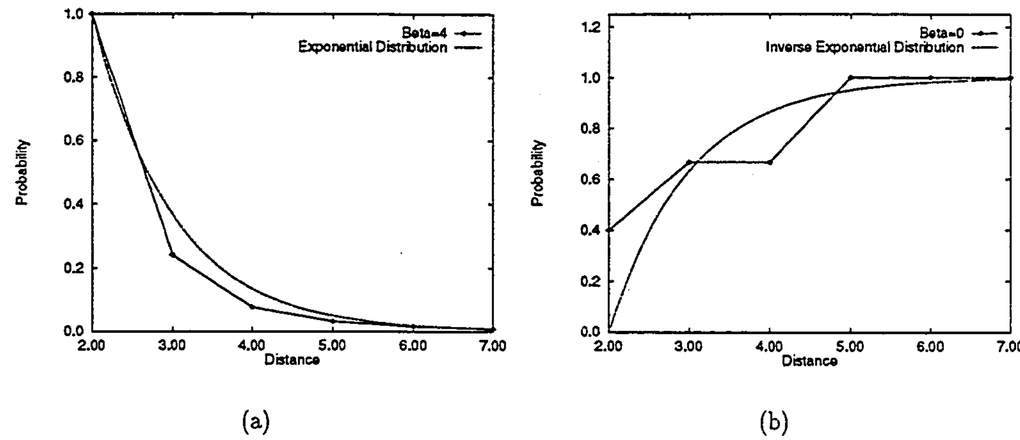


Figure 4.10: Analysis of Physical Topology: (a) Distribution of Weights for High Beta (b) Distribution of Weights for Low Beta

$W_{d_{ij}}$ is the normalized weight for each distance (minimum physical hop count) d_{ij} between node pair (i, j) and c_{ij} is the cost value generated by *assignment*. The same distances will have the same $W_{d_{ij}}$ value. Also, since all the single physical hop length VPs will always be assigned in the *BasicOverlay*, the weights for single physical hop VPs will be ignored. The distribution of weights generated this way gives the relative importance of a node pair getting a VP given their physical distance.

In Figure 4.10, we have the plotted the normalized weights (using the above equation) for each distance when $\beta = 4$ in the *assignment* cost function. The characteristics of the distribution follows a exponential function e^{-x}, where $x = 0, 1 \cdots max(d_{ij}) - 2$. Similarly, the normalized weight distribution for $\beta = 0$ is given in Figure 4.10. In this case the distribution is loosely approximated by the inverse of an exponential function.

Thus, given a physical topology, we can approximate the *assignment* weight distribution for low and high values of β by using exponential and inverse exponential distributions. To approximate the weight distribution when β is in between the two extremes we can use linear interpolation. In order to do the interpolation, first a given β_i value is normalized by using the following Eq:

$$\beta_i^N = \frac{\beta_i + 1}{\beta_4 + 1}.$$

The numerator gives the current level of β_i and the denominator gives the total levels of β. Now the normalized weight distribution for β_i can be interpolated using the following Eq:

$$W_{dij}(\beta_i) = (1 - \beta_i^N)(1 - e^{-x}) + \beta_i^N e^{-x}$$

Once the distribution is computed for a particular β_i, a matrix is generated by indexing the d_{ij} (for all node pairs (i,j)) value in the distribution and initializing the (i,j) cell with the indexed (W_{dij}) value. The matrix simply gives the relative importance of selecting a particular node pair as VP terminator for a given β_i. Also, all the single physical hop entry in the matrix are initialized to 1 to represent the *BasicOverlay* generated by Algorithm I. Thus, we can approximate the cost function of *assignment* by using the above procedure.

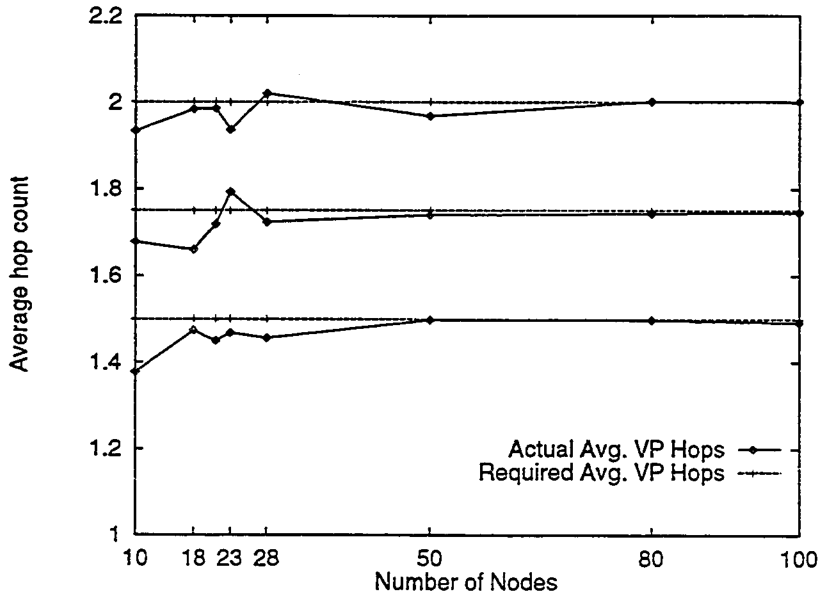
4.4.2 Approximating Assignment

The *assignment* can now be approximated by using a greedy heuristic. The rows of the matrix are scanned (left to right) to find the largest W_{dij} node pair and that node pair is selected as the VP terminator. The W_{dij} of a selected VP terminator is then set to 0 in the matrix so it will not be chosen again. Each row is scanned in a similar manner such that:

$$\sum_{i=1}^{n} x_{ij} = 1, \quad i = 1, 2, \ldots, n$$

$$\sum_{j=1}^{n} x_{ij} = 1, \quad j = 1, 2, \ldots, n$$

The above process is repeated until after an iteration I, the specified average VP hops can be achieved using the generated VP terminators. The procedure then gives a value equal to $I \times N$ as the number of VPs that should be generated by using *assignment* in order to achieve the desired average VP hops.



(a)

Figure 4.11: Assignment Satisfying the Required Performance

The above procedure approximates the cost function of *assignment* by using exponential distributions and the assignment procedure by using greedy heuristics. The greedy heuristics does not attempt to find the optimal local solution for $N \times N$ variables and therefore is only an approximate for *assignment.* Since the above procedure does not attempt to optimize, it is fast and can be used as a preprocessor of *assignment.* Thus, before using *assignment* an estimate of the number of VPs (in order to achieve a specific average VP hops) can be generated.

In Figure 4.11, the required average VP hop is plotted against the actual average VP hop resulting from using *assignment* with $\beta = 3$. The number of VPs is computed by using the above mentioned procedure and then generated by using *assignment.* From the figure it can be deduced that the procedure provides a good estimate for the number of VPs for *assignment,* since the actual average VP hops is sufficiently close to the required VP hops.

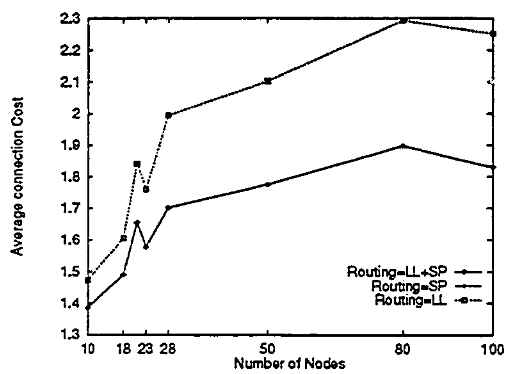
Apart from the above procedure we have also looked into characteristics of a physical topology and how it would affect the *assignment* model. In this study we have looked into the relationship of sparsity and correlation of physical links on the number of iterations (i, see the complexity analysis in Chapter 4) of assignment of N VPs. The experiments were performed for a number of varied topology and no relationship was determined among sparsity and correlation of links to the number of iterations.

4.5 Analysis of Routing for VPs

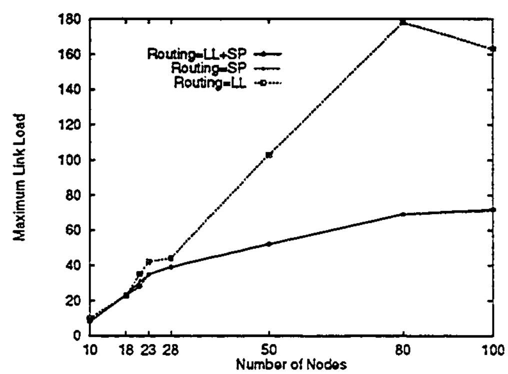
The routing algorithm for VPs can do three types of routing (*least loaded shortest path*, *shortest path* and *least loaded path*) as explained in the last chapter. In this section, we compare these algorithms for the *assignment* algorithm. In Figure 4.12, it can be seen that the shortest path and least loaded shortest path routing have almost identical performance. However, least loaded routing gives unacceptable results. The least loaded routing is not suitable because the *assignment* cost function uses the distance(shortest path) between the node pairs in making decisions to select VP terminators. Thus, the least loaded routing violates the assumption, on which *assignment* generates the VP terminators. On the other hand, the shortest path and the least loaded shortest path routing selects the shortest route between the source and destination and therefore they do no violate the assumption taken in the cost function of *assignment.* Thus shortest path and the least loaded shortest path routing are more appropriate routing approaches for VPs generated using the *assignment* algorithm.

4.6 Conclusions

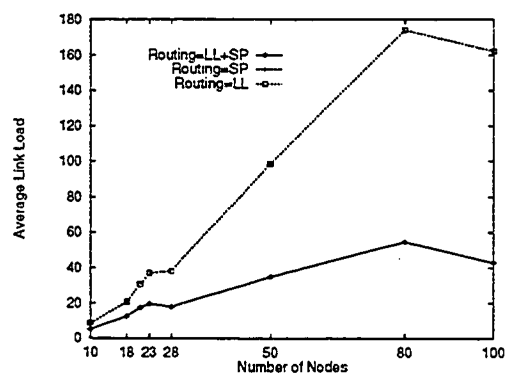
The properties of VP topology generated by *assignment* meet the important properties identified in the previous chapter. Our approach can easily adapt to various



(a)



(b)



(c)

Figure 4.12: Analysis of Routing:(a) Average Connection Cost (b) Maximum Link Load (c) Average Link Load

topological constraints and is also flexible in terms of establishing a prespecified number of VPs. We have also developed an approach to give an estimate of the number of VPs, using the physical topology and the β value, in order to bring the average VP hops to a specific level. The procedure gives our approach additional flexibility to quickly get a good estimate of the number of VPs needed in order to get a certain performance level. This gives our approach a similar advantage to the *graph* algorithm in which the user can specify a maximum hop value and then an appropriate number of VPs are generated. On the other hand, new VPs can also be added easily without redoing the whole topology.

Chapter 5

Simulation Results

5.1 Introduction

In this chapter the analysis of call-level simulation results are given. The simulator was created by researchers at the University of Waterloo. In this simulator calls arrive at ATM source nodes and decisions are made to accept or reject them based on the availability of bandwidth on the paths to their destinations. Calls are routed on VPs only (VP routing) and the route selected is the shortest hop path with enough capacity to service the call. If no such route exists the call is rejected. The simulator generates calls according to a specified traffic pattern and the efficiency of the underlying VP overlay is measured in terms of the number of calls being blocked and the average VP hops taken by the call to reach the destination. In the simulation, a dynamic capacity allocation scheme is used to assign and adjust capacities on the VPs according to the traffic pattern.

In this chapter we highlight that the *assignment* can provide effective service in real networks and also some advantages can be gained due to its flexibility in selecting the number of VPs. We also give results on the importance of the properties on which a VP overlay is based and demonstrate that not all VP overlays could be effective for a network.

5.2 Simulation Experiment Setup

In order to compare the effectiveness of a VP overlay (based on certain properties) the following measures of interest are used:

- Average loss rate: This measure gives the percentage of calls blocked.
- Average VP hops for a connection: This is used to give a measure of the delay in connection setup.

For the simulation, the VPs are assigned the same initial capacity. The capacities of the VP are then changed during the operation of the simulation by using a dynamic bandwidth control algorithm based on [3]. In this algorithm, when the spare bandwidth in a VP falls below a predetermined *threshold* level, the VP requests more bandwidth. The amount of bandwidth incremented is dependent upon a prespecified *step-size* value. Also, when a VP has spare capacity greater than the *threshold+step-size* level, its capacity is reduced by *step-size* but the capacity of a VP never falls below its initial assigned capacity. At any instance the sum of all the bandwidths of VPs on a physical link is within the bandwidth alloted on the physical link. A VP overlay is created for each class of service. Therefore only a limited capacity would be available to the VP overlay out of the total 622Mbps on each physical link. The capacity available to the VP overlay on each physical link is computed by multiplying the initial capacity of VPs with the maximum number of VPs sharing a link.

In this experiment only one class of service (1 VP overlay) is considered. The metric used for routing a connection is minimum hop count. That is, the connection is established on the minimum hop count path with enough capacity. A call would be blocked if the routing algorithm cannot find a set of VPs with adequate bandwidth to its destination. When the bandwidth of a VP is adjusted, the VP is locked for a prespecified amount (*delay*) of simulated time and cannot be used by any call during that period. This is done to simulate the processing time needed by the dynamic bandwidth allocation algorithm.

The number of source pairs and the traffic characteristics are also specified. The overlay is considered to be for one class of traffic so the traffic characteristics of all the source pairs are kept the same. The traffic characteristics are specified by the *call interarrival time*, *call holding time* and the *call peak rate*. In our simulation experiments the first two values are given as the mean values of a geometric distribution, while the peak rate is a deterministic value. The source pairs are uniformly distributed throughout the network. In our experiments we have considered a 28 node (US) network and 50 node (A-1) network.

The simulation parameters are initialized as follows:

- The VPs are assigned *initial bandwidth=2Mbps.*
- For the capacity reallocation algorithm [3], *threshold=1.5Mbps*, *step-size=2Mbps* and *delay=1 tick.*
- For the 28 node network, the traffic and physical topology parameters are initialized as:
  - Number of source pairs = 100. Note that this does not include all possible source pairs.
  - *call holding time=150 ticks*, *call peak rate=1.5Mbps* and *call interarrival time* is varied from 35-50 ticks.
  - The physical link *bandwidth = 57Mbps* .
- For the 50 node network the traffic and physical topology parameters are initialized as:
  - Number of sources = 150.
  - *call holding time=350 ticks*, *call peak rate=1.5Mbps* and *call interarrival time* is varied from 70-102 ticks.
  - The physical link *bandwidth = 70Mbps.*

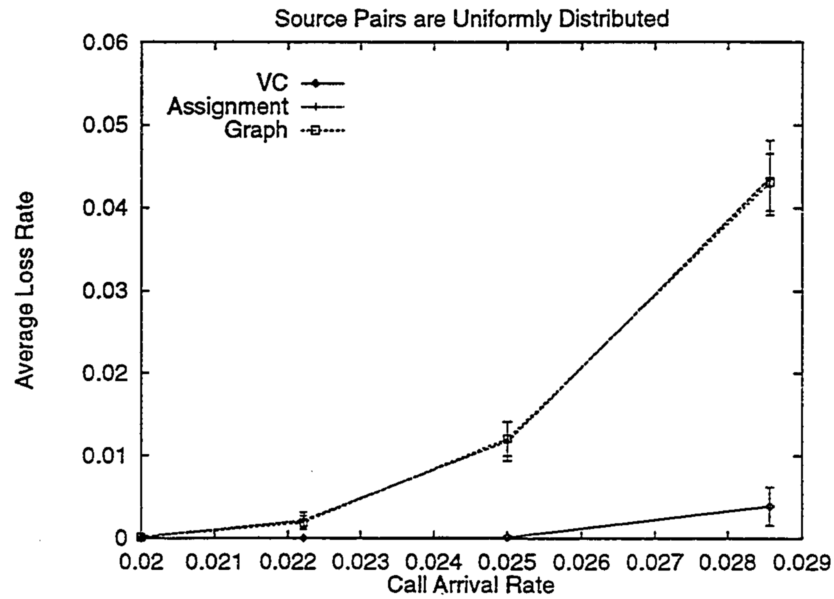
The call parameters are dimensioned to keep the loss rate within 6%. The capacity of physical links is determined by multiplying the maximum link load of a VP overlay with the initial capacity of VPs.

The simulator only routes calls on the established set of VPs. So, it is important that the VP topology would result in a single component graph of the network. The *assignment* and *graph* always result in a connected VP overlay, but this cannot be guaranteed for *clustering* and *diameter*. Also, *clustering* and *diameter* algorithms are designed to support both VC+VP routing of calls; while the simulation considers only VP routing when a VP overlay is used. However, some obvious points related to *clustering* and *diameter* can be drawn here. Due to their maximum link load the minimum physical link capacities required for these methods are 116Mbps for US (28 node) network and 129Mbps for A-1 (50 node) network. Thus if the same physical link capacity is given to *assignment* and *graph*, due to their lower maximum and average link load it would result in lower bandwidth fragmentation and consequently could provide lower call blocking rate than *clustering* and *diameter*.

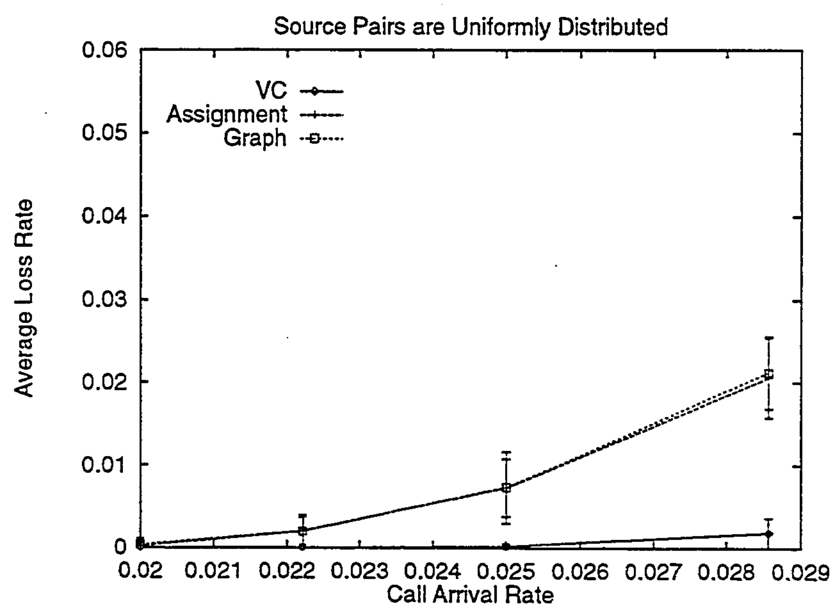
We, therefore, consider the *assignment* and the *graph* algorithm for the simulation experiment. The VP topology generated by *assignment* uses the value of $\beta = 3$. The results for VC routing are also generated. In this case each physical link can be thought of as a bidirectional VP with its capacity equal to the capacity of the physical link.

5.2.1 Performance Evaluation of Experiments

For the experiments multiple runs are made and 95% confidence intervals are found. In Figure 5.1 two different sets of uniformly distributed source pairs are used for the US (28 node) network and the average loss rate is plotted for the VP overlays (*graph* and *assignment*) and for VC routing. In Figure 5.2, the loss rate is plotted only for the overlays for A-1 (50 node) network. When using VC routing, the whole bandwidth of the physical link is at its disposal, therefore this results in



(a)



(b)

Figure 5.1: Loss Rate Analysis for US (28 node) network: (a) - (b) Sources Pairs are Uniformly Distributed

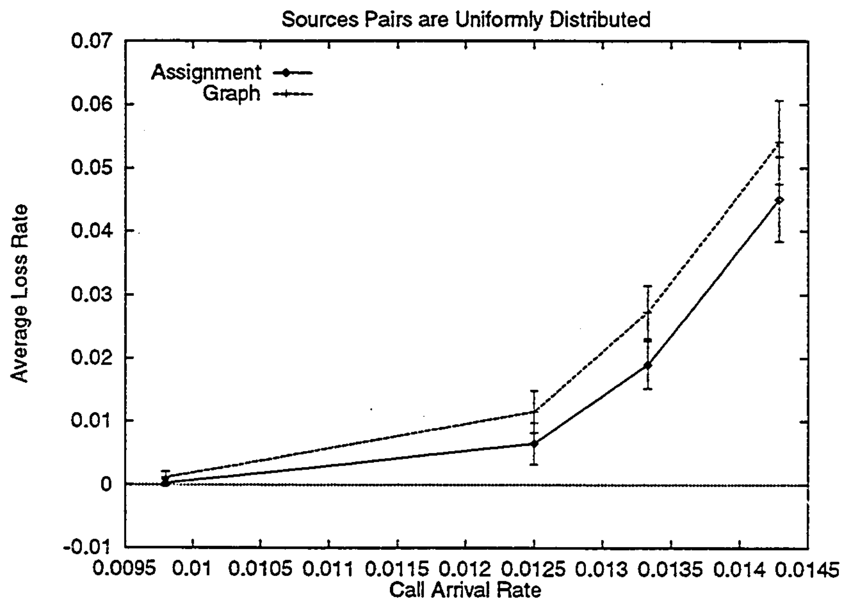


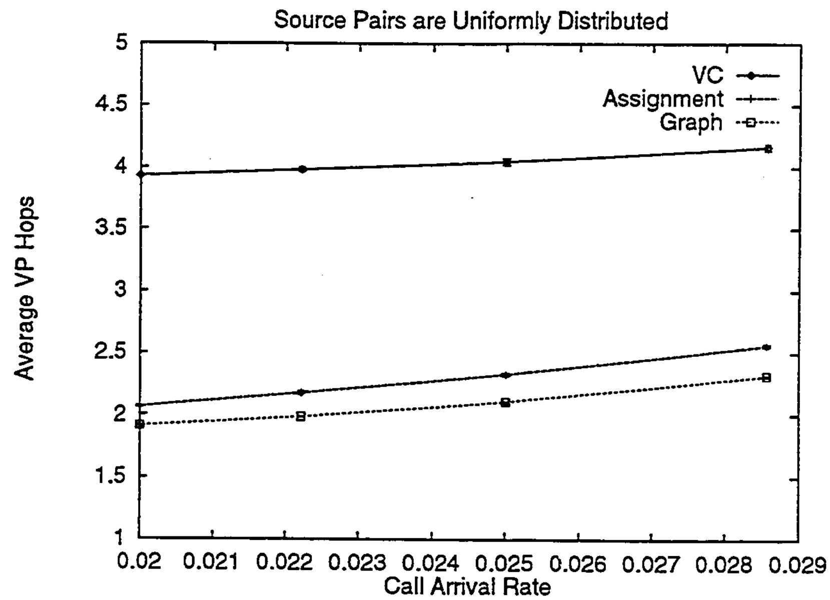
Figure 5.2: Loss Rate Analysis for A-1 (50 node) network: Sources Pairs are Uniformly Distributed

the most efficient bandwidth utilization and lowest call blocking (see Figure 5.1). In a VP topology, each physical link is shared by one or more VPs which results in bandwidth fragmentation and correspondingly higher levels of call blocking than the VC routing. Also due to dynamic bandwidth adjustments of VPs the blocking rate will be increased.

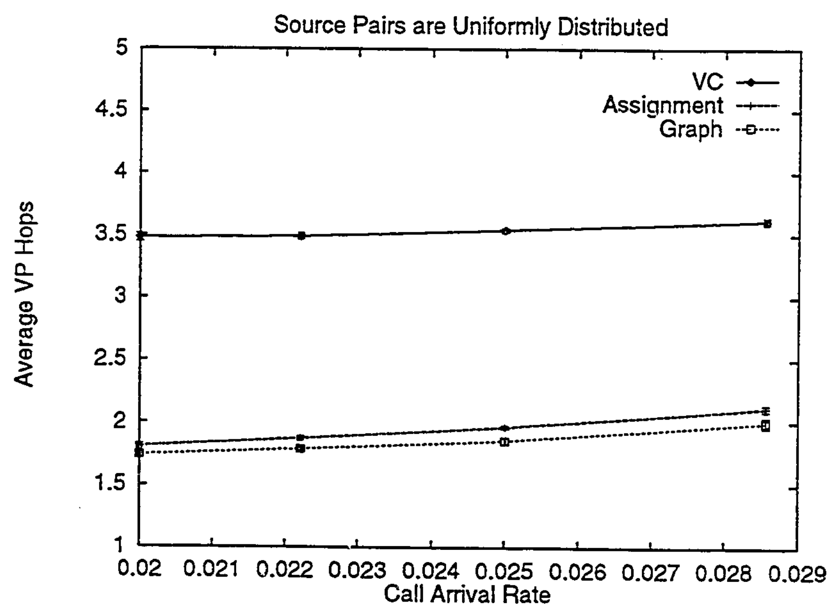
In Figure 5.1, for the US network it can be seen that the overlay reacts to different distribution of sources with different loss rates for the same load. However, we have claimed that the performance of *assignment* and *graph* will not be strongly affected by a change in the traffic pattern. When the load stays the same there are two reasons for this behavior. First we have only sampled a small number of source pairs in the network; the total source pairs for US network is 756. Second, now in the performance analysis the capacity allocation scheme is also included. Thus depending upon the availability of spare capacity on the physical links through which the calls are routed, the loss rate can fluctuate. However, if the capacity on each physical link is large, the above scenario can be avoided. In the next section some more results are given to support these conclusions.

The performance of *assignment* and *graph* is nearly identical in Figure 5.1. However, for the A-1 network the performance of *assignment* may be slightly better than *graph* (see Figure 5.2) although with overlapping confidence intervals this is not certain. The differences in performance can be explained by looking at the average link load graph in the previous chapter. There we can see that the average link load of *assignment* and *graph* for US (28 node) network are perfectly matched; whereas there is a very slight difference between *graph* and *assignment* for A-1 (50 node) network. This difference could result in lower bandwidth fragmentation for *assignment* and therefore lower loss rate.

The real advantage of a VP overlay is its ability to reduce management cost (by bundling VCCs) and to reduce connection setup time. The latter advantage can be seen in Figure 5.3, where by using a VP topology, the delay involved in setting up



(a)



(b)

Figure 5.3: Connection Setup Delay Analysis for US (28 node) network: (a) - (b) Sources Pairs are Uniformly Distributed

a call is significantly lower than for VC routing. The *graph* algorithm has somewhat lower average hop count but the relative performance of *graph* and *assignment* are also similar for this measure.

5.2.2 Effect of Changes in Traffic

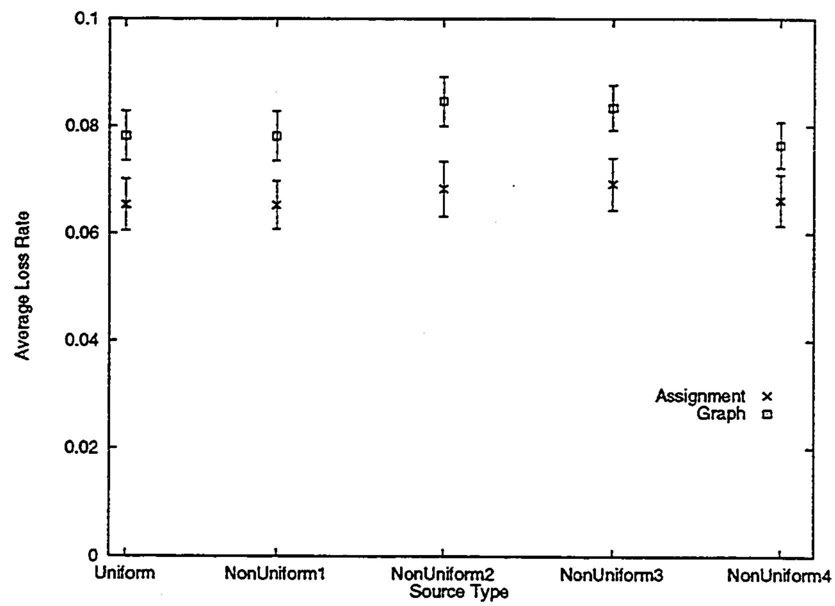
In the previous section we have identified two reasons for the change in performance for *assignment* and *graph*, when the choice of sources is changed. In this section, we will test a smaller network EN-1 (10 nodes) in order to sample all the possible source pairs (in this case 90).

Since all the source pairs are considered in this experiment, we bring some non-uniformity to the traffic by changing the *holding* time of calls. A random number is deducted from the call holding time between all the source pairs with the same source node. The same number is then added to the holding time of another set of source pairs having the same source node. In this way the load remains the same as with the uniform traffic. The process is repeated N times where N is the number of nodes in the network, in order to cover all the sources in the network.

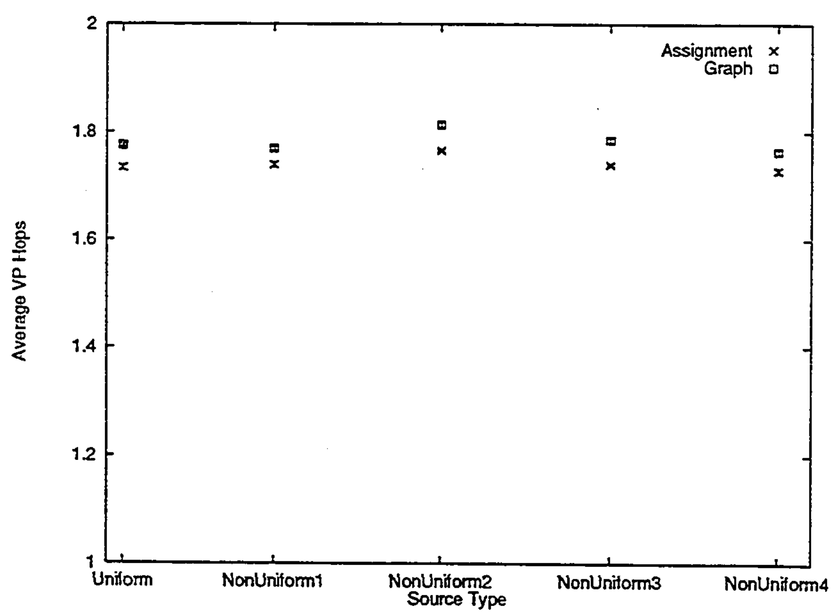
First the experiment is run multiple times for *assignment* and *graph* under uniform traffic and then for four different non-uniform traffic examples with the same load as uniform traffic. The average loss rate and 95% confidence intervals are then plotted in Figure 5.4. In Figure 5.4 we can see that the mean loss rate for uniform traffic and non-uniform traffic remains almost the same. For the same algorithm the variation in loss rate can be explained by the affects of the capacity allocation algorithm pointed out in the previous section. This experiment, therefore, confirms that both *assignment* and *graph* generate a VP overlay in which changes in traffic pattern will not adversely affect the performance of the overlay provided the load remains the same.

5.2.3 Effect of Number of VPs

In this section, we use simulation results to highlight the flexibility of *assignment*.

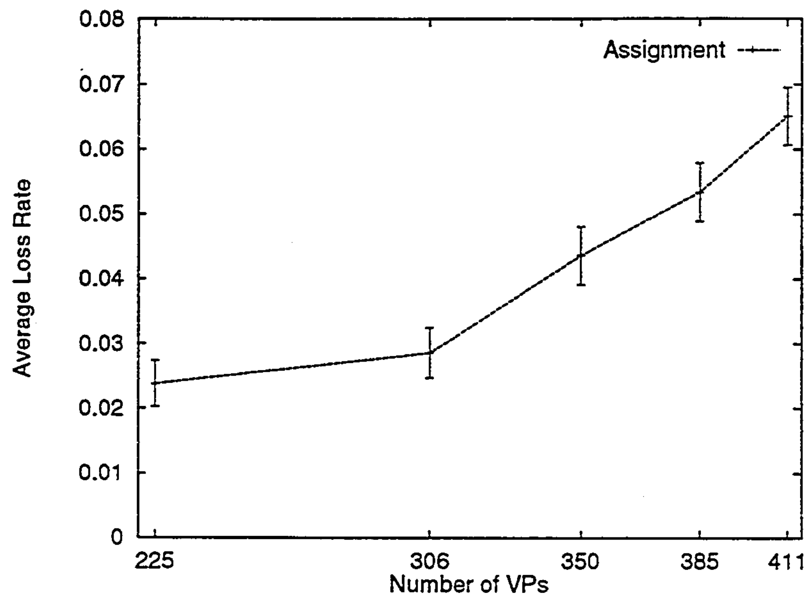


(a)

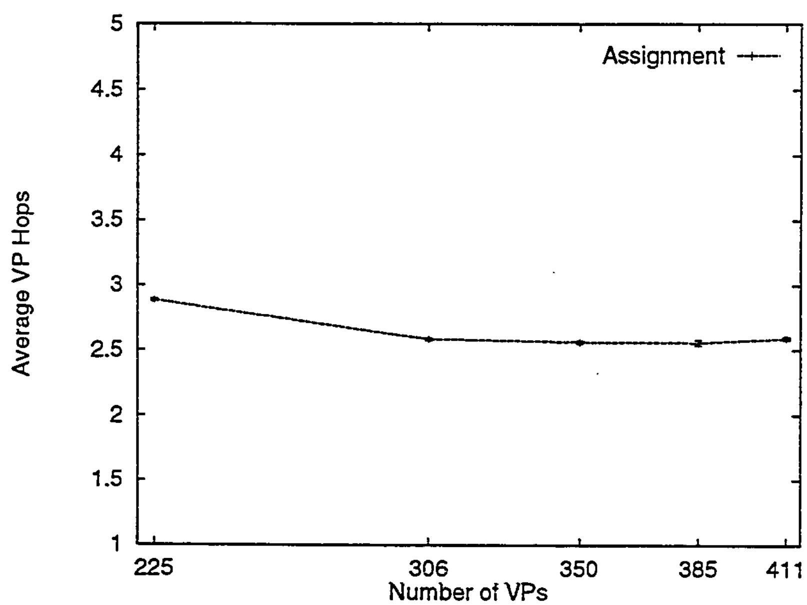


(b)

Figure 5.4: Changing the Holding Time in Non-uniform Traffic



(a)



(b)

Figure 5.5: Changes in the Number of VPs for Assignment: (a) Loss Rate (b) Average VP Hops

In the previous chapter it is pointed out that *assignment* gives better control in the selection of the number of VPs. In the experiment we have varied the number of VPs for US (28 node)and plotted the loss rate and the average VP hops. The capacity available on each physical link for the overlay is fixed at 57Mbps in all the cases. Again for each configuration multiple runs are made and the average value and 95% confidence intervals are found. The number of VPs is determined by using the procedure developed in the previous chapter for average VP hop value of 2.0, 1.7, 1.6, 1.55, 1.5 which correspond to 225, 306, 350, 385 and 411 VPs respectively. The number of VPs (350) for 1.6 average VP hops is similar to the number of VPs established using *graph* for a maximum VP hop count of 2.

In Figure 5.5 it can be seen that when the number of VPs are lowered from 350, the loss rate improves. However, the average VP hop is also increased. The reduction in the loss rate is due to the lower bandwidth fragmentation by using a smaller number of VPs. The increase in the number of VPs reduces the average VP hops but increases the loss rate due to the greater number of VPs sharing a link since 57Mbps is too small for an overlay with a large number of VPs. Thus for a given capacity available on the physical links for the overlay, there is a tradeoff between the average VP hops and the loss rate. However, *assignment* provides better control in adjusting to a particular tradeoff than *graph*. In the figure it can be seen that by establishing (306) VPs the *assignment* can provide lower loss rate without any degradation in performance to the average VP hops.

5.2.4 Effect of Changes in Properties of the Overlay

The properties identified in this thesis are important for an overlay to be effective. We have highlighted the importance of individual properties in Chapter 4 along with the effect of satisfying and not satisfying the property on performance measures. However, the use of property 5 (availability of alternative paths) in the overlay can be demonstrated more effectively by using the simulation results. Here, an experiment is

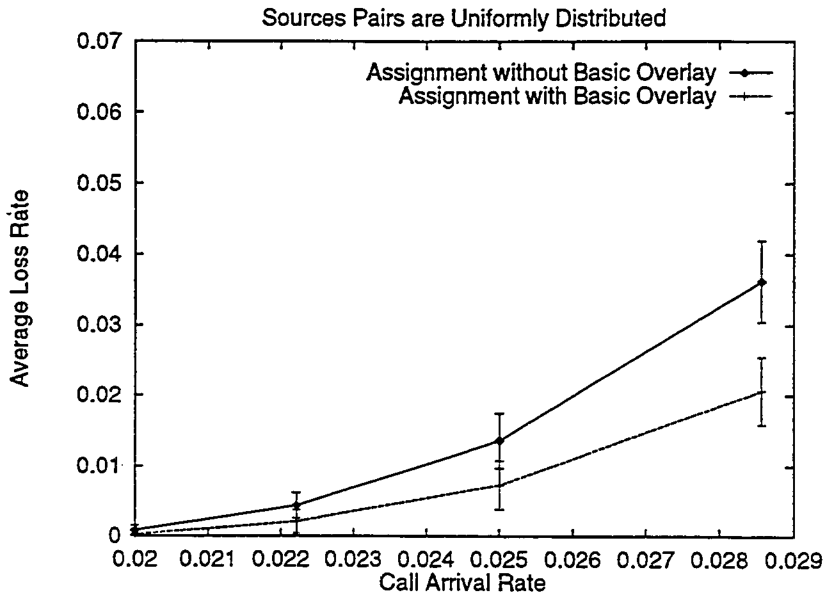


Figure 5.6: Assignment with and without BasicOverlay

conducted to test the effect of removing property 5 (availability of alternative routes) from the *assignment* method. This can be done by not generating the *BasicOverlay* and using $M = TotalVPs$ (see Section 3.3.1). In the simulation, the connection routing algorithm looks for alternative routes when the VPs on its shortest hop path are blocked. If the source pairs are not well-connected, higher loss rates would be expected. In *assignment* the use of the *BasicOverlay* provides better connectivity and, therefore, alternative paths between the source and destination. This is shown in Figure 5.6, where *assignment* with *BasicOverlay* has significantly lower loss rate.

5.3 Conclusion

In this chapter the performance of the VP overlay is analyzed using a call-level simulator of an ATM network. The dynamic bandwidth allocation scheme used is designed independently of the VP overlay and, therefore, may not be an ideal scheme for a particular overlay. The capacity allocation scheme is crucial for improving the loss rate of VP overlay and should be designed taking into account the design philosophy of VP overlay. However, even using a bandwidth allocation scheme developed independently of the VP overlay, the average loss rate is within a reasonable range relative to VC only routing. An effective VP overlay should be capable of reducing the connection setup delay without causing unacceptable call blocking due to bandwidth fragmentation. The *assignment* algorithm can be used to provide such an overlay and it also gives the flexibility of establishing a specific number of VPs to fine tune a specific loss rate/connection setup requirement. Finally, we have demonstrated that the properties presented in this thesis are important and by not satisfying them the resulting overlay can give poor performance.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

In ATM networks, a VP overlay can play an important role in reducing management cost and connection setup cost. However, the VP overlay must be developed with certain properties such that the the network can get the maximum benefit from it. We have identified the following properties:

1) Connection Setup Cost.

2) Physical Link Utilization.

3) Survivability.

4) Resource Utilization.

5) Availability of Alternative Paths.

6) Minimal Reconfiguration.

7) Flexibility

Each of the above properties is important in a VP overlay. We have demonstrated the importance and tradeoffs of these properties. Also a method has been developed to incorporate these properties in the generated overlay. In our approach, we have proposed the use of *assignment* technique in solving the VP assignment problem.

The cost function in our approach satisfies properties 1-4. Property 5 is included by establishing the *BasicOverlay*. Property 6 is satisfied due to the special structure of our algorithm i.e. the algorithm distributes VPs regularly throughout the network and property 7 is satisfied due to the step-wise nature of the algorithm and the ability to assign independent costs between selected node pairs. Thus by including property 7, *assignment* can adapt to switching and routing constraints quite efficiently. Also since it proceeds in a step-wise manner at any stage the total number of VPs generated would follow the laid down properties. This will also make this approach more applicable to real conditions in which physical topology changes should not require the regeneration of the whole VP topology as would be required in the case of some of the other approaches. The algorithm proposed is traffic independent and, therefore, it can provide effective service for a wide range of traffic patterns. Thus the VP overlay can remain effective for a long period of time without requiring any change.

The inclusion of the above properties produces a VP topology which gives consistent results under varied network topologies, in terms of connection setup cost, average connection cost and survivability. We have compared the performance of *assignment* with other algorithms and in all cases *assignment* provides small maximum and average link load with reasonable average connection cost. The *graph* algorithm gives comparable performance but it does not provide the same flexibility as *assignment*. The performance of *assignment* is unaffected due to the changes in routing(VP routing or VP+VC routing) or changes in traffic. This makes *assignment* suitable for ATM where the fluctuation in performance is not desirable.

From the analysis of the simulation results we have seen that although VP overlays may result in higher loss rate than VC only routing, the potential benefits of management and reduced set up time of using a VP overlay outweighs the higher loss rate. With bandwidth becoming cheaper, the loss rate can be reduced by increasing the link capacity. Thus through the simulation analysis the *assignment* can be effective in a real world scenario as well.

6.2 Future Extensions to the Model

During the development of our approach various extensions to the model have been identified. We now explain these extensions and the benefits to be gained through them.

6.2.1 Capacity Allocation

In the simulation results we have pointed out the importance of a bandwidth allocation scheme for the VPs. The scheme should be designed by taking into account the characteristics of a particular overlay which has been used. The dynamic bandwidth allocation scheme we have used can be improved and, based on the knowledge of a particular overlay, its behavior can be altered. In our bandwidth allocation scheme, more work is needed to identify the optimal allocation of initial bandwidth on VPs and thereby reducing the fragmentation of resources on each physical link. A good bandwidth allocation scheme can greatly improve the loss rate of VP overlays making them feasible for operation in real networks.

6.2.2 Dynamic Topology Allocation

The *assignment* algorithm has the flexibility of generating VPs in a step-wise fashion which allows it to generate a given number of VPs such that the generated VPs and all the previously established VPs still obey the laid down properties. This flexibility of *assignment* can also be used to generate a short term VP topology to facilitate multipoint connection. The generated VP overlay will have a dynamic nature and will be removed when the multipoint connection is terminated. However, the effectiveness of such an overlay generated by *assignment* needs further investigation.

6.2.3 Routing Extensions

In our approach the routing step is isolated from the VP generation step. However, the routing can be performed as the VP terminators are generated, this way it can provide feedback on the congestion of links. A link is congested if the number of VPs assigned to it has reached a predetermined threshold level. The threshold value can be dependent upon the available capacity of the link or some other parameter determined by the network operator. After each iteration of Algorithm I, the routing is done for each VP starting from the first row. At any VP routing, if the routing procedure indicates the congestion of a link, the *assignment* would generate a least loaded shortest path spanning tree for each node by using $LL = SP = 1$(see the routing algorithm in Chapter 3). If the congested link belongs to a spanning tree of a node then a high cost value would be assigned between the root node of the tree and all the nodes in the subtree on which the congested link terminates. If there is a VP already assigned between the root node and any node of the subtree (in the subsequent rows of matrix), then that VP would be ignored.

The advantage of this approach is that each link can be assigned a threshold value based on some important parameter and *assignment* would automatically change its behavior to satisfy the link load constraints. Thus a more realistic VP overlay can be generated by extending the routing functionality. However, the extension would require sufficient processing power and therefore the need to develop a parallel algorithm for multiprocessor system would be suitable rather than a sequential implementation.

6.2.4 Very Large Networks Encompassing Different Geographic Domains

For very large networks, it is possible that the network comes under different domains who want to control the switches in their own domain. Our model has the potential to be extended to operate in a distributed manner, in which each domain would independently generate VPs according to their own requirement.

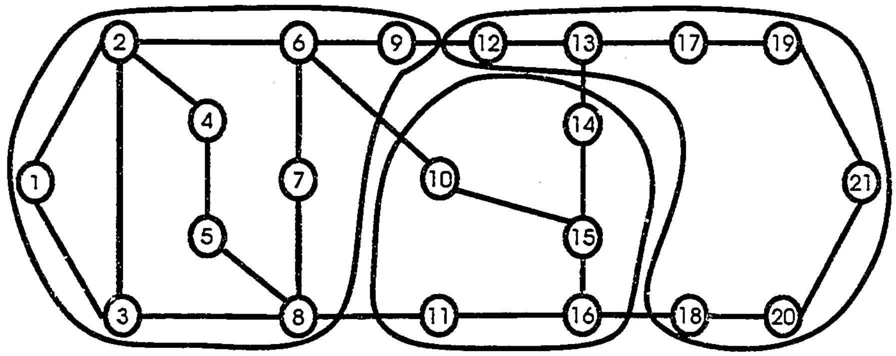


Figure 6.1: ARPA2 Network after Clustering

For example in Figure 6.1, the APRA2[1] network is divided into three clusters. Each cluster is independent of the other and for each cluster *assignment* can be used to generate VPs according to the desired performance (determined by using the procedure in Chapter 4). The total number of VPs for a desired performance for the entire network is also determined by using the procedure presented in Chapter 4. The number of VPs generated for each cluster would be deducted from the total number of VPs available for the entire network. After each cluster has generated its set of VPs and the sum of VPs generated is deducted from the total available VPs. The *assignment* is again used to assign the remaining VPs among nodes which do not belong to the same cluster.

Both the intra-cluster and inter-cluster assignment of VPs are independent of each other and can be established in parallel. The overall affect of establishing VPs in distributed manner is expected to be similar to the VPs that are generated using the whole network. The advantage gained by this approach is that the task of maintaining and operating a very large network is distributed into manageable domains. Also each cluster has full control of establishing VPs to achieve the desired performance.

[1]This network is only used here to demonstrate the idea and is not considered as a very large network

It would also speed up the process of generating VPs and assigning their physical routes.

6.2.5 Polynomial Time Execution

In the complexity analysis of algorithm I (see Chapter 3), the value of i should remain within reasonable bounds in order to have polynomial run time. In most practical situation this value is expected to remain within reasonable limits. However, more work is needed to determine the behavior of i for a particular problem.

Appendix A

Diagrams of Real Network Topologies

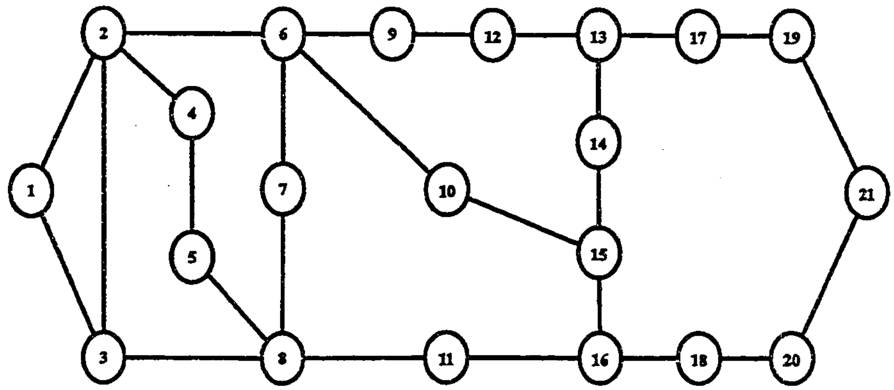


Figure A.1: ARPA2 Network of 21 Nodes

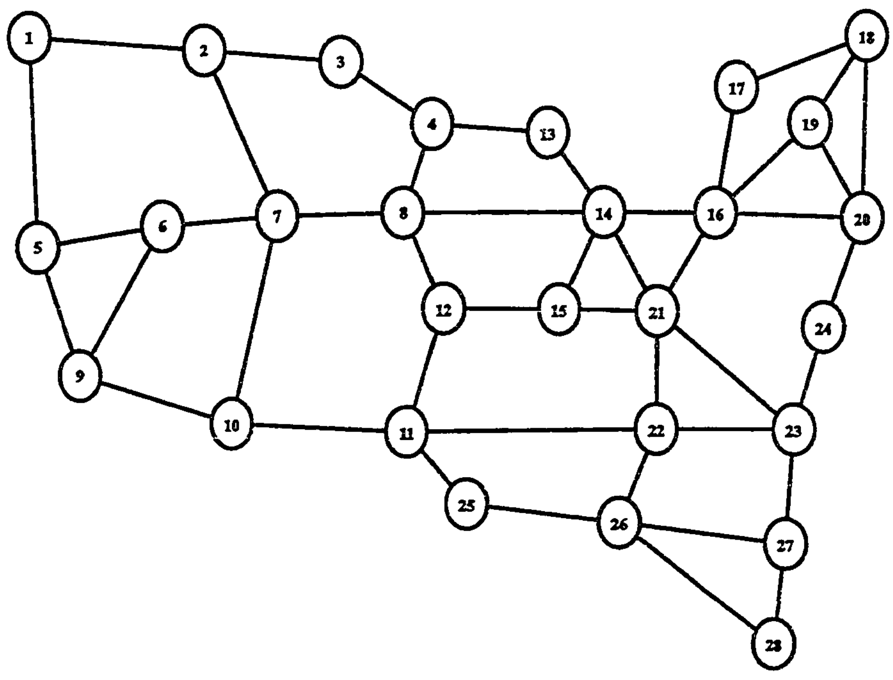


Figure A.2: US Network of 28 Nodes

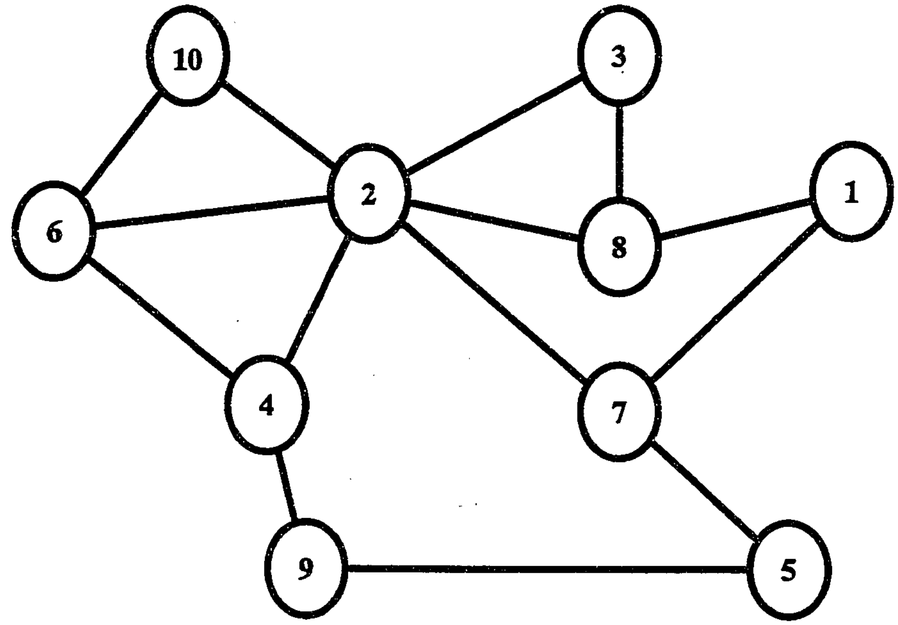


Figure A.3: EN-1 Network of 10 Nodes

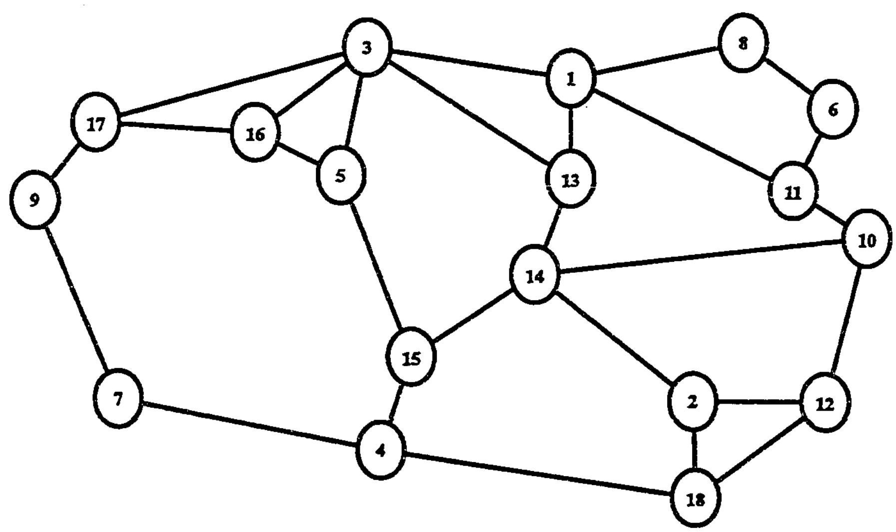


Figure A.4: EN-2 Network of 18 Nodes

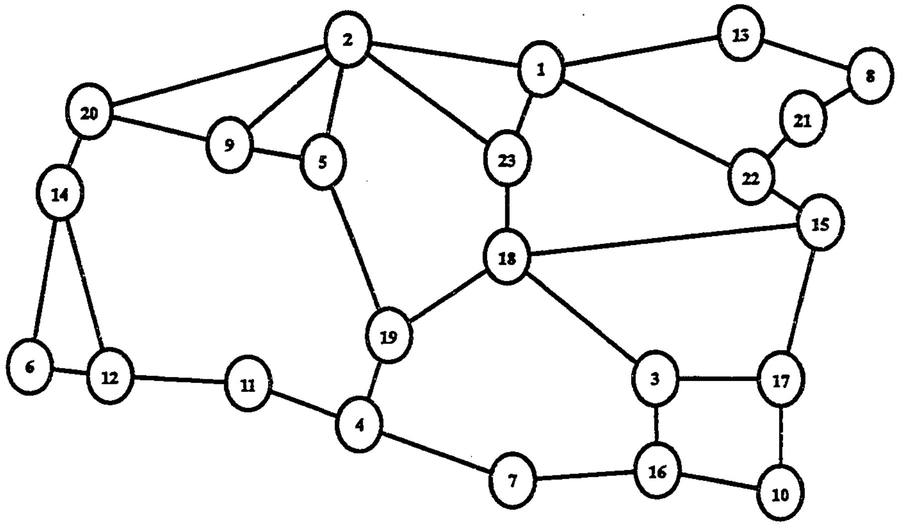


Figure A.5: EN-3 Network of 23 Nodes

Bibliography

[1] S. Ahn, R. Tsang, S Tong, and D. Du. Virtual path layout design on ATM networks. *INFOCOM'94*, pages 192–200, 1994.

[2] A. Alles. ATM internetworking. *Cisco Systems, Inc. report*, 1995.

[3] K.-I. Baba, M. Murata, and H. Miyahara. Performance analysis for the dynamic virtual path bandwidth control method in ATM networks. *International Journal of Communication Systems*, 7(4):283–294, 1994.

[4] G. Bell. Ultracomputers: A teraflop before its time. *Communications of the ACM*, 35(8):26–47, 1992.

[5] J. Burgin and D. Dorman. Broadband ISDN resource management: The role of virtual paths. *IEEE communications Magazine*, 29(9), 1991.

[6] R. Byung, O. Hiroyunki, M. Masayunki, and M. Hideo. Design algorithm for virtual path based ATM networks. *IEICE Transactions on Communications*, E79-B(2):97–107, February 1996.

[7] I. Chlamtac, A. Faragó, and T. Zhang. How to establish and utilize virtual paths in ATM networks. *Proceedings of IEEE ICC93*, pages 1368–1372, 1993.

[8] O. Eiji and Y. Naoaki. An optimum logical-design scheme for flexible multi-qos ATM networks guaranteeing reliability. *IEICE Transactions on Communications*, E78-B(7):1016–1024, July 1995.

[9] S. Fotedar, M. Gerla, P. Crocetti, and L. Fratta. ATM virtual private networks. *Communications of the ACM*, 38(2):102–109, 1995.

[10] V. Friesen, J. Harms, and J. Wong. Resource management with virtual paths in ATM networks. *IEEE Network*, 10(5):10–20, September/October 1996.

[11] O. Gerstel, I. Cidon, and S. Zaks. Efficient support for the client/server paradigm over heterogeneous ATM networks. *INFOCOM'96*, pages 10c.3.1–10c.3.8, March 1996.

[12] O. Gerstel, I. Cidon, and S. Zaks. The layout of virtual paths in ATM networks. *IEEE/ACM Transactions on Networking*, 4(6):873–884, December 1996.

[13] R. Jain. ABR service on ATM networks: What is it? *Network World, June 24*, 1995.

[14] J. Jiau. Study of integrated bandwidth allocation and multiplexing strategy for VP-based ATM networks. *Proceedings of the 14th Annual Int'l Phoenix Conf. on Computers and Communications*, pages 226–232, 1995.

[15] B. Kolman and R Beck. *Elementary Linear Programming with Applications.* Academic Press, 1980.

[16] H. W. Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2:83–97, 1955.

[17] D. Medhi. Multi-hour, multi-traffic class network design for VP-based wide-area dynamically reconfigurable ATM networks. *INFOCOM'95*, pages 7c.4.1–7c.4.8, 1995.

[18] K. Murakami and H. Kim. Virtual path routing for survivable ATM networks. *IEEE/ACM Transactions on Networking*, 4(1):22–39, February 1996.

[19] S. Parekh and K. Sohraby. Some performance trade-offs associated with ATM fixed-length vs. variable-length cell formats. *Proceedings, GLOBECOM*, 3:39.4.1–39.4.6, November 1988.

[20] C. Partridge. *Gigabit Networking.* Addison Wesley, 1994.

[21] K. Sato, S. Ohta, and I. Torizawa. Broad-band ATM network architecture based on virtual paths. *IEEE Transactions on Communications*, 38(8):1212–1222, 1990.

[22] R. Sedgewick. *Algorithms.* Addison-Wesley Publishing Co., 1984.

[23] W. Stallings. *Data and Computer Communications.* Macmillan Publishing Co., Inc, 1991.

[24] M. Steenstrup. *Routing in Communications Networks.* Prentice Hall, 1995.

[25] H. Taha. *Operations Research: An Introduction.* Macmillan, 1989.

[26] J. Wong and R. Pankaj. A diameter based method for virtual path layout in ATM networks. *Proc. Int'l. IFIP-IEEE Conf. on Broadband Commun., L. Mason and A. Casaca, eds., Canada*, 1996.