ROUTING AND SWITCHING IN TELECONFERENCING NETWORKS

LEUNG, Yiu-Wing

A DISSERTATION SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

IN THE DIVISION OF INFORMATION ENGINEERING

GRADUATE SCHOOL

THE CHINESE UNIVERSITY OF HONG KONG

HONG KONG

1992

Acknowledgments

I would like to express my deepest gratitude and appreciation to my supervisor, Dr. T.S.Yum, for his continuous guidance throughput my years of study. His valuable comments and suggestions have a fundamental influence on the development of this thesis, and his continuous support and encouragement sustain my graduate study.

I would also like to thank Dr. Soung C. Liew of Bellcore. He has read an early version of Chapter 4 and has made a number of suggestions for improving the technical content.

Thanks are also due to the Department of Information Engineering which provides me with a comfortable working place and a wealth of computing facilities.

Abstract

The routing and switching of packets are fundamental functions of a packet switched network. For a packet switched network to support multipoint videoconferencing, the routing and switching functions are complicated by the fact that a source station needs to send packets to multiple destination stations and that multiple source stations need to send packets to the same destination station. We first review the potential broadband video services and discuss why videoconference is a viable service that will get more and more popular in the near future. We then proposed algorithms for a generic routing problem called multiple destinations routing (MDR). In this problem, a minimum cost connection path is to be found to connect a given set of nodes (called conference nodes) of a network. This problem has been shown to be NP-complete. We proposed a method to reduce the number of enumerations required for optimal solutions. Numerical results show that, when the number of conference nodes is small (say ≤ 5), this method can find the optimal solutions using reasonable computation time. This method is therefore suitable for multipoint videoconference application because the number of involved conference nodes in each conference is usually not very large. Three heuristics are designed for large MDR problem (i.e. the number of conference nodes is large). With the use of the property that a good path to connect the source node and a destination node should not be too much longer than the shortest path connecting them, we designed heuristic $A^{(k)}$. The parameter k allows us to trade-off between optimality and computation time. Heuristic B is modified from the Prim's algorithm for finding minimum spanning tree, taking advantage of the property that two or more paths may share common edges. Heuristic C is for networks with uniform edges weights and it overcomes the deficiency of heuristic B. We then proceed to formulate and solve the routing problems of two types of videoconferences called selectable media conferences and common media conferences. Algorithms that use a combination of table look-up and on-line processing are designed for computing the optimal paths to connect a given set of conference

sites. The blocking probabilities of these two types of conferences in fully connected networks are derived and compared. The sensitivity of network throughput to conference size distribution is also studied. It is found that, for a given mean conference size, the variance of the conference size distribution has a small but non-negligible effect on the network throughput. We then turn our attention to the switching issues and design a TDM-based multibus packet switch for high speed switching. This switch has the advantages of (1) very simple control circuit, (2) 100% potential throughput under heavy traffic, (3) internally unbuffered and (4) adding input and output ports without increasing the bus and I/O adaptor speed. A combined analytical and simulation method is used to obtain the mean packet delay and packet loss probability. Numerical results show that for satisfactory performance the buses need to run about 30% faster than the input line rate. Based on this multibus design, we design a shared media switch architecture for distributing broadcast and switched videos using a set of concentration buses, a TDM-based bus matrix and a set of distribution buses. Dedicated time slots in a frame are reserved for the broadcast videos. The remaining time slots are allocated to the switched videos on a first-come-first-serve basis. Videos are switched via time slot assignments which determine the connections within the bus matrix. Two slot assignment algorithms are designed, one for point-to-point transmissions and the other for point-to-multipoint transmissions. This switch architecture has three advantages. First, multirate video channels can be accommodated. This can accommodate a variety of video services that have different bit rate requirements. Second, videos can easily be broadcast or multicast to the customers through the shared buses. Hence, multipoint communication services (e.g. videoconferencing) can be provided. Third, it can be used as a building block for constructing large video distribution networks.

Table of Content

Chapter 1
Introduction

1.1 Background

While the vast majority of today's telephone and entertainment video services are still provided to subscribers over metallic transmission media and/or via radio waves, the use of optical fiber transmission systems to deliver these and other services has been the focal point of considerable research and development worldwide. Using single mode fibers, data can now be transmitted at a few gigabits per second over un-repeated distances of a few hundred kilometers at a bit error rate of 10^{-9} [ACAM 90, MUKH 91]. This technological advance will change the communication infrastructure in the way that optical fiber will be used as the transmission medium. On the other hand, semiconductor technology can now offer high-speed, high-complexity and low cost integrated circuits for signal processing, transmission and switching. With these technologies, the growing demand for broadband video services for various business and residential applications will be economically met in the near future.

1.2 Broadband Video Services

A number of broadband video services have been proposed for business as well as residential applications. CCITT Recommendation I.211 [CCIT 90] classifies these services into *distribution services* and *interactive services* (Fig.1).

A. Distribution Services

For distribution services, information is primarily transferred from the service provider to the customers. Customers either have no control or have limited control over the presentation of the information.

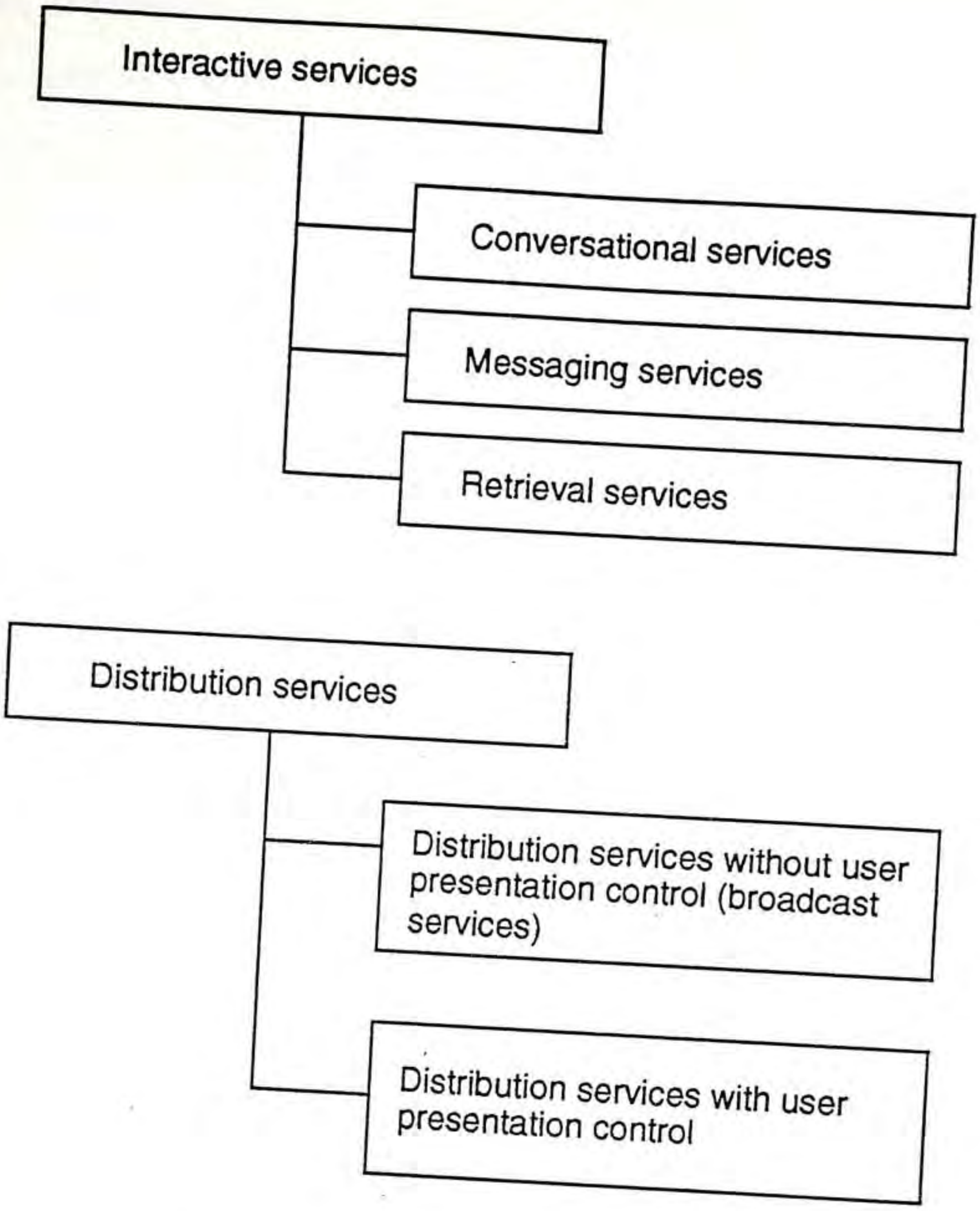


Fig.1: Broadband services

Some distribution services do not allow the users to control the order of information presentation and these services are also known as broadcast services. These services distribute a continuous flow of information from a central source to a virtually unlimited number of authorized receivers connected to the network. Each user taps into and accesses this flow of information but he cannot control the starting time or the order of presentation of the broadcast information. The most common example of this type of services is broadcast TV. Currently, broadcast TV is provided to the subscribers via radio waves and/or cable TV distribution systems. With the availability of broadband networks, a larger number of TV channels of higher resolution can be provided.

Some distribution services allow the users to control the order of information presentation. For these services, information is distributed from a central source to a large number of users in the form of a sequence of information entities (called *pages*) with cyclical repetition. A user can access to and retrieve any pages and hence he can control the order of information presentation. Although the service appears to be interactive to the user, it is actually a one-way broadcast of information. For example, the Teletext service [STAL 92] can be enhanced with the availability of a broadband network to cyclically transmit pages of text, images, audio and video.

B. Interactive Services

Interactive services provide a two-way exchange of information between two or more users or between a user and a service provider. They can be divided into three classes: conversational services, messaging services and retrieval services.

Conversational services provide dialogue communication and real-time information transfer between two or more users. This service class encompasses a wide range of applications, and probably the most important applications are videophone and videoconferencing.

Messaging services provide asynchronous communication between users using message handling functions and storage units. In contrast to conversational services, messaging services are not in real-time. Hence, messaging services have a less stringent transmission delay requirement and the users need not be available at the same time. An example of this type of services is multimedia mail, which is analogous to today's electronic mail. Multimedia mail can carry video, images, voice as well as text to its destinations.

With retrieval services, users can selectively retrieve information stored in the information centers. Information is retrieved on an individual basis and the transfer start time is under user control. Broadband communication networks make large volume of information of various types (e.g. high resolution images, video) within reach that would otherwise be difficult to obtain.

C. Summary of CCITT Recommendation I.211

Table 1 summarizes CCITT Recommendation I.211 [CCIT 90], listing the broadband video services and their potential applications.

Categories	Classes	Types of information	Service examples	Applications
Distribution services	without user presentation control	video and sound	- TV distribution - pay TV	- TV programme distribution
	with user presentation control	text, graphics, sound, image and video	- full channel broadcast videography	- tele-advertising - news distribution
Interactive services	Conversation al	video and sound	- videophone - videoconference	- remote video lecture, - business meetings - teleshopping
	Messaging	video and sound	- multimedia mail	- electronic multimedia mailbox service for the transfer of moving pictures and sound
	Retrieval	text, data, graphics, sound, still images and video	- videotex - video on demand - high resolution image retrieval	- tele-shopping - tele-advertising - news retrieval - telesoftware - remote education - multimedia document retrieval - medical image communication

Table 1: Broadband video services and their applications

D. Service Bit Rates

Video requires large transmission bandwidth and determines, in more or less extent, the overall capacity requirement for a broadband network. Analog video signal requires 6 MHz bandwidth. If the video signal is sampled at 12 MHz and digitized using 8 bits per sample, the resulting data rate is 96 Mb/s. To reduce the data rate, video can be compressed before transmission to remove the redundant and less significant information while the resulting distortion is acceptable. What is acceptable in terms of image quality and data rate depends on the applications. For example, the acceptable quality for entertainment video is usually higher than that of interactive videos (e.g. videophone or videoconference) and so entertainment video can be transmitted at a higher data rate. CCITT has defined five levels of video quality (Table 2) [STAL 92]. Class A video has the best quality but requires the highest data rate, and the converse is true for class E video.

Quality Class	Descriptions
A	High-definition TV
B	Digital component-coding signal
C	Digitally coded NTSC, PAL, SECAM
D	Reduced spatial resolution and movement portrayal
E	Highly reduced spatial resolution and movement portrayal

Table 2: Five levels of video quality defined by CCITT

The required data rate for each service class is decreasing with the advance of video compression techniques. Table 3 summarizes the recently reported video quality and the corresponding data rate requirement [BODS 89, FISC 90, TOBA 91].

Quality	Data Rates
HDTV; uncompressed	397 Mb/s
HDTV; compressed; no degradation	44 Mb/s
NTSC; uncompressed	96 Mb/s
NTSC; compressed; VCR quality	1.5 Mb/s
adequate for facial expression communication; noticeable degradation when there are large motion; for videoconference	112 kb/s
noticeable inferior images when there are large motion; for videophone	64 kb/s

Table 3: Video quality versus data rates requirement

1.3 Videoconference Services

The growing desires for improving office productivity result in an increasing demand for advanced information services from the business community. Since the business community can afford and is willing to pay higher service charges, it is an important source of revenue for the initial economic justification of broadband network. Therefore, it is expected that a significant portion of broadband video services will initially be provided for business applications [ARMB 87]. Among the many potential broadband video services for business applications, videoconference service has been regarded as the most "futuristics" [ARMB 86, FISC 90]. This service allows businessmen to conduct meeting without leaving their offices. Sitting in front of their workstations, users can see each other via real-time video, talk and listen via real-time audio, and watch presentation via on-line electronic blackboard. With this service, businessmen can save travel time and expenses, and they can convene more frequently with greater flexibility in order to speed up and improve the decision-making process.

1.4 Videoconference Systems

An ideal videoconference system should incorporate all the features of face-to-face meeting:

(a) Should allow any number of conferees at any number of sites;

(b) Should maintain continuous visual live presence as well as spatial relationship of all conferees located in the remote conference sites and provide "eye-contact" information (i.e. information about who is looking at whom);

(c) Should allow one or more conferees to speak simultaneously and allow private audio communication among a subgroup of conferees;

(d) Should allow the sharing of all visual media (e.g. documents) and allow the interactive updating and annotation of all visual media;

(e) Should allow a conferee to leave a conference and allow a new comer to join an ongoing conference.

It may not be economically justifiable to incorporate all the above ideal features in a videoconference system. Since the effectiveness of videoconference service depends on how much it can emulate a face-to-face meeting, a good videoconference system should incorporate only the most important features while the cost is reasonable. This results in many design options, depending on what features of face-to-face meeting are included. In the following subsections, we describe six videoconference systems that were reported in the literature. Since video requires large transmission bandwidth and must be transmitted in real-time, it has the most stringent transmission and switching requirements. Our descriptions below will emphasize the video communication aspects in the six videoconference systems.

A. Single Camera System [SABR 85]

Single camera system is designed to connect two conference studios. Fig.2 shows the configuration of a single camera system. A camera captures the images of all conferees and the resulting video signal is transmitted to the other conference studio. This system can provide continuous visual presence of all conferees located in the other conference studio as well as their spatial relationship. Since the present camera and TV systems use 4:3 aspect ratio, the resolution per conferee decreases as the number of conferees in a conference studio increases. As a result, the number of conferees in each conference site should not be large. By using a V-shaped table, the aspect ratio can be better utilized.

B. Voice Switched System [WRIG 83, CANA 84]

Voice switched system is designed to connect two conference studios. Fig.3 shows a typical layout for a voice-switched videoconference system. Several cameras are used to capture the images of the conferees. However, only the output of the camera covering the current speaker is transmitted to the other conference studio. In other words, the audio signal is used to switch the output signal of one of these cameras to the transmitter. As a result, the conferees can only see the current speaker and his neighbour(s). Although the resolution per conferee is higher than that of the single camera system, the spatial relationship and continuous presence of the conferees are not conveyed in this system.

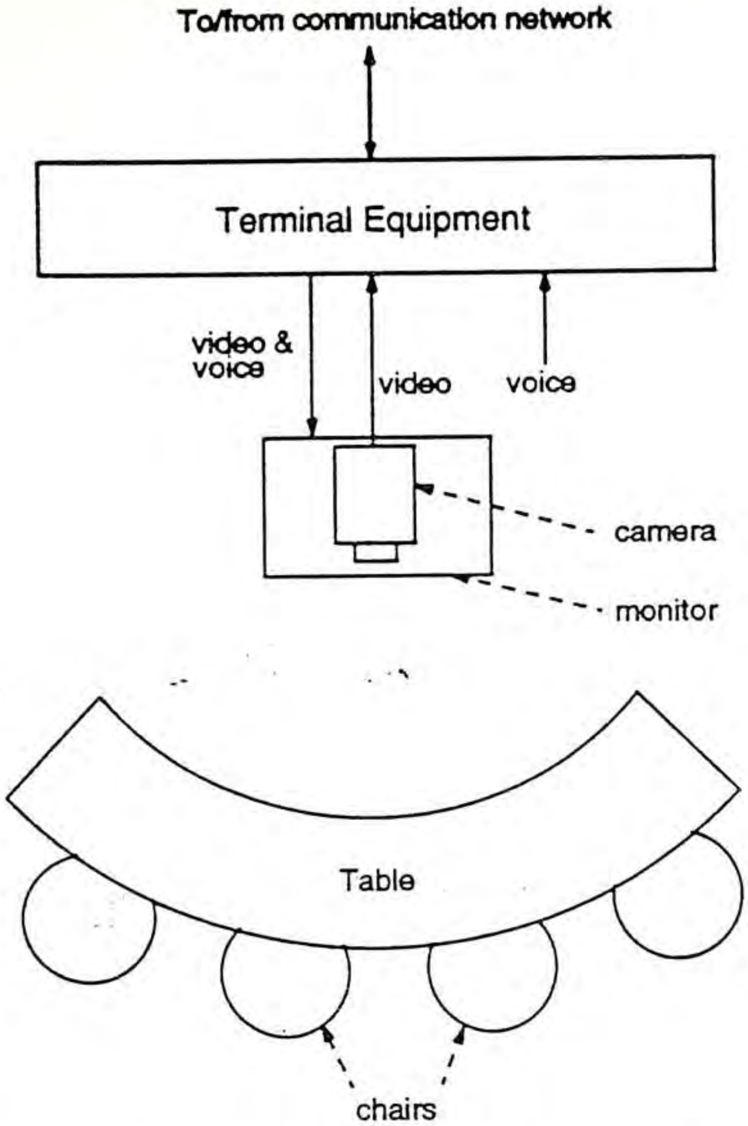


Fig.2: Single camera videoconference system

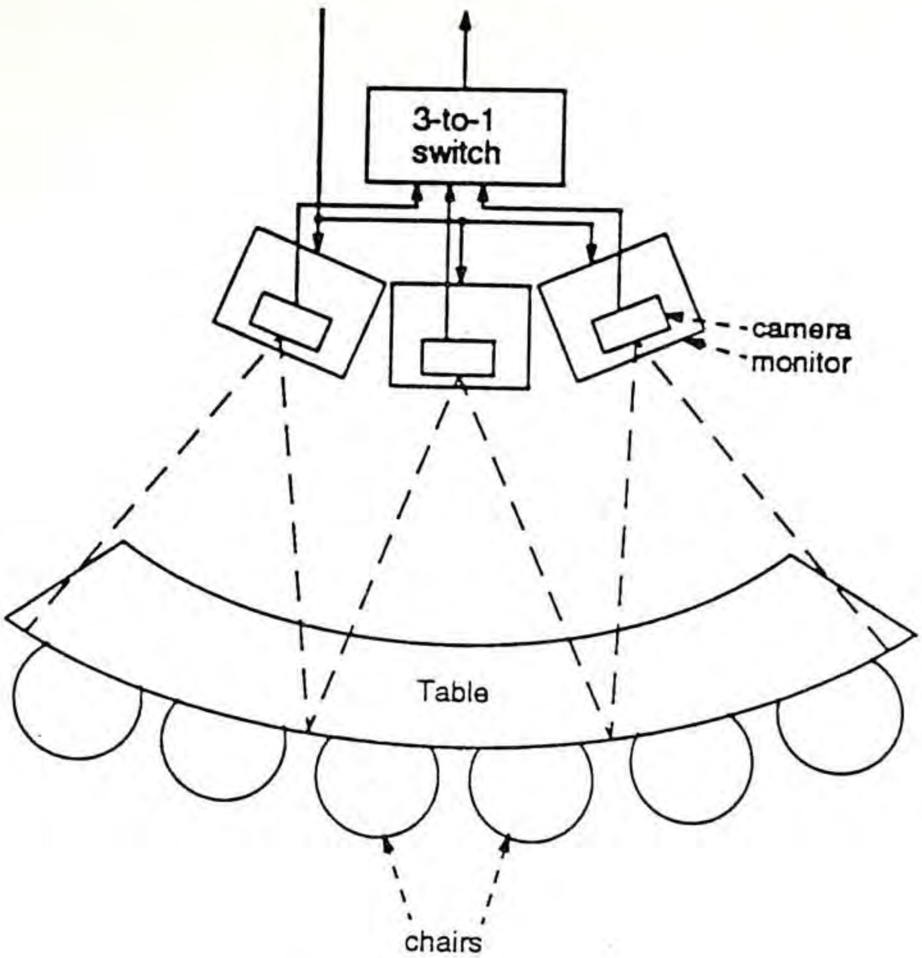


Fig.3: Voice switched videoconference system

C. Split Screen System [SEYL 73]

Split screen system is designed to connect two conference studios. Consider the example layout shown in Fig.4. The conferees are divided into two groups of three conferees each. The image of each group is captured by one camera. The two captured images are synthesized as one composite TV image, which is then transmitted to the other conference studio using one video channel. At the receiving end, the composite image is split into the original two images which are displayed on two monitors placed side by side.

This system provides continuous visual presence of all conferees located in the other conference site, but their spatial relationship is only partially maintained. For the same number of conference conferees, this system provides better resolution per conferee than the single camera system.

D. Continuous Presence System [BROW 80]

The continuous presence system is designed to connect two conference studios. Fig.5 shows a typical layout of a continuous presence system. The conference scene is partitioned into N segments such that each segment contains one or two conferees. The image of each segment is captured by one video camera. A total of N video channels is therefore needed to transmit the images of the N segments to the other conference studio. To reduce the total bandwidth requirement, the available bandwidth can be dynamically allocated to the N cameras based on their resolution requirements (e.g. allocate more bandwidth to that with large amount of motion or to the current speaker). The monitors are organized in a concave arrangement to give continuous spatial coverage, such that if a person walks across the room his appearance and disappearance on different monitors is consistent.

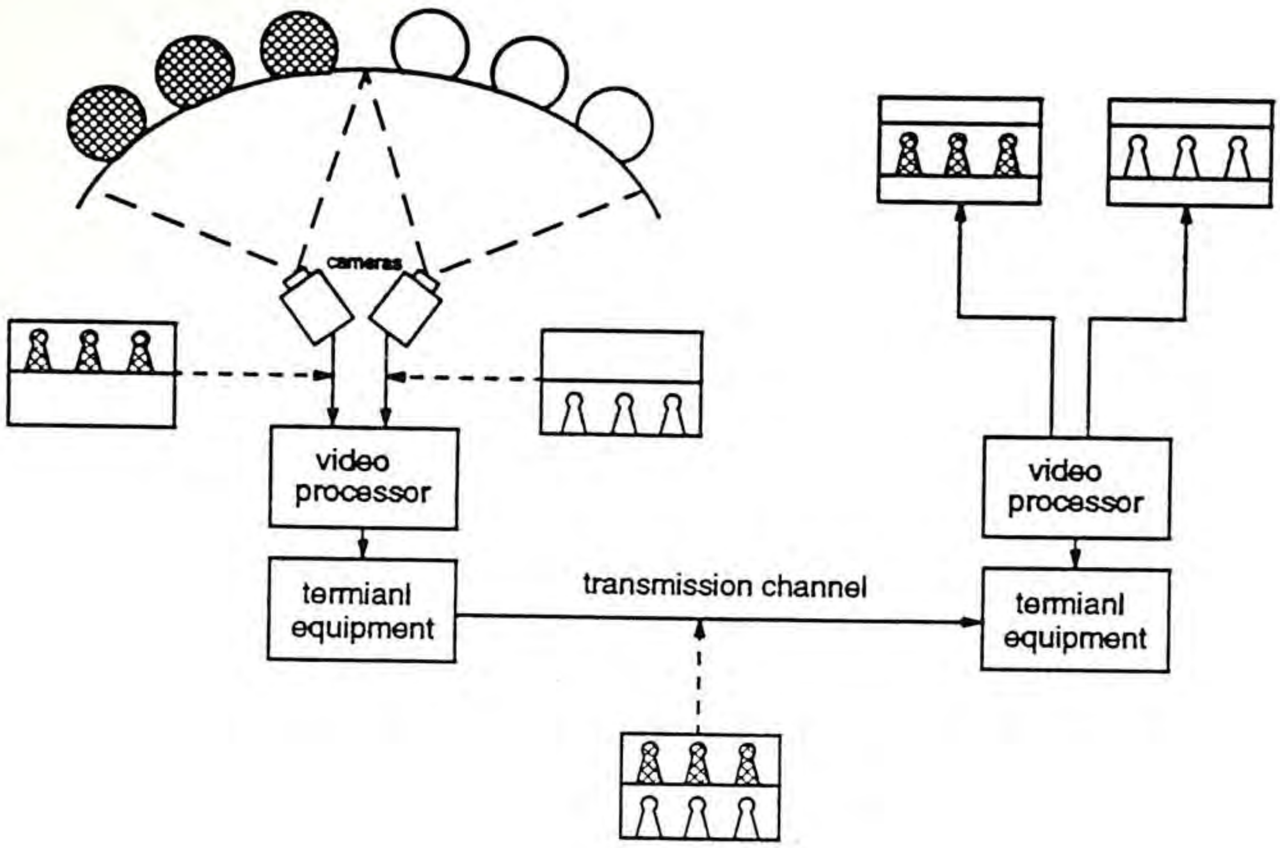


Fig.4: Split screen videoconference system

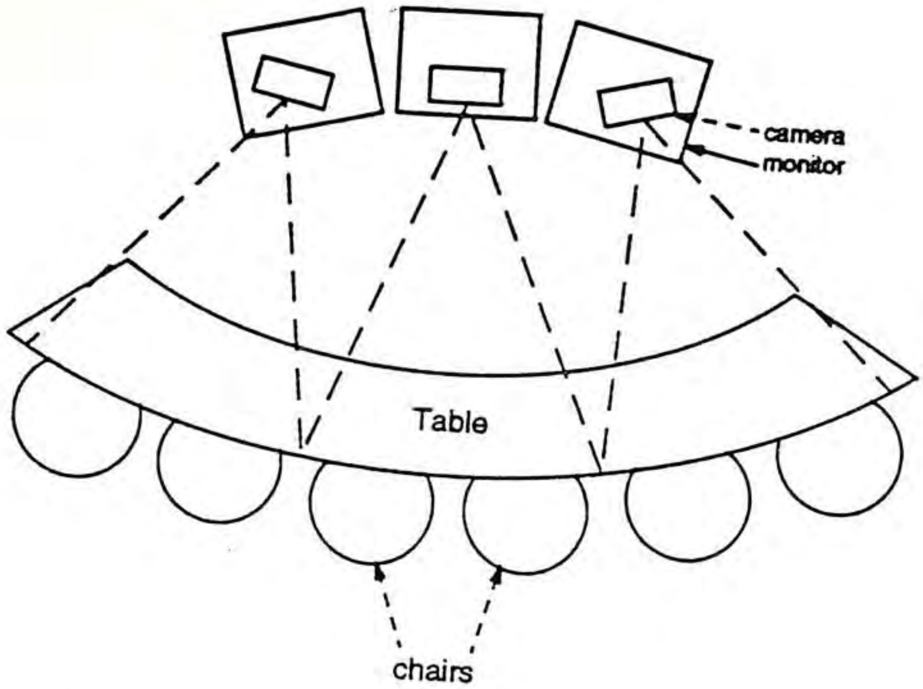


Fig.5: Continuous presence videoconference system

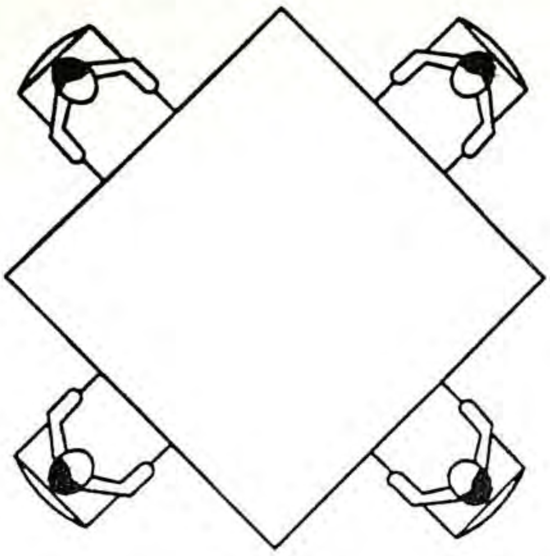
E. Virtual Space System [KELL 82]

The virtual space system is designed to connect multiple conference sites, to provide continuous visual presence of all conferees and to convey the spatial relationship among them. In other words, each conferee can view any conferee and can also determine where this conferee is looking. Consider the example shown in Fig.6. Fig.6(a) shows the simulated conference environment. Fig.6(b) depicts the setup needed for this simulated conference environment. The positions of the camera/monitor pairs in the four locations are arranged to provide a consistent virtual seating arrangement pattern as perceived by all conferees. When one conferee views another through the appropriate monitor at his location, he is presented with a viewing angle consistent with his virtual position in the remote room.

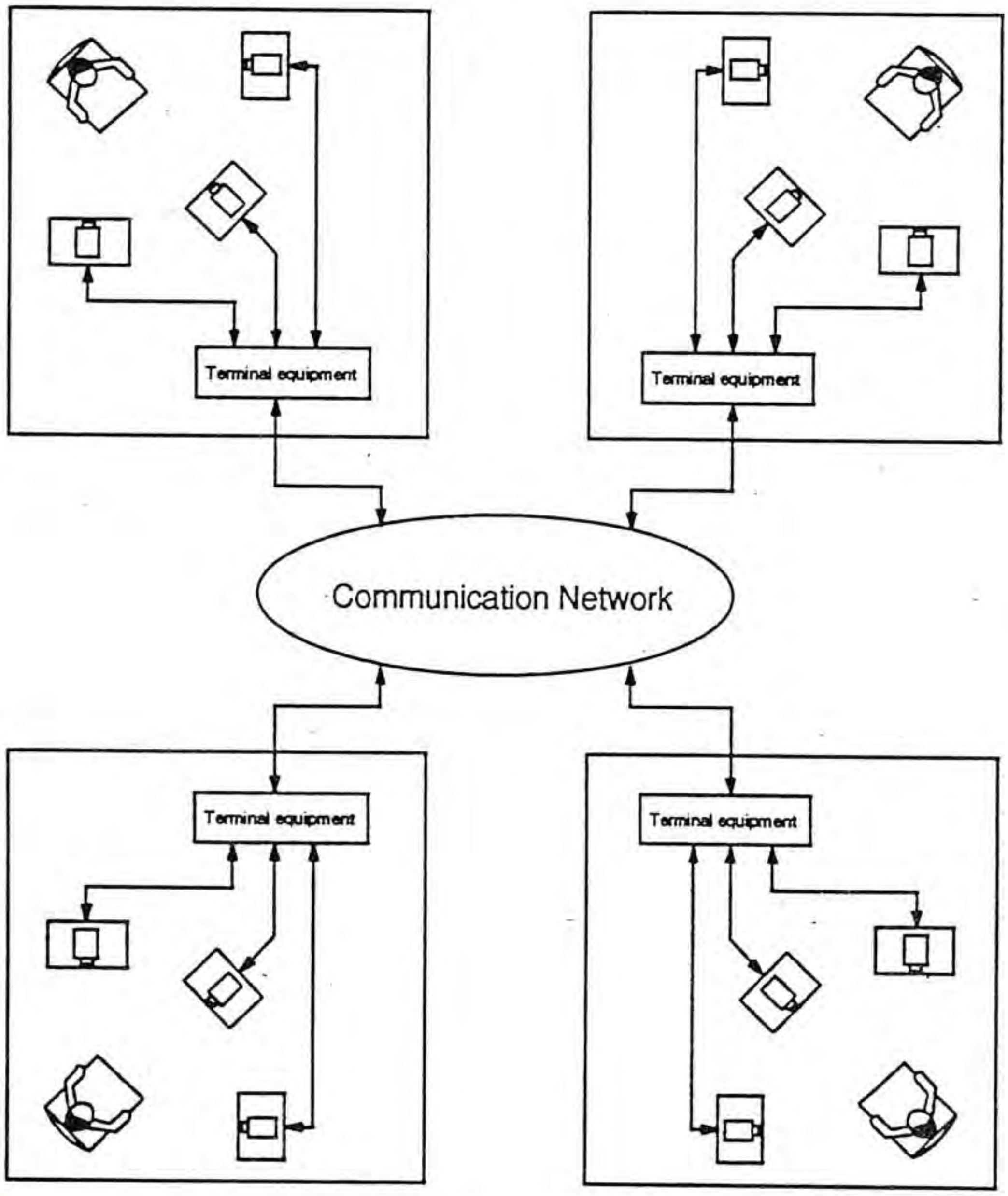
The biggest drawback of this system is that it is expensive in terms of equipment cost and transmission cost. In a conference with *N* conferees, each conference studio requires *N*-1 monitors, *N*-1 cameras and *N*-1 video channels.

F. Desk-Top Systems [KLEI 85, WEIS 90, ROBI 91]

This type of videoconference systems is designed to connect multiple conference sites. It emphasizes the use of general purpose hardware (e.g. a personal computer or workstation) as the terminal hardware, rather than using dedicated hardware placed in specially designed conference studio. This can reduce the system cost and allow the conferees to participate in a videoconference in their offices. This type of videoconference systems usually uses single monitor with segmented windows such that each window displays one conferee or displays multiple conferees located in the same conference site (e.g. see Fig.7).



(a) Simulated conference environment



(b) Connection of conference studios

Fig.6: Virtual space vieoconference system

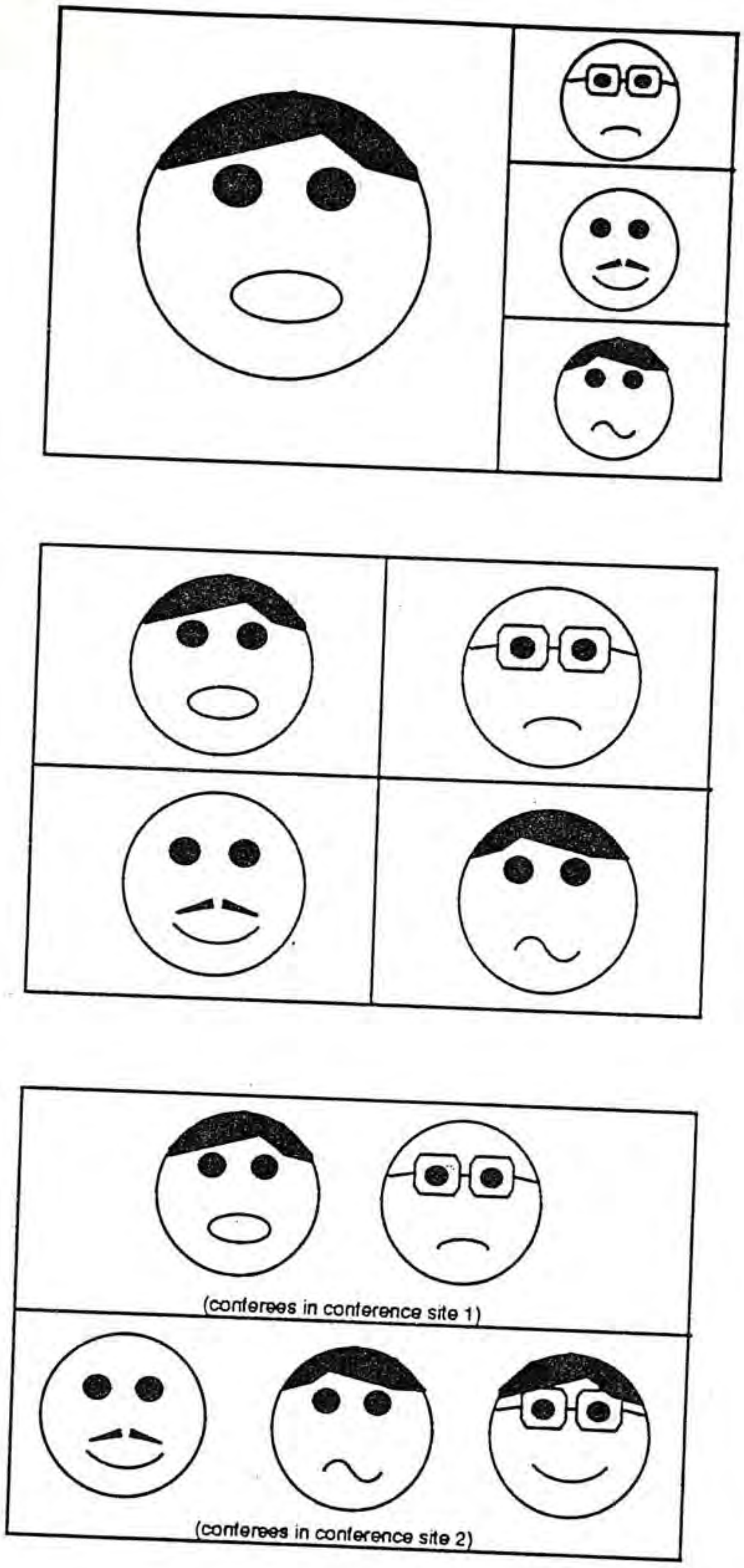


Fig.7: Examples of screens with segmented windows

G. Comparisons

Table 4 summarizes and compares the features of the six videoconference systems.

Systems	Connection	Video Communication				Equip.	Trans.
		Continuous Presence	Spatial Relationship	Eye-Contact	Resolution	Cost	Cost
Single Camera	point-to-point	yes	yes	partial	poor	low	low
Voice Switched	point-to-point	no	no	no	good	high	low
Split Screen	point-to-point	yes	partial	partial	good	high	low
Continuous Presence	point-to-point	yes	yes	partial	good	high	high
Virtual Space	multipoint	yes	yes	yes	good	very high	very high
Desk Top	multipoint	yes	no	no	medium	low	low

Table 4: Comparison of the six videoconference systems

1.5 Chapter Summary and Thesis Organization

In this chapter, we reviewed the broadband video services proposed in CCITT Recommendation I.211 and discussed their potential applications. Among the many proposed video services, videoconference service is considered to be one of the most "futuristics" services because the businessmen using this service can save both travel time and expenses and the business community can afford and is willing to pay higher service charges. We described six videoconference systems that were proposed in the literature and compared their merits and demerits.

In chapters 2 and 3, we tackle the routing issues of multipoint videoconferencing. In particular, we study a generic routing problem called *multiple destination routing* in chapter 2. In this problem, a minimum cost connection path is to be found to connect a given set of nodes of a network. We propose an optimal solution technique and three heuristic algorithms for this problem. In chapter 3, we formulate the connection optimization problem for two types of videoconferences. Algorithms that use a combination of table look-up and on-line processing are designed to compute the minimum cost connection path for connecting a given set of conference sites.

In chapters 4 and 5, we tackle the switching issues of multipoint videoconferencing. In chapter 4, we propose and analyze a TDM-based multibus packet switch for high speed packet switching. Based on this multibus design, we propose in chapter 5 a video switch architecture for distributing broadcast and switched videos.

In chapter 6, we summarize the work reported in this thesis and identify a number of issues for further investigations.

Chapter 2

Efficient Algorithms for Multiple Destinations Routing

A number of important problems in computers and telecommunication such as multipoint videoconferencing and distributed database all require the solution of a generic problem called multiple destinations routing (MDR). In this problem, a minimum cost connection path is to be found to connect a given set of nodes (called conference nodes) of a network. This problem has been shown to be NP-complete. In this chapter, we formulate the MDR problem as a zero-one integer programming problem and propose a method to reduce the number of enumerations required for optimal solutions. Numerical results show that, when the number of conference nodes is small (say ≤ 5), this method can find the optimal solutions using reasonable computation time. This method is therefore suitable for multipoint videoconference application because the number of conference nodes in each conference is presumably not large. Three heuristics are designed for large MDR problems. Heuristic A can offer different degrees of optimality with different amount of time allowed for the solution. Heuristic B is modified from the Prim's algorithm for finding minimum spanning tree, taking advantage of the property that two or more paths may share common edges. It gives a fairly good solution with very little computation. Heuristic C is for networks with uniform edges weights and overcomes the deficiency of heuristic B. It always gives a better solution than Heuristic B. Simulation on two example networks shows heuristics A and C always give better solutions (or lower cost connection paths) than the Improved RS algorithm, which is up to now the best heuristic.

2.1 Background

In recent years, there is an increase in the number of applications requiring group-based communications. These applications can be characterized in a common way: a source node multicasts to multiple destination nodes. In updating a distributed database, for example, an updated data item is sent to the nodes where the duplicated copies of the data item are resided. Multicast is also required for a class of multipoint broadband services where the video signals from a source node are distributed to multiple destinations (e.g. multipoint videoconferences and remote video lectures). To minimize the use of network resources, minimum "cost" connection paths should be used. The problem of finding such paths is known as the multiple destinations routing (MDR) [KADO 83] problem in the communication literature and Steiner tree problem in the graph theory literature [GARE 79].

The MDR problem has been shown to be NP-complete [GARE 79]. Several heuristics for this problem were proposed in the literature and we describe these heuristics in the following subsections. Let $G=(V,E)$ be a graph where V and E are the sets of nodes and edges of this graph respectively, and let Ω be a set of nodes ($\Omega \subseteq V$) such that the nodes in Ω are to be connected.

A. MST Algorithm [WALL 80, KOU 81, BHAR 83, AHMA 90]

MST algorithm is probably the most well known heuristic for the MDR problem (e.g. see [WALL 80, KOU 81, BHAR 83, AHMA 90]). The idea of the MST algorithm is quite simple and it uses any minimum spanning tree algorithm (e.g. see [AHO 83]) to find a suboptimal solution. The MST algorithm is given below:

<u>MST Algorithm</u>

Step 1: Construct a complete graph $G'=(\Omega,E')$ from G, where the cost of edge uv in E' is the length of the shortest path from node u to node v in G.

Step 2: Construct a minimum spanning tree T for G'.

Step 3: Convert the edges in T to paths in G to form a solution T^*.

Fig.1 shows an example to illustrate the execution of the MST algorithm.

B. Improved MST Algorithms [KOU 81, WAXM 88]

Kou, Markowsky and Berman [KOU 81] proposed a method to improve the MST algorithm. The idea is to find a minimum spanning tree for the nodes in T^* (the solution given by the MST algorithm) and then delete those redundant edges in the resulting minimum spanning tree in order to form a tree such that all the leaves of this tree are the nodes in Ω.

Waxman [WAXM 88] proposed another method for further improving the MST algorithm: rerun the MST algorithm for the nodes in T^* to give a subgraph and then delete those redundant edges in this subgraph in order to form a tree such that all the leaves of this tree are the nodes in Ω.

C. RS Algorithm [RAYW 83, RAYW 86, WAXM 88]

RS algorithm was proposed by Rayward-Smith [RAYW 83, RAYW 86]. Initially, let any node in Ω be called a subtree. The RS algorithm is given below:

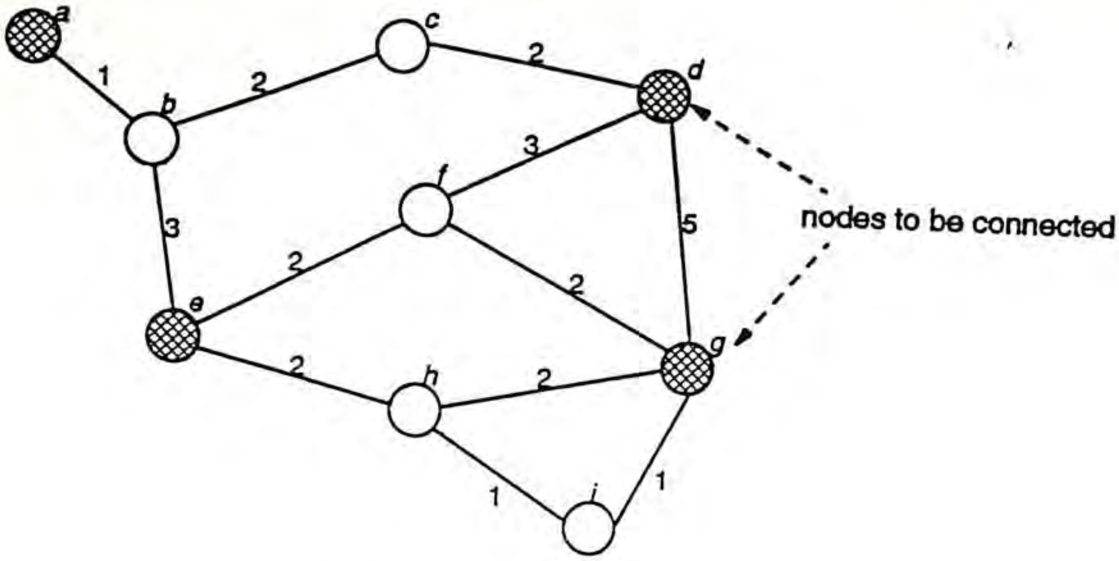
<u>RS Algorithm</u>

Step 1: Find a node v in G such that the average cost to connect node v to the existing subtrees is the smallest.

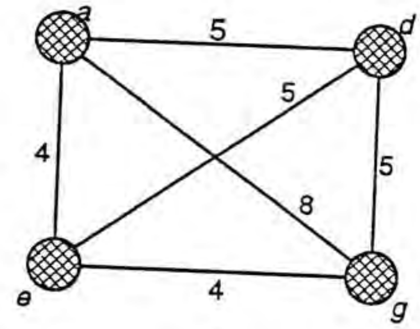
Step 2: Let T_1 and T_2 be the two closest subtrees to node v. Join T_1 and T_2 by a shortest path through v to form a new subtree.

Step 3: Repeat steps 1-2 if there are two or more subtrees.

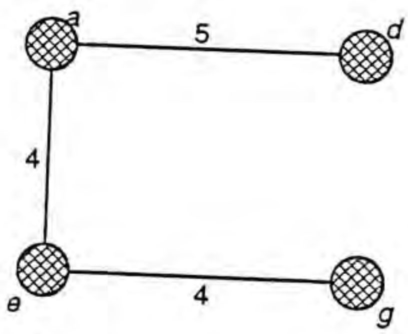
Fig.2 shows an example to illustrate the execution of the RS algorithm.



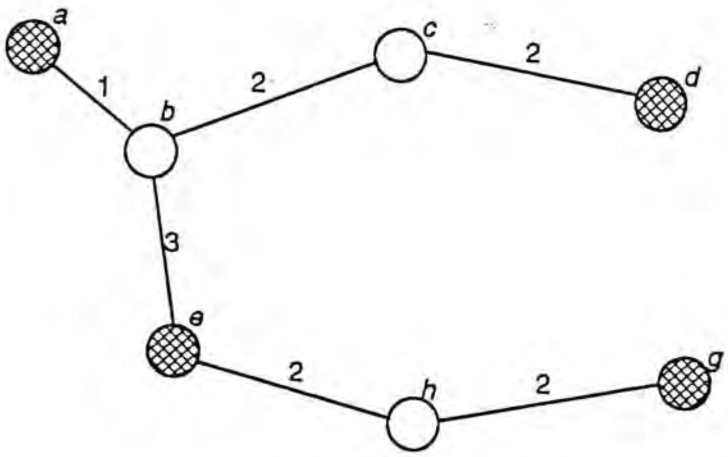
(a) Graph G



(b) Graph G'



(c) Minimum spanning tree for G'



(d) solution; cost=12

Fig.1: An example illustrating the execution of the MST algorithm

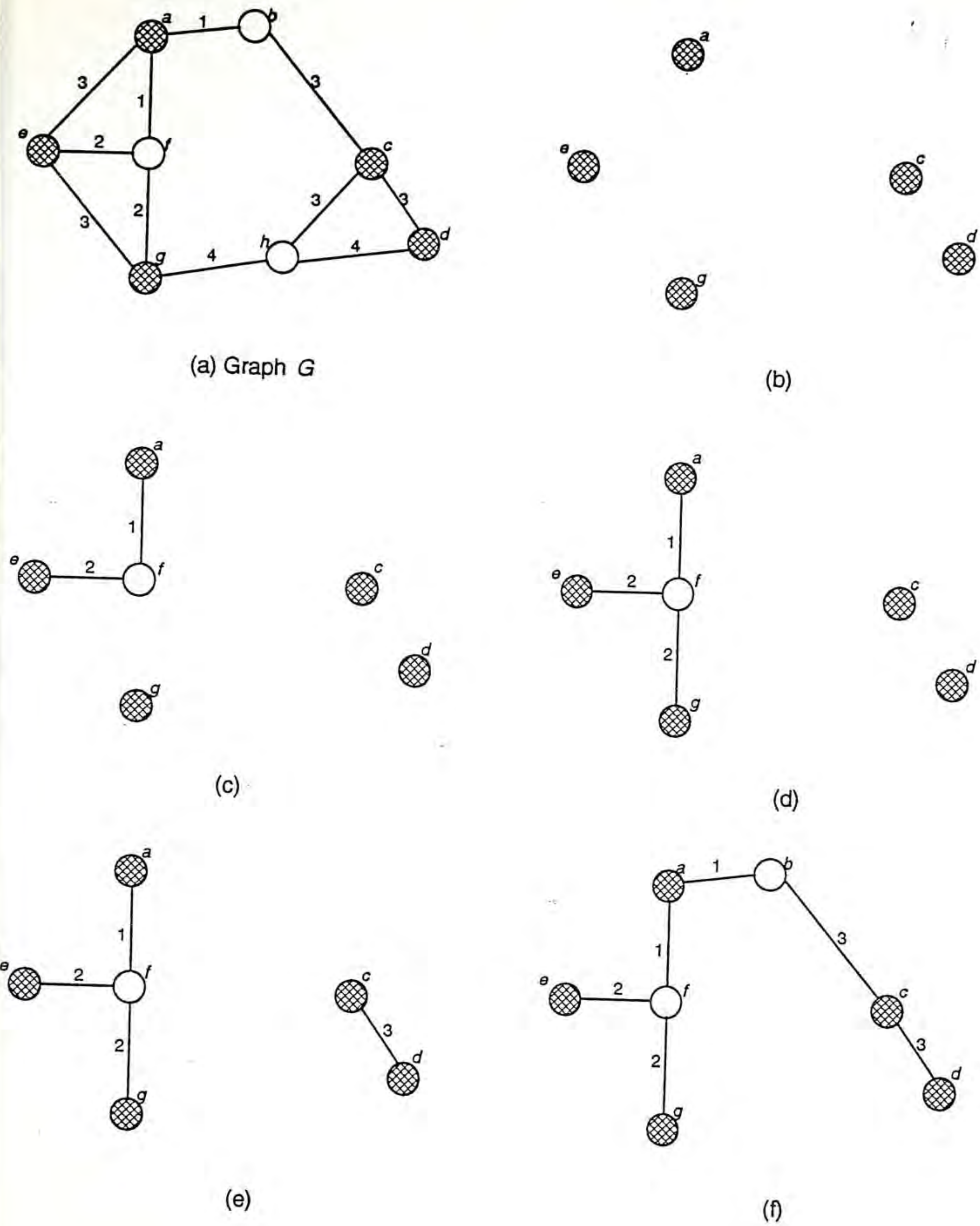


Fig.2: An example illustrating the execution of the RS algorithm

Waxman [WAXM 88] proposed an improved version of the RS algorithm: rerun the RS algorithm for the nodes in the tree found by the RS algorithm. Of all the algorithms reported in the literature, this improved RS algorithm was shown through simulation studies to have the best performance [WAXM 88].

In this chapter, we formulate the MDR problem as a zero-one integer programming problem and propose a method to reduce the computation required for optimal solutions. Numerical results show that, when the number of conference nodes is small (say ≤ 5), this method can find the optimal solutions using reasonable computation time. This method is therefore suitable for multipoint videoconference application because the number of involved nodes in each conference is presumably not large. Moreover, for many multipoint broadband services (such as videoconferencing and remote video lecture) the communication is prescheduled and the set of nodes to be connected is known in advance. These applications allow more time to determine the optimal connection paths.

For applications where the number of nodes to be connected is large, the search for optimal connection path becomes infeasible. For this reason, we design three heuristics that find close to optimal connection paths. We show by simulation that two of these heuristics compare favourably to the Improved RS heuristic [WAXM 88]. Besides, one of the three heuristics has a nice feature: it has a parameter k which allows us to trade off between optimality and computation time. A larger k can give closer to optimal solution but requires longer computation time. With a faster computer or for prescheduled communication applications (e.g. prescheduled videoconferences or remote video lectures), we can just increase k to get a closer to optimal solution. For applications that require fast determination of the connection paths (e.g. immediate updating of distributed database), we can set k to a small value.

2.2 Integer Programming Formulation of the MDR Problem

For the formulation of the MDR problem we model the communication network as a weighted graph $G=(V,E)$ where V is the set of network nodes and E is the set of edges. Let node i be denoted as n_i, the edge connecting n_i and n_j as e_{ij} and the weight of e_{ij} as w_{ij}. Consider the connection of a source node n_o to a set of destination nodes with i.d.'s specified by the set D. Borrowing the terminology from the videoconferencing application, we shall, for convenience, call the source node and the set of destination nodes collectively as the set of *conference nodes*. The corresponding graph theory terminology is Steiner nodes [GARE 79, GILB 68]. The connection between the source and destination nodes forms a multicast tree in the network. Nodes in the multicast tree which are not conference nodes are called *linking nodes*. Edges that appears in the multicast tree are called *linking edges*. A *minimum multicast tree* (MMT) is a multicast tree with minimum weights. Its optimality is not affected when the role of any destination node is exchanged with the role of the source node.

Let S_i be the set of paths[1] connecting the source node and the destination node n_i ($i \in D$), $|S_i|$ be the number of such paths and $P_i^{(j)}$ be the set of edges in the j^{th} path. In addition, let $x_i^{(j)}$ be a binary variable defined as:

$$x_i^{(j)} = \begin{cases} 1 & \text{the } j^{\text{th}} \text{ path is used to connect } n_o \text{ and } n_i \\ 0 & \text{otherwise} \end{cases}$$

and let $\boldsymbol{x}$ be the set of $x_i^{(j)}$'s with $i \in D$ and $1 \le j \le |S_i|$. Let the dot operation on any set A be defined as:

$$\begin{aligned} 1 \cdot A &= A \\ 0 \cdot A &= \phi \quad \text{(empty set)} \end{aligned}$$

1 All paths referred in this chapter are elementary paths, i.e. paths that does not meet the same vertex twice

and let Y, defined as

$$Y = \bigcup_{i \in D} \bigcup_{\substack{1 \le j \le |S_i| \\ \sum_{j=1}^{|S_i|} x_i^{(j)} = 1}} x_i^{(j)} \bullet P_i^{(j)} \tag{1}$$

be the set of edges on the multicast tree. Using the above definitions, we can formulate the MDR problem as the following zero-one integer programming problem:

$$\text{Minimize} \qquad W(x) = \sum_{mn: e_{mn} \in Y} w_{mn}$$

The constraint on Y in (1) assures that only one path is selected to connect n_o and a destination node. With this constraint, the objective function $W(x)$ is the sum of all the edge weights on the multicast tree. This constraint reduces the number of enumerations from $\prod_{i \in D} 2^{|S_i|}$ to $\prod_{i \in D} |S_i|$ as when $x_i^{(k)}=1$, $x_i^{(j)}=0$ for all $j \neq k$.

The solution of any heuristics (see section 2.3) can serve as an upper bound on $W^*(x)$, the minimum tree weight. Enumerating paths $P_i^{(j)}$ with weights greater than the upper bound is therefore not needed. In other words, if the weight of path $P_i^{(j)}$ is larger than the upper bound, $x_i^{(j)}$ is set to zero. This reduction is especially important when the network is large and network connectivity is high.

As all enumerations can be performed independently, this enumerative approach is well suited for parallel implementation.

2.3 Heuristics For Multiple Destinations Routing

A. Heuristic $A^{(k)}$

The path connecting the source node and a destination node should not be too much longer than the shortest paths connecting them because otherwise the savings gained by the sharing of common edges with other paths (e.g. see Fig. 3) might not be enough to cover the "loss" of using a longer path. Based on this property, we can employ the enumerative approach similar to that given in section 2.2, but restricting the lengths of all paths to be enumerated. This approach can significantly reduce the number of enumerations but does not guarantee optimality. Based on this approach, we design heuristic $A^{(k)}$.

Let α_i be the length of the shortest path connecting the source node and a destination node $n_i, i \in D$. Let Z and M be sets. Initially, both Z and M contain all the conference nodes. Heuristic $A^{(k)}$ is given below:

<u>Heuristic $A^{(k)}$</u>

Step 1: Select one of the nodes in Z as the source node and denote it as n^*;
$M \leftarrow M \backslash \{n^*\}$.

Step 2: From n^* to n_i, $i \in M$, enumerate only the paths with lengths smaller than or equal to $\alpha_i + k$. Find the multicast tree with the smallest weights for this source node n^*.

Step 3: $Z \leftarrow Z \backslash \{n^*\}$;
$M \leftarrow M \cup \{n^*\}$.

Step 4: Repeat step (1)-(3) until $Z = \phi$.

Step 5: Among the multicast trees found in Step (2), select the one with the smallest weights.

The parameter k is used to restrict the lengths of the paths to be searched. A larger k can give a closer to optimal solution but requires longer computation time. Hence, we can trade off between optimality and computation time easily. For the same value of k, when different conference nodes act as the source node, trees with different weights may result. Hence by enumerating all possible source node heuristic $A^{(k)}$ selects the tree that gives closest to optimal solution.

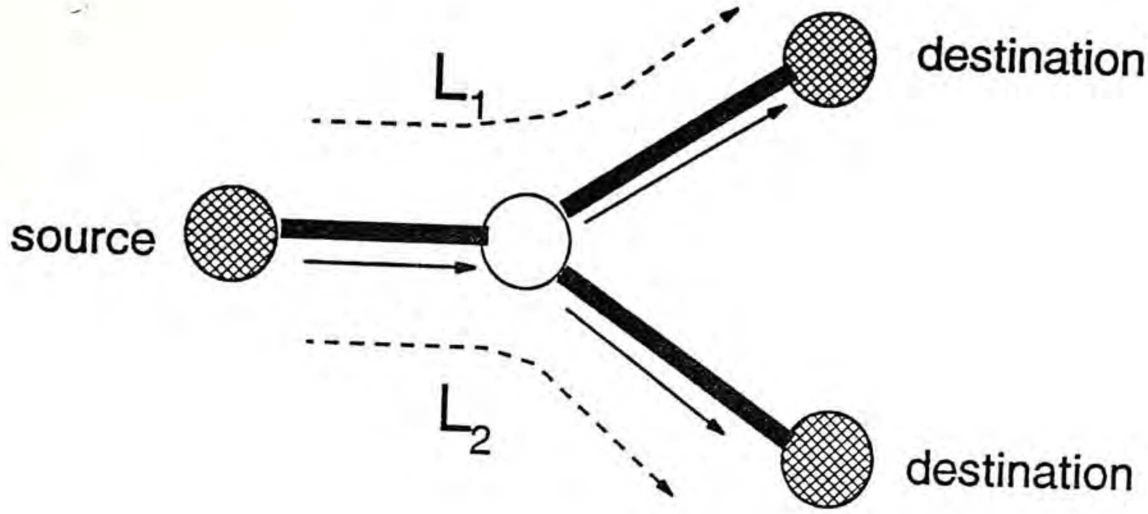


Fig.3: Paths L_1 and L_2 can share a common edge

As all enumerations in heuristic $A^{(k)}$ can be performed independently, it is therefore well suited for parallel implementation.

B. Heuristic B

Heuristic B is modified from the Prim's algorithm [SYSL 83] for finding minimum spanning trees. Prim's algorithm finds in each iteration a shortest *edge* that connects an unconnected node to the existing subtree. Heuristic B, on the other hand, finds in each iteration a shortest *path* that connects an unconnected conference node to the existing subtree, taking advantage of the property that an edge may be shared by two or more paths. Let C, S and T be sets. Initially, S contains all the conference nodes and T and C are empty sets. The outputs are T and C which contain the nodes and edges of the multicast tree respectively.

<u>Heuristic B</u>

Step 1: Find the shortest path P connecting any two nodes in S. Denote the two nodes as n^* and n^{**}.

Step 2: $S \leftarrow S \backslash \{n^{**}\}$;

GOTO Step 4.

Step 3: Search for the shortest path P that connects a node in S to the existing subtree defined by (T,C). Denote the node concerned as n^*.

Step 4: $S \leftarrow S \backslash \{n^*\}$;

$T \leftarrow T \cup \{\text{nodes on path } P\}$;

$C \leftarrow C \cup \{\text{edges on path } P\}$.

Step 5: Repeat Step (3)-(4) until $S = \phi$.

As an example on the use of heuristic B consider the network shown in Fig.4(a). The weights of all edges are equal. The sequence of edges selected is shown from (b) to (d). The resulting multicast tree is shown in (e).

Prim's algorithm can give optimal solution because spanning trees possess the matroidal properties [SYSL 83] such that a local optimum turns out to be the global optimum. However, multicast trees do not have this property. In fact the best path to connect any two conference nodes may not be the shortest path (e.g. in Fig.5 the shortest path L_1 is not the best path).

When there are more than one shortest path connecting a conference node to the existing tree, heuristic B selects one of these shortest paths randomly and hence it does not give a unique solution. Some of them may have larger weights (e.g. see Fig.6). This problem can be relieved by using backtracking at the price of longer search time. We call this heuristic B with back-tracking.

C. Heuristic C for Minimizing the Number of Edges In the Multicast Tree

When the weights of all edges are equal, the problem becomes the minimization of the number of edges in the multicast tree. Under this condition, heuristic B can be further improved as follows. When the algorithm cannot determine which one of the many shortest paths is the best in each iteration, all such paths are chosen. After all conference nodes are connected, redundant linking nodes and linking edges are removed following a set of four rules. We call this heuristic C.

Among the set of linking nodes to be removed we consider only those that will not disconnect the conference nodes. The other linking nodes and the conference nodes are collectively called the *must-be-present nodes*. Also, when a linking node is removed, so is the set of edges attached to it.

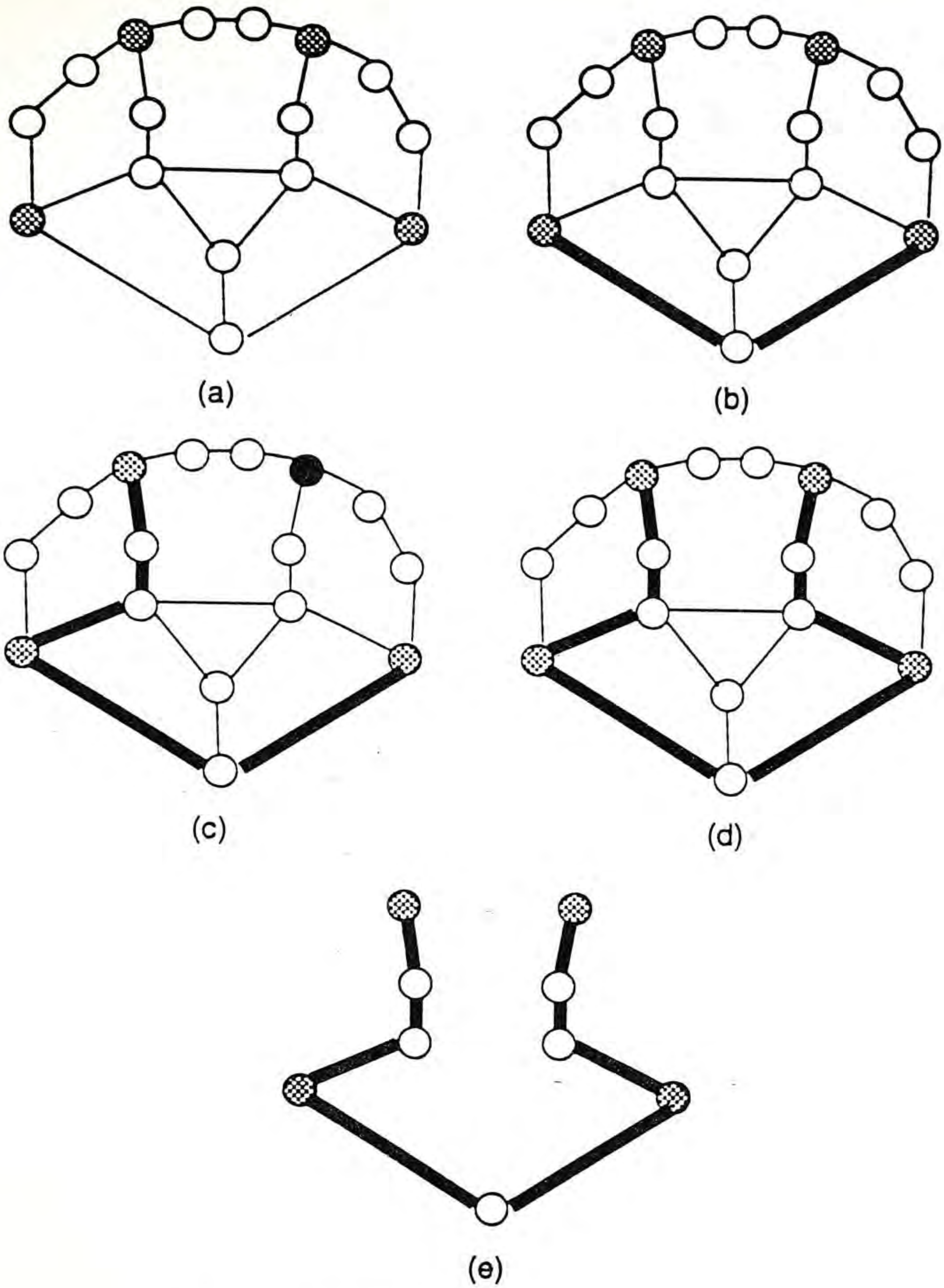


Fig.4: An example illustrating heuristic B

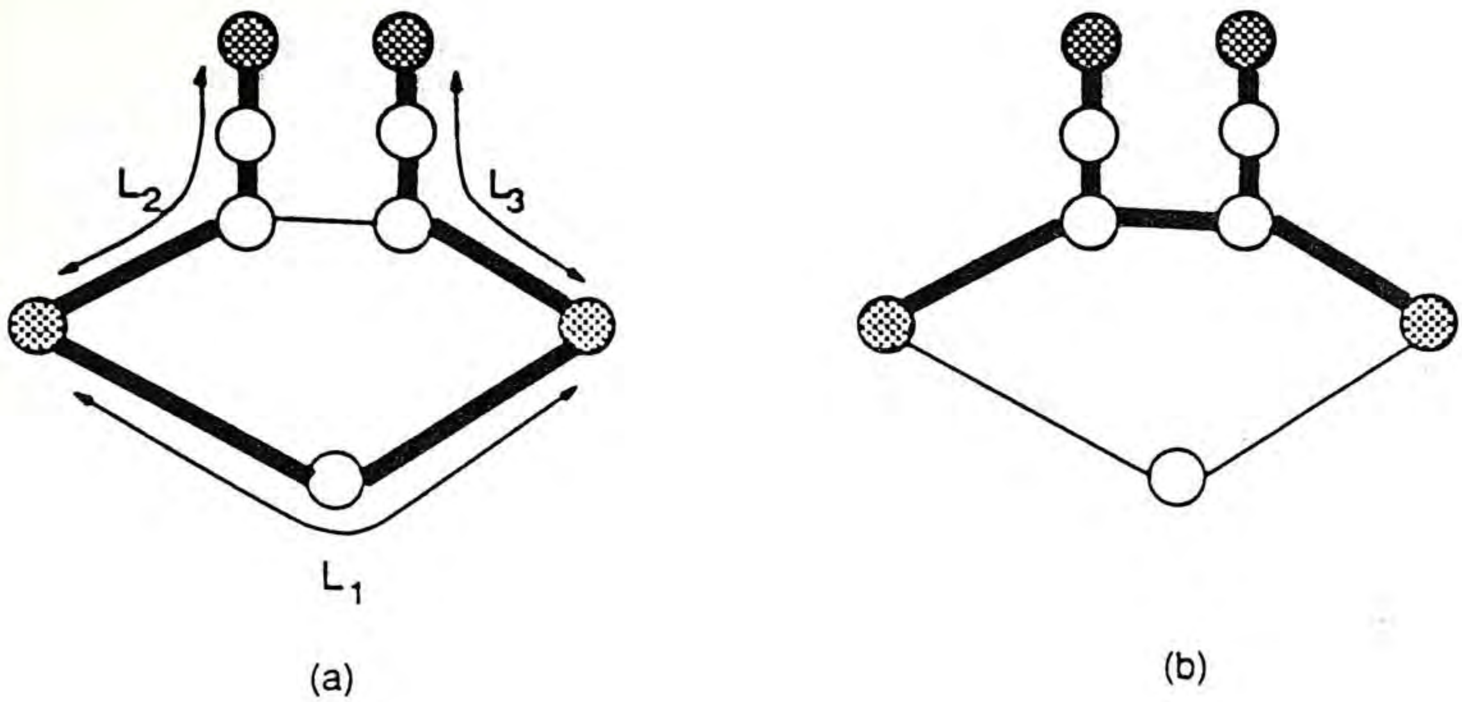
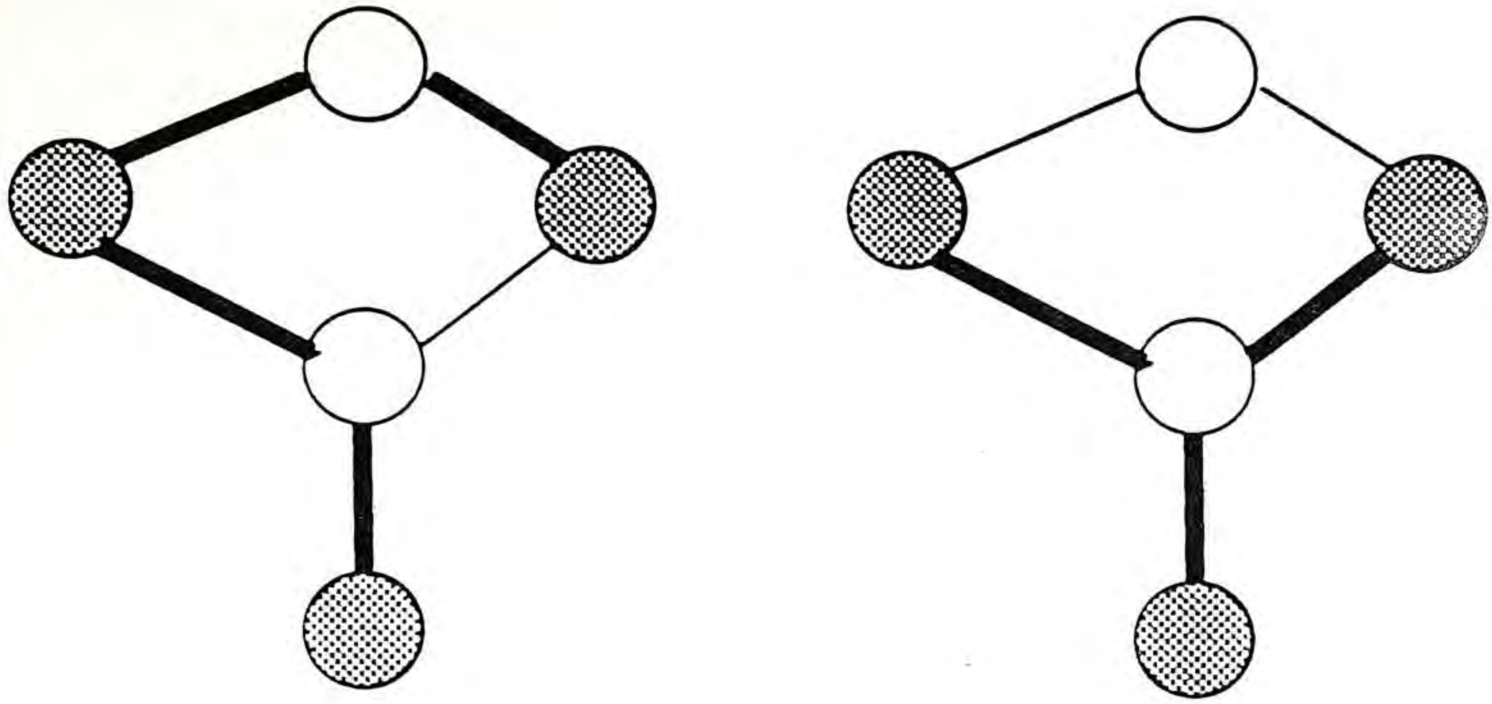


Fig.5: (a) Heuristic B selects L_1, then L_2 and finally L_3
(b) Optimum connection



(a) (b)

Fig.6: Both (a) and (b) are solutions of heuristics B

The basis of rule 1 is as follows. A linking node with the largest number of linking edges is likely to have a large number of paths passing through it (e.g. see Fig.7). As a result, we should remove the linking node with the smallest number of linking edges. If there are more than one, rule 2 is used for further resolving.

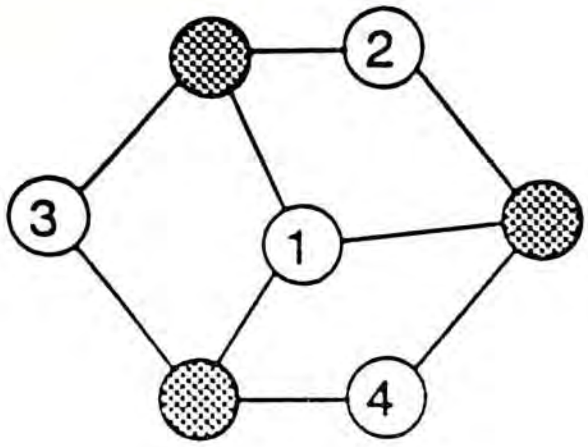
The basis of rule 2 is as follows. Among the linking nodes with the smallest number of linking edges, denote the one that is directly connected to a large number of must-be-present nodes as n^*. Node n^* will likely connect a large number of the must-be-present nodes directly. In other words, a large number of paths connecting the must-be-present nodes should pass through n^* and thus can share common edges (e.g. see Fig.8). Hence, among the linking nodes with the smallest number of linking edges, we should remove the one that is connected to the smallest number of linking edges. If there are more than one, rule 3 is used for further resolving.

The basis of rule 3 is as follows. Among the linking nodes not yet resolved by rule 2, the node n^{**} that is connected to a linking node with the largest number of linking edges is likely to have many paths passing through it (e.g. see Fig.9). In other words, n^{**} is on the path that contains many common edges. Therefore node n^{**} should be retained.

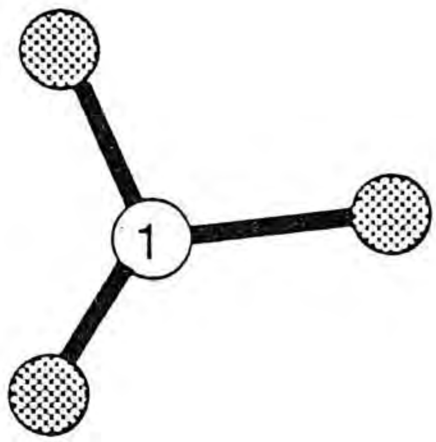
The basis of rule 4 is as follow. Since rules 1, 2 and 3 only remove the edges associated with the redundant nodes, there can be some redundant edges that cannot be removed by these rules. But note that after the redundant linking nodes have been removed, the remaining nodes are the nodes that must be present in the final multicast tree. Hence we can simply use minimum spanning tree algorithms (e.g. see [SYSL 83]) to remove the remaining redundant linking edges.

The four rules for removing the redundant linking nodes and linking edges are summarized as follows:

Rule 1: If there is a linking node with the smallest number of linking edges, remove that node. Otherwise, employ rule 2.

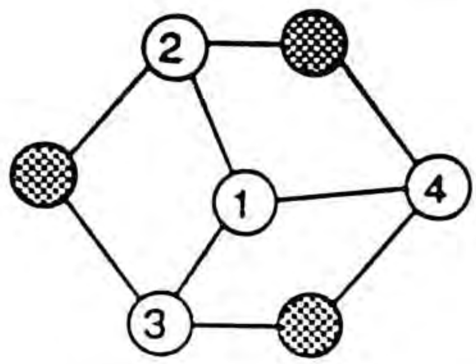


(a) Among the four linking nodes, node 1 has the largest number of linking edges. Node 1 should be retained.

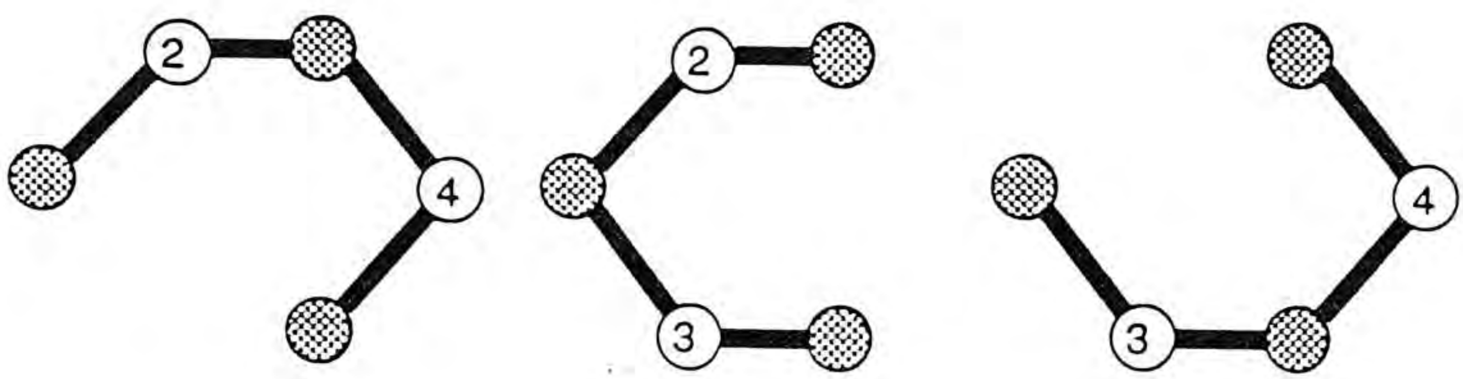


(b) Optimum connection

Fig.7: An example illustrating rule 1 of heuristic C

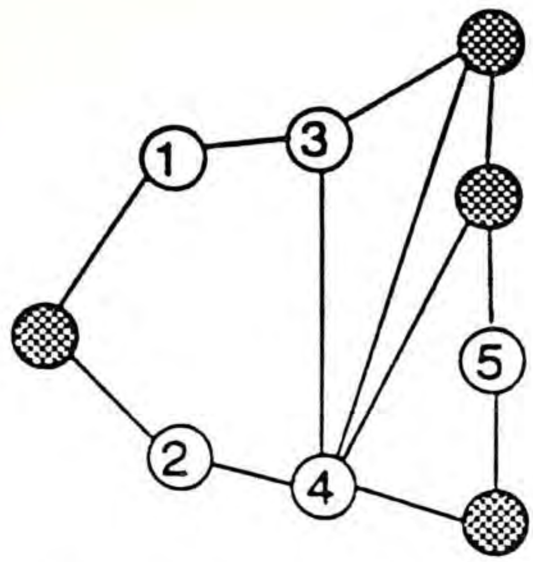


(a) The four linking nodes have the same number of linking edges. Only node 1 is not directly connected to the conference nodes. Node 1 should be removed.

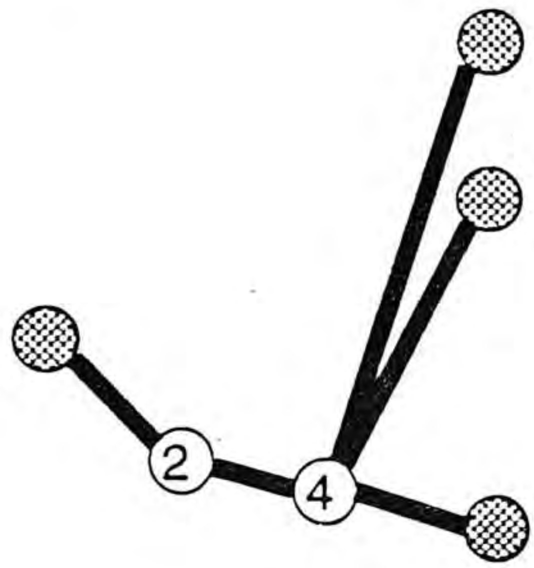


(b) Optimum connections

Fig.8: An example illustrating rule 2 of heuristic C



(a) Among the five linking nodes, nodes 1, 2 and 5 are not resolved by rule 1; and nodes 1 and 2 are not resolved by rule 2. Rule 3 removes node 1.



(b) Optimal connection.

Fig.9: An example illustrating rule 3 of heuristic C

Rule 2: Among the linking nodes with the smallest number of linking edges, if there is one that is directly connected to the smallest number of must-be-present nodes, remove that node. Otherwise, employ rule 3.

Rule 3: Among the linking nodes not yet resolved by rule 2, retain the nodes that are directly connected to the linking nodes with the large number of linking edges.

Rule 3: Remove the redundant linking edges by minimum spanning tree algorithm.

Let S, P and T be sets of nodes and C be a set of edges. Initially, S and P both contain the set of conference nodes, and T and C are empty. The outputs are P and C which contain the nodes and edges of the multicast tree respectively. Heuristic C is given below:

Heuristic C

Step 1: Search for two nodes n^* and n^{**} in S such that the path connecting them is shortest. Let the number of shortest paths connecting n^* and n^{**} be m and denote the paths as P_i, i=1,2,...,m.

Step 2: $S \leftarrow S \setminus \{n^{**}\}$;

GOTO Step 4.

Step 3: Search for a node n^* in S such that the path connecting it to the subgraph (T,C) is shortest. Denote the shortest paths connecting n^* and (T,C) as P_i, i=1,2,...,m.

Step 4: $S \leftarrow S \setminus \{n^*\}$;

$T \leftarrow T \cup \{\text{nodes on } P_i, i=1;2,...,m\}$;

$C \leftarrow C \cup \{\text{edges on } P_i, i=1,2,...,m\}$.

Step 5: Repeat Step (3)-(4) until $S=\phi$.

Step 6: $R \leftarrow T \setminus P$.

Step 7: $X \leftarrow \{\text{nodes in } R \text{ that must be present to connect the conference nodes}\}$;

$R \leftarrow R \setminus X$;

$P \leftarrow P \cup X$.

Step 8: Remove a node in R and its associated edges in C according to rules 1,2 and 3.

Step 9: Repeat Step (7)-(8) until $R=\phi$.

Step 10: Remove the redundant linking edges by rule 4.

Step 1 to step 5 connect the conference nodes. Redundant linking nodes and linking edges are added in this stage. Step 6 to step 10 remove these redundancies. In step 7, the set P contains the must-be-present nodes, and the set R contains the linking nodes that are not in set P. This heuristic does not give a unique solution. We can again use backtracking to find better solutions.

As an example illustrating heuristic C, consider the network in Fig.10(a). The sequences of edges selected are shown in (b) to (d). (e) shows the topology after all the conference nodes have been connected. (f) to (l) show the steps by which the redundant nodes and edges are removed. When all the redundant linking nodes are removed, there is no redundant linking edge. So, in this case, we do not need to run the minimum spanning tree algorithm. The resulting multicast tree shown in (l) is optimal while that obtained by heuristic B (Fig.5) is not.

2.4 Performance Comparisons

We compare the average performance of the heuristics via simulation on two example networks. The first network shown in Fig.11(a) has 30 nodes and an average connectivity of 2.9. The weights of all edges are one. The second network shown in Fig.11(b) has 19 nodes and an average connectivity 2.6. It is based on an early version of the ARPANET topology [KADO 83]. The conference size varies from 2 to 10 nodes and the nodes are randomly located in the network. In case the heuristics give non-unique solution, the average of all possible solutions are taken.

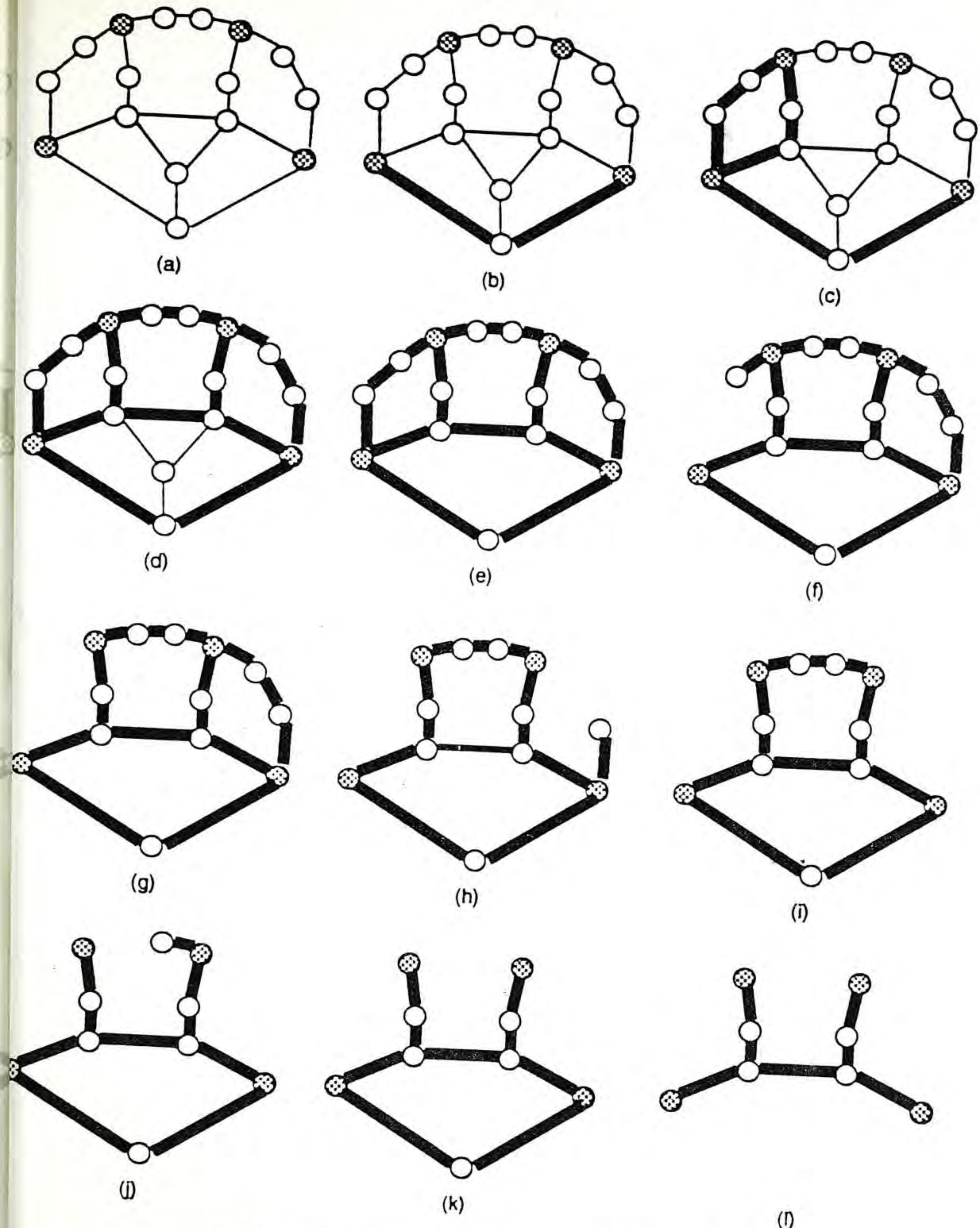
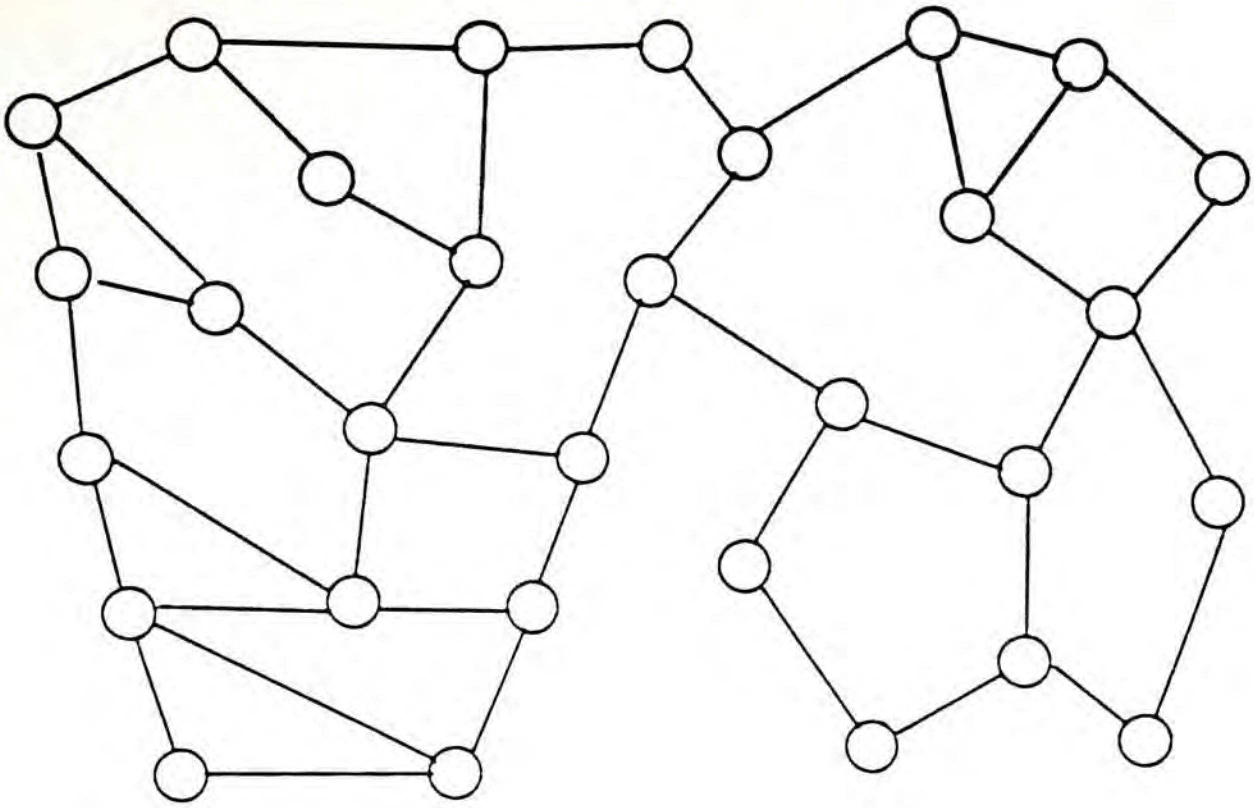
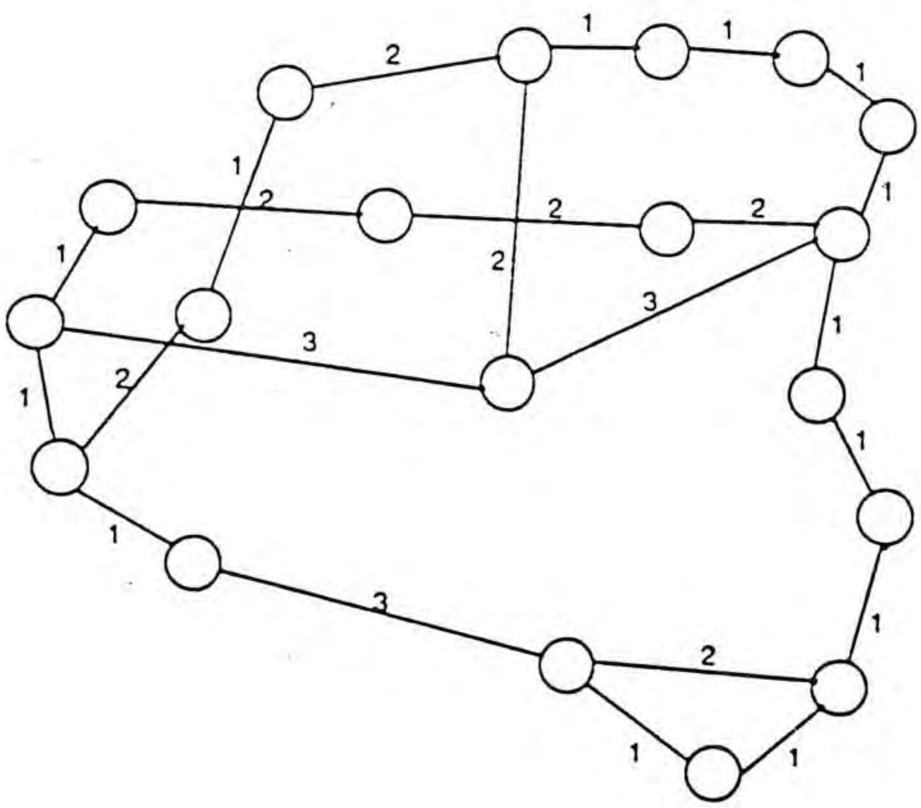


Fig. 10: An example illustrating heuristic C



(a) Network I



(a) Network II

Fig.11: Two example networks

Fig.12 shows the number of enumerations required for the optimal solution as a function of conference size. By employing the upper-bounding technique, the number of enumerations is reduced by about five orders of magnitude for network I and two orders for network II. Hence, this reduction is more significant for large networks with high network connectivity. For five-nodes conferences, the number of enumerations is about 10^5 for network I and 10^3 for network II, as compared to 10^{12} and 5×10^5 with constrained exhaustive enumerations.

Fig.13 shows the average performance of heuristic $A^{(k)}$. The y-axis shows the *normalized weight*, which is defined as the ratio of the multicast tree weight obtained by the heuristics to the optimal multicast tree weight. Heuristic $A^{(1)}$ for network I and heuristic $A^{(2)}$ for network II can already give optimal solutions in our cases. In other words, the optimal paths are either the shortest paths or the next shortest paths. This confirms our conjecture that a good path for connecting a source node to a destination node should not be much longer than the shortest paths. Fig.12 also shows that the number of enumerations required by heuristics $A^{(0)}$, $A^{(1)}$ and $A^{(2)}$ are much smaller than that required by the upper-bounding technique, although optimality cannot be guaranteed.

Fig.14 compares the performance of the following five algorithms on network I: heuristics B, C, B with backtracking, C with backtracking and the Improved RS algorithm. Heuristic B performs in general poorer than the Improved RS algorithm, but heuristics C, B with backtracking and C with backtracking all perform better than the Improved RS algorithm. For network II, only heuristics B, B with backtracking and the Improved RS algorithms are compared as heuristic C is for networks with uniform weights only. The results are shown in Fig.15.

It is interesting to note that when the number of conference nodes is two, the six algorithms: heuristic $A^{(k)}$, B, C, B with backtracking, C with backtracking and the Improved RS algorithm all reduce to the shortest path algorithm and hence the optimal solution (i.e. the shortest path) can always be found.

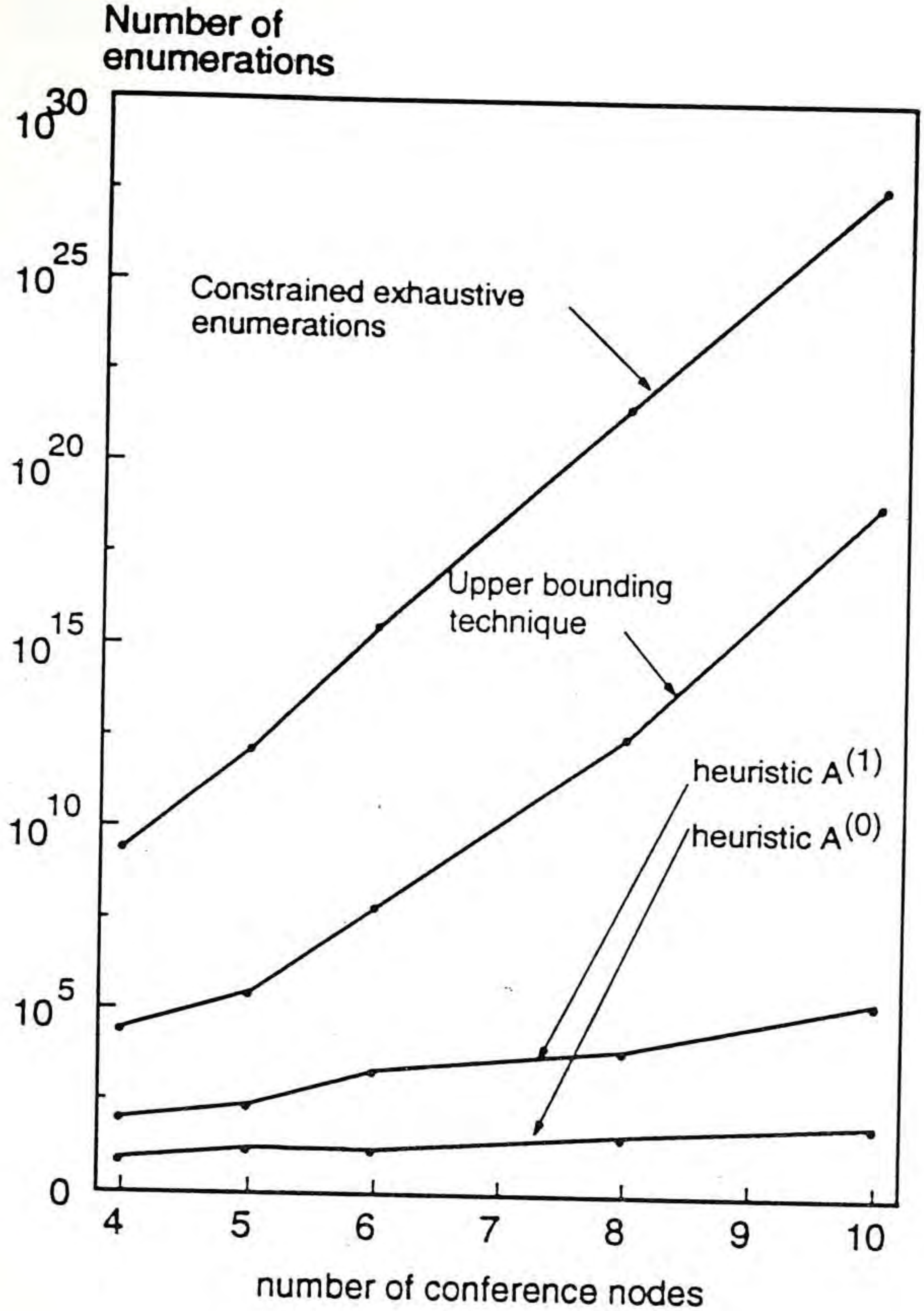


Fig.12(a): Number of required enumerations; Network I

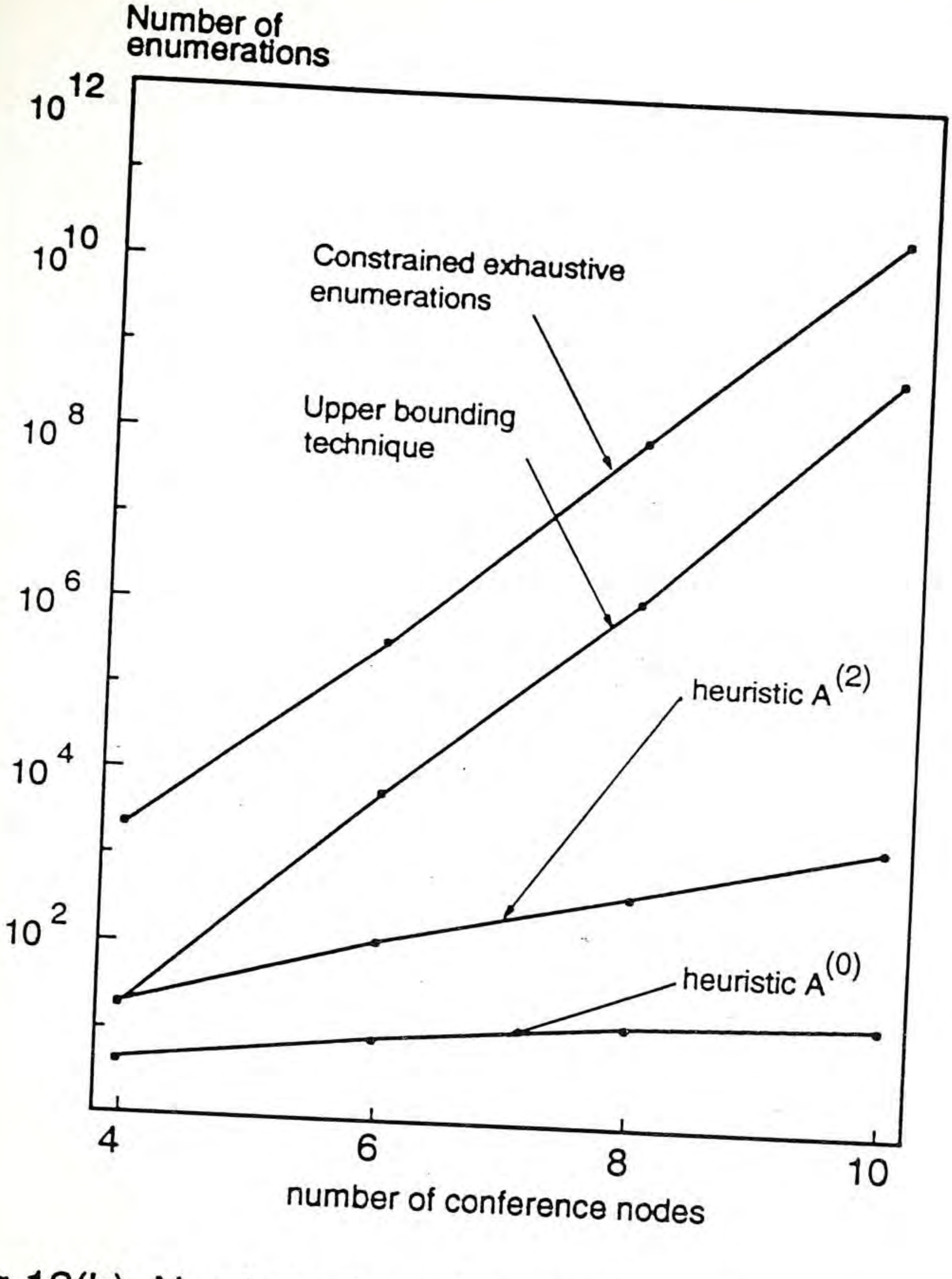


Fig.12(b): Number of required enumerations; Network II

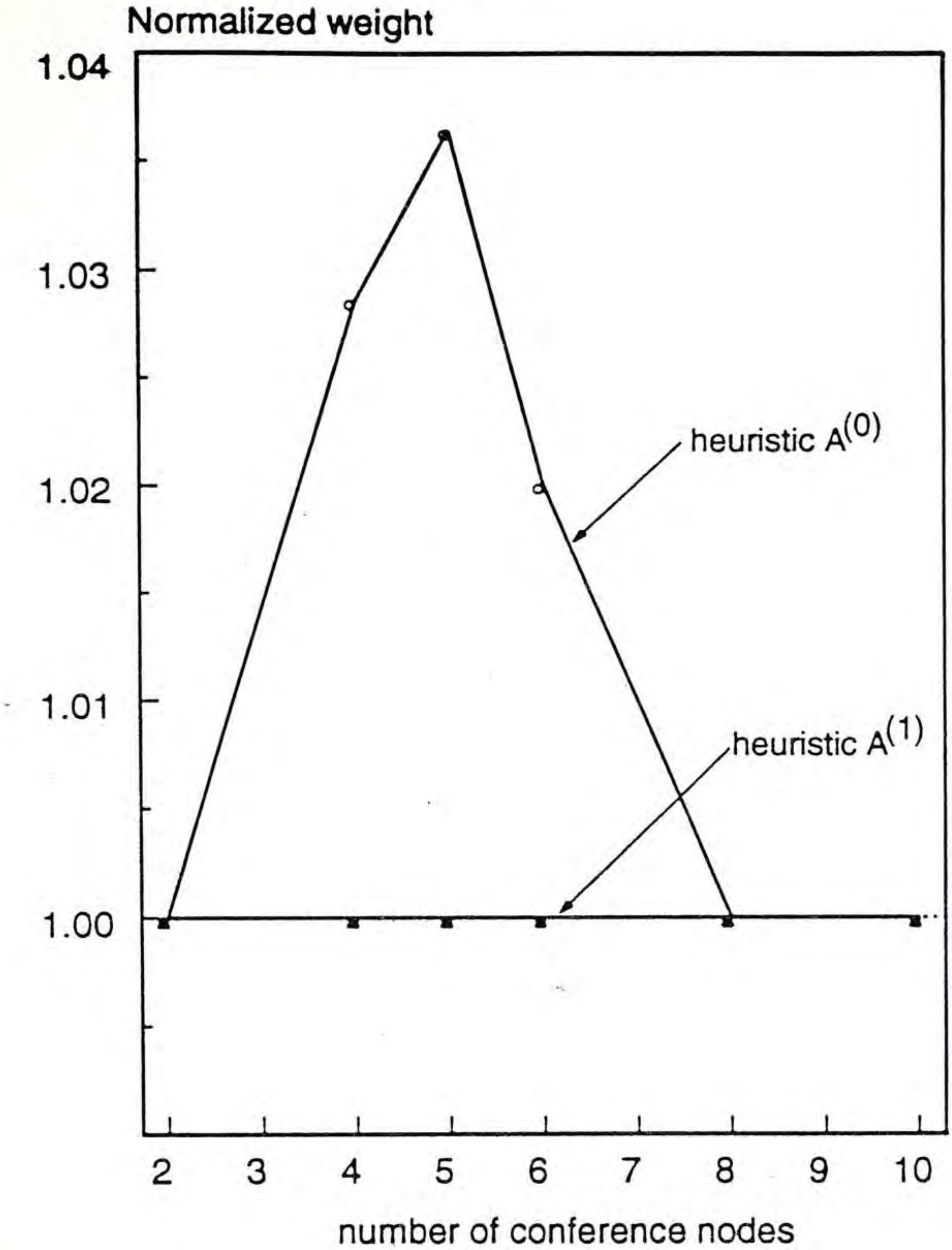


Fig.13(a): Average performance of heuristic $A^{(k)}$;
Network I

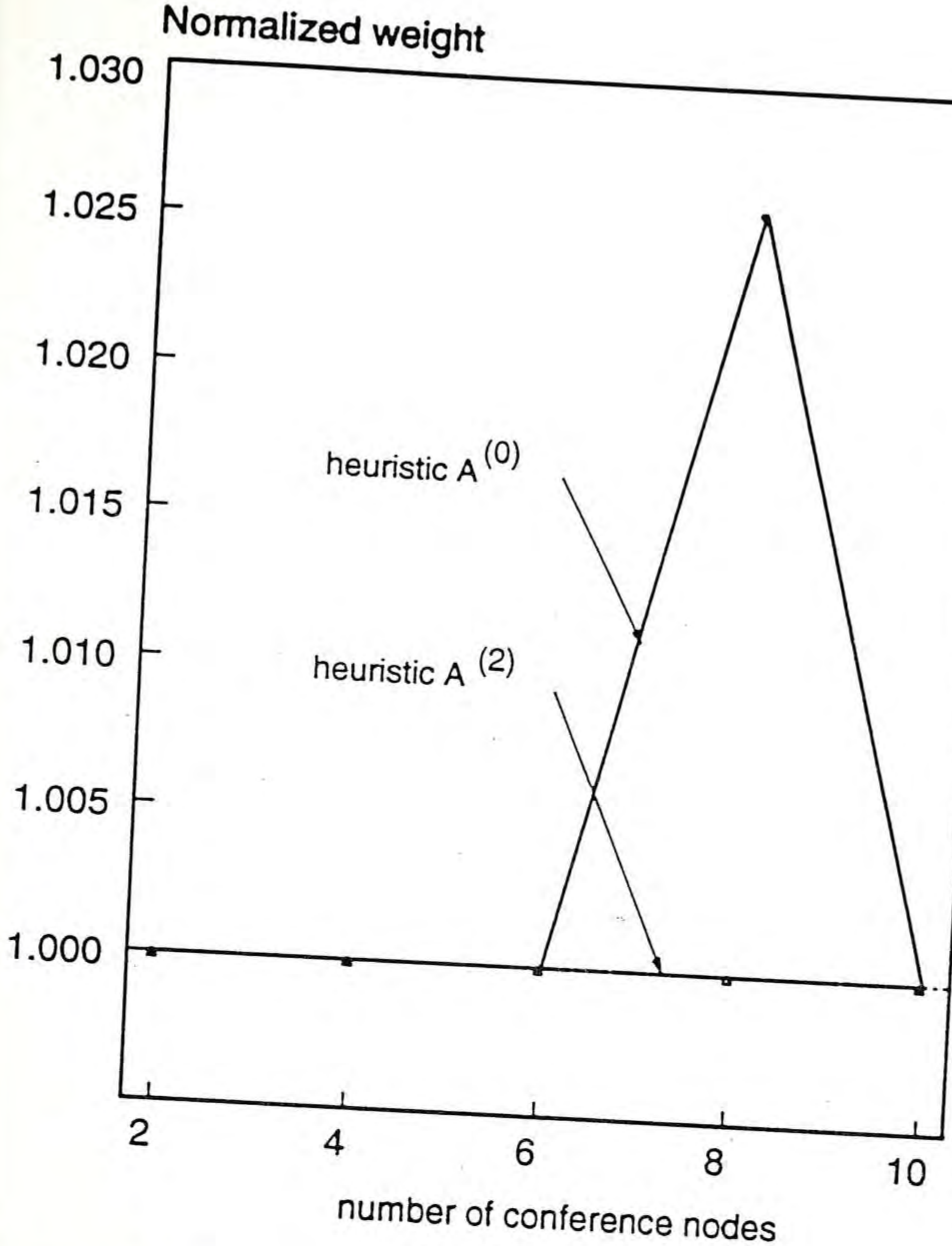


Fig.13(b): Average performance of heuristic $A^{(k)}$;
Network II

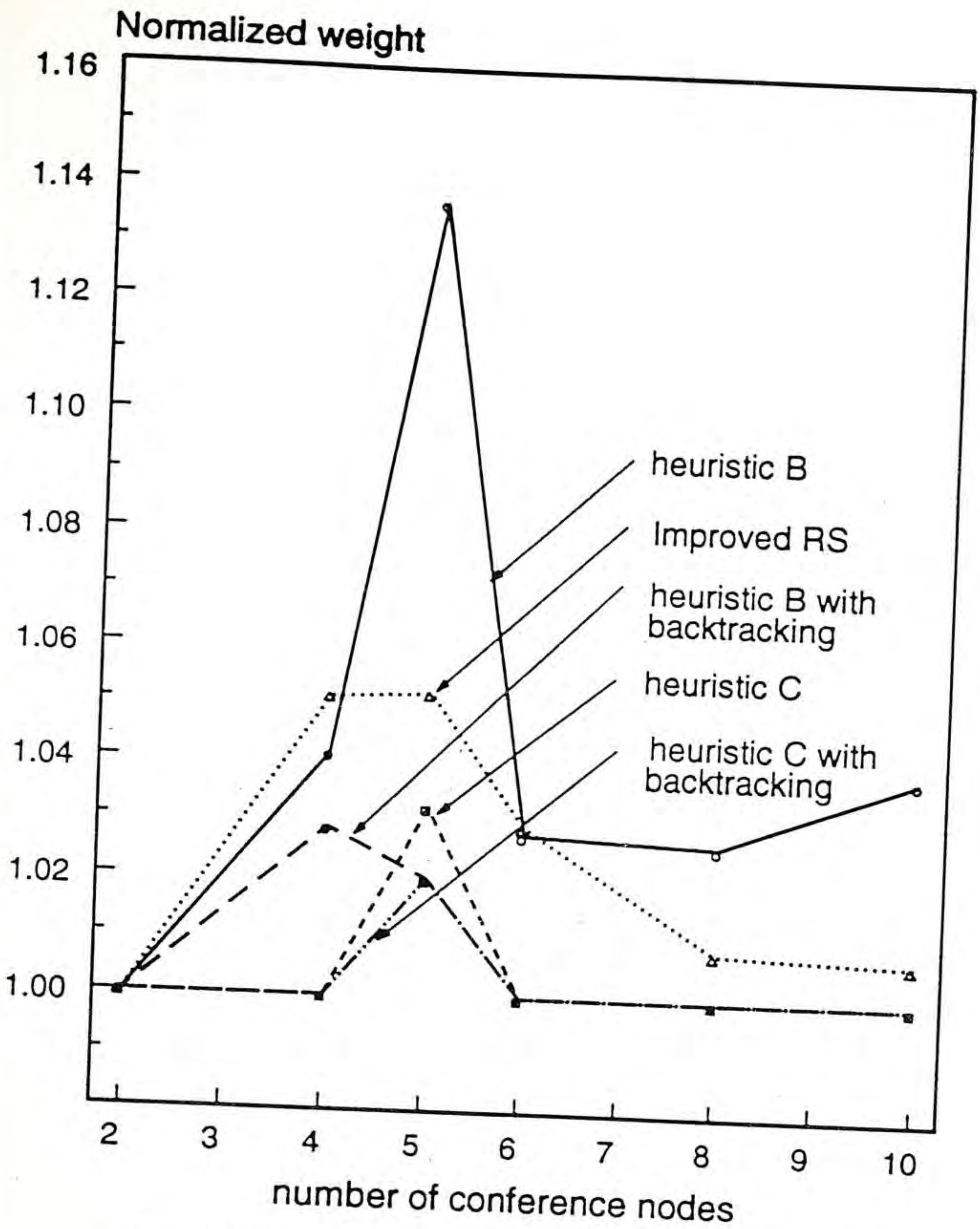


Fig.14: Average performance; Network I

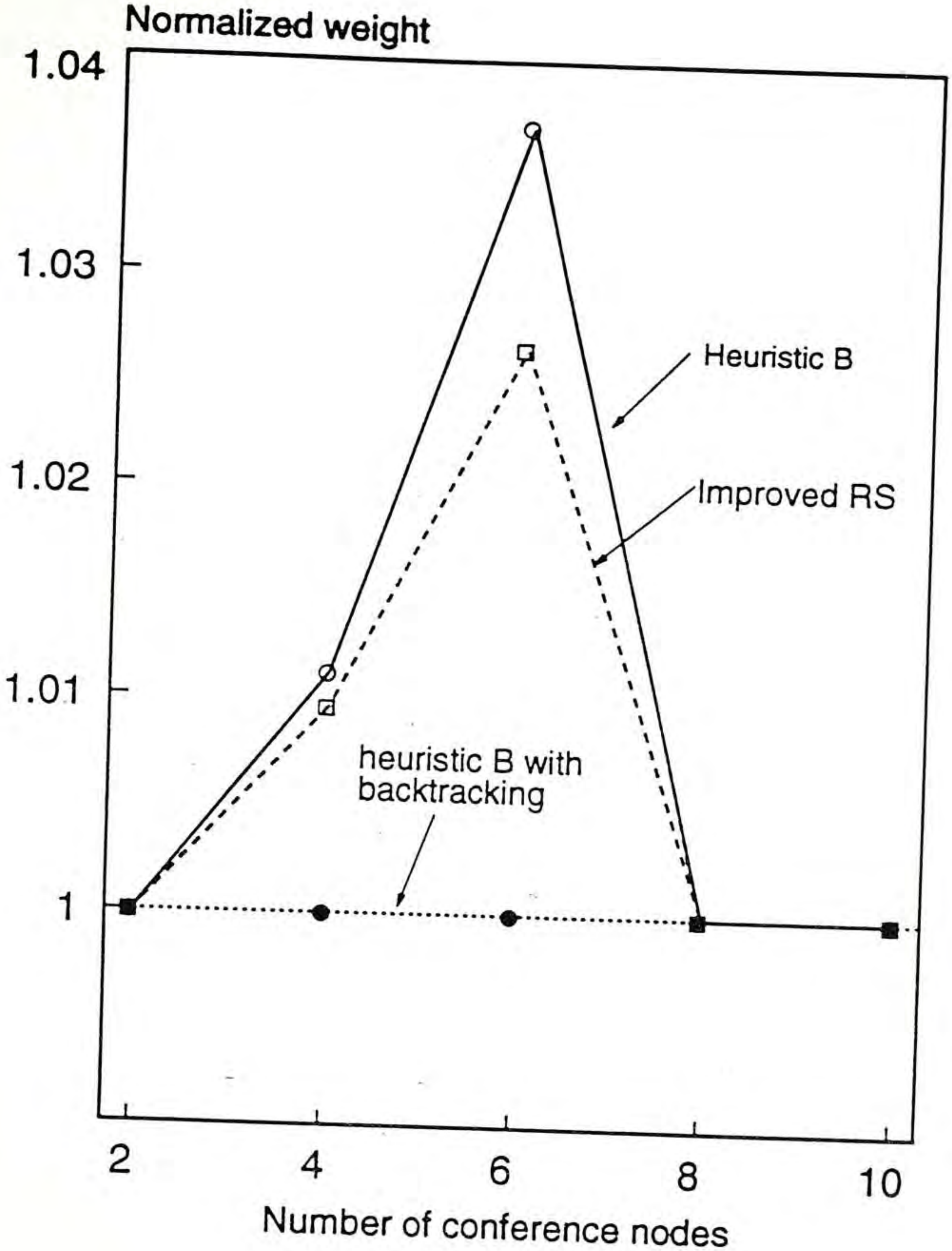


Fig.15: Average performance; Network II

2.5 Chapter Summary

In this chapter, we formulated the multiple destinations routing problem as a zero-one integer programming problem and proposed a method to reduce the number of enumerations required for optimal solutions. When the number of conference nodes is small (say ≤ 5), it is feasible to determine the optimal solutions. Three heuristics were designed for large MDR problems. With the use of the property that a good path to connect the source node and a destination node should not be too much longer than the shortest path connecting them, we designed heuristic $A^{(k)}$. The parameter k allows us to trade-off between optimality and computation time. Heuristic B was modified from the Prim's algorithm for finding minimum spanning tree, taking advantage of the property that two or more paths may share common edges. Heuristic C is for networks with uniform weights and it overcomes the deficiency of heuristic B. Simulation experiments showed that heuristic $A^{(1)}$ and $A^{(2)}$ can already yield optimal solutions for the two network examples. Moreover, heuristic $A^{(k)}$ with small k and heuristic C can give lower cost paths than the Improved RS algorithm.

Chapter 3

Connection Optimization for Two Types of Videoconferences

In this chapter, we formulate the connection optimization problem for two types of multipoint videoconferences. The first type is called selectable media conference. It allows all conferees to receive their own compositions of videos. The second type is called common media conference. Only one composite video is generated for all conferees in this type of conference. Algorithms that use a combination of table look-up and on-line processing are designed for computing the optimal paths to connect the conference sites. The blocking probabilities of these two types of videoconferences in fully connected networks are derived and compared. The sensitivity of network throughput to conference size distribution is also studied. It is found that, for a given mean conference size, the variance of the conference size distribution has a small but non-negligible effect on the network throughput.

3.1 Background

The effectiveness of videoconference services depends on how much it can emulate a face-to-face meeting. Audio should be unconstraintеd because it occupies relatively small transmission bandwidth. On the other hand, transmission of good quality compressed video requires much larger bandwidth. The finding of good connection paths for a conference therefore is an important problem in the provision of videoconference services. In the following sections, we classify two types of videoconferences, namely those with selectable media and those with common media, and propose algorithms for determining the optimal connection paths for both types. The blocking probabilities of these two types of videoconferences in fully connected networks are also derived.

3.2 The Videoconference Services

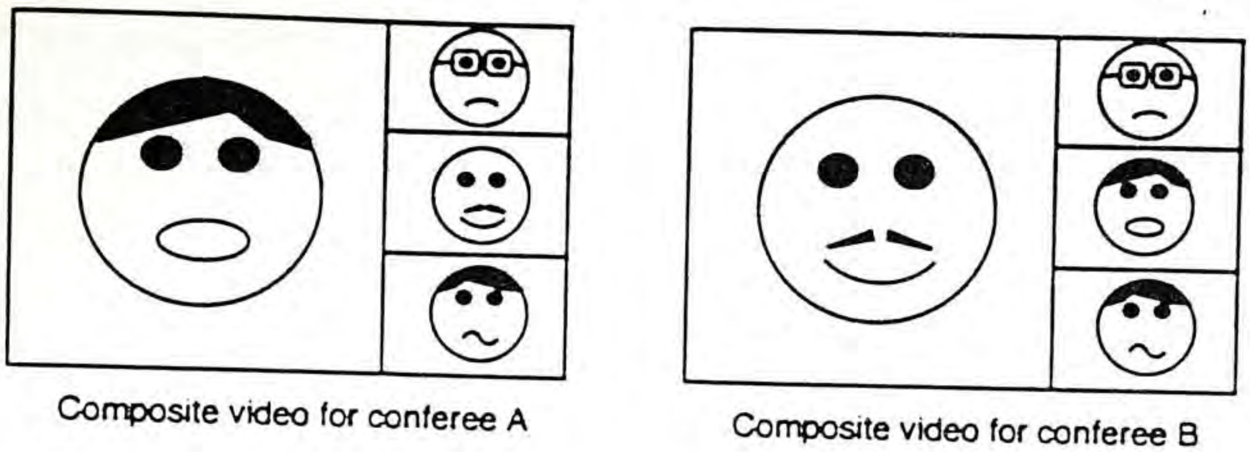
A. The Services

Videoconference calls can be classified into *prescheduled calls* and *on-demand calls*. Prescheduled calls require users to make reservations in advance. Access conflicts can be resolved in advance and efficient scheduling of channel usage can be done. On-demand calls, on the other hand, require immediate seizure of the necessary communication resources in the same way telephone calls do. The determination of the connection paths must also be done in real-time.

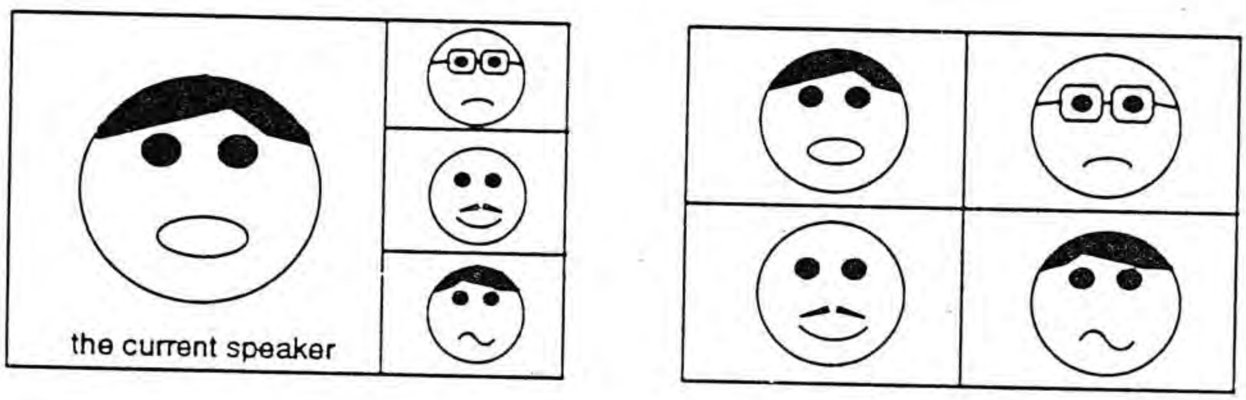
There are various ways to present the video images of the conferees at different locations (e.g. see section 1.4 in chapter 1). Using a single monitor with segmented windows is a promising approach [ROMA 87, BODS 89] because each conference site needs only one monitor (i.e. less studio cost) and one logical channel (i.e. less transmission cost). In this chapter, we focus on this approach and study the connection optimization problem for two types of conferences, the *selectable media conferences* and the *common media conferences*. For a selectable media conference, each conferee can choose to receive a particular composition of video. Fig.1(a) shows the two video compositions sent to two conferees. For a common media conference, all conferees receive the same composite video. Fig.1(b) shows two examples of common media composite video.

B. The Networks

Let the videoconference service be provided by a wide area packet switched network. Let there be N nodes in the network and let a subset of these nodes be called *processing nodes*. Each processing node is equipped with a number of *conference bridges*. It should be noted that the more the number of processing nodes in the network, the more flexible is the assignment of connection paths. Thus there is an obvious tradeoff between the number of processing nodes



(a) Example of selectable media



(b) Examples of common media

Fig. 1: Examples of selectable media and common media conferences

and the optimality of the resulting connection paths. Fig. 2 shows the internal structure of a processing node. Each conference requires a conference bridge to collect the video and voice packets from the conference sites, to mix the audio signals, to synthesize the composite video signals and to packetize the resulting video and audio signals for distribution to the conference sites.

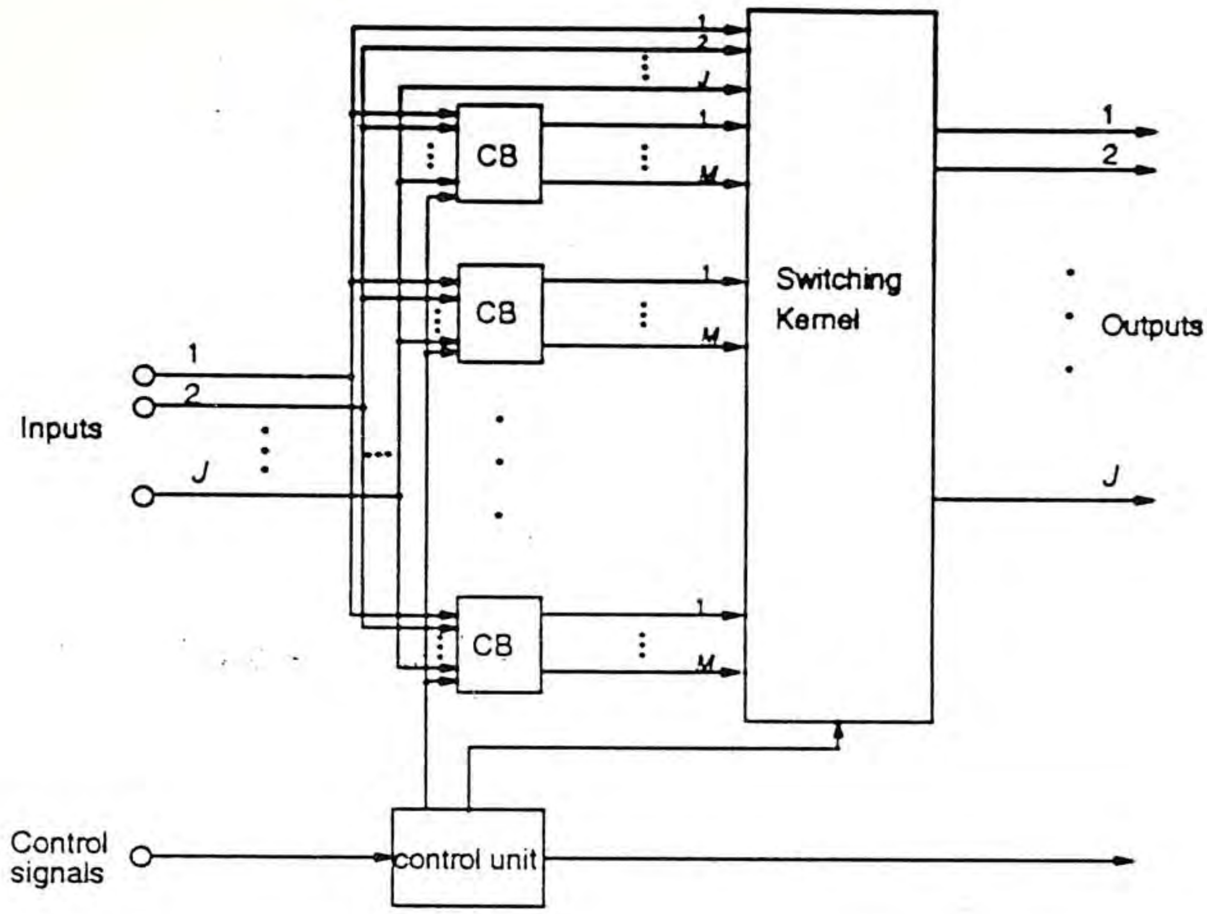
The videos from all conference sites are compressed, packetized and statistically multiplexed onto the network links. Each network link can be characterized in terms of the number of logical video channels by known queueing techniques [MAGL 88, SEN 89], which need as inputs the calibrated video quality in terms of packet loss rate and delay statistics.

3.3 Connection Optimization

A. For Selectable Media Conferences

Each conference site needs to forward its video/voice packets to the conference bridge via a connection path. Let the set of all connection paths from the conference sites to the conference bridge be called an inbound connection. Conferees also need to receive videos through individual connection paths fanning out from the conference bridge. Let this set of paths be called the outbound connection.

For selectable media conferences, the optimal inbound and outbound connections, i.e. those with the minimum number of channels, have the same topology. Let V be the set of nodes in the network, $\mathfrak{I}$ be the set of processing nodes and $|\mathfrak{I}|$ be the number of elements in $\mathfrak{I}$. For a particular conference, let the network nodes with one or more conference sites connected be called *conference nodes*. Note that the conference bridges are only located in the processing nodes and the optimal conference bridge may not be located at a conference node. For example, if all the nodes of the network shown in Fig.3 are processing nodes, the optimal conference bridge is located at node 1, which is not a conference node. Therefore, we need to enumerate



CB: Conference Bridge

Fig. 2: Internal structure of a processing node

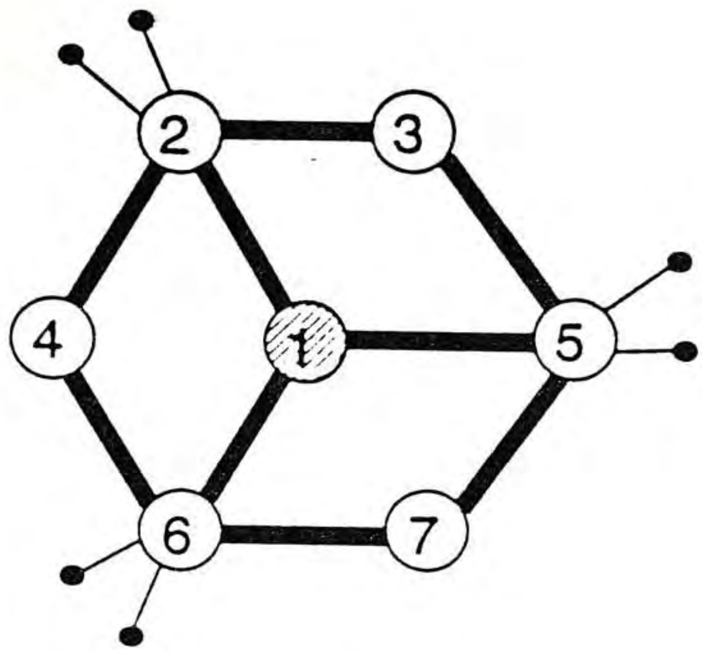


Fig.3: The optimal conference bridge is located at node 1

all the nodes in $\Im$ to determine the optimal conference bridge location. Given locations of the conference sites, the subroutine *CB(i)* gives the sum of the shortest path lengths connecting node i and the conference sites. The algorithm for determining the optimal connection paths for selectable media conferences or the SM algorithm is given below:

<u>SM Algorithm</u>

```
[Input:   locations of the conference sites           ]
[Output:  the optimal conference bridge location C    ]
min ← ∞;
FOR i=1 TO N DO
    IF i ∈ ℑ THEN
        x ← CB(i);
        IF x<min THEN
            min ← x;
            C ← i;
        ENDIF;
    ENDIF;
ENDFOR.
```

After the optimal conference bridge location is determined, the optimal connection paths can then be constructed by connecting the conference sites to node C by the shortest paths.

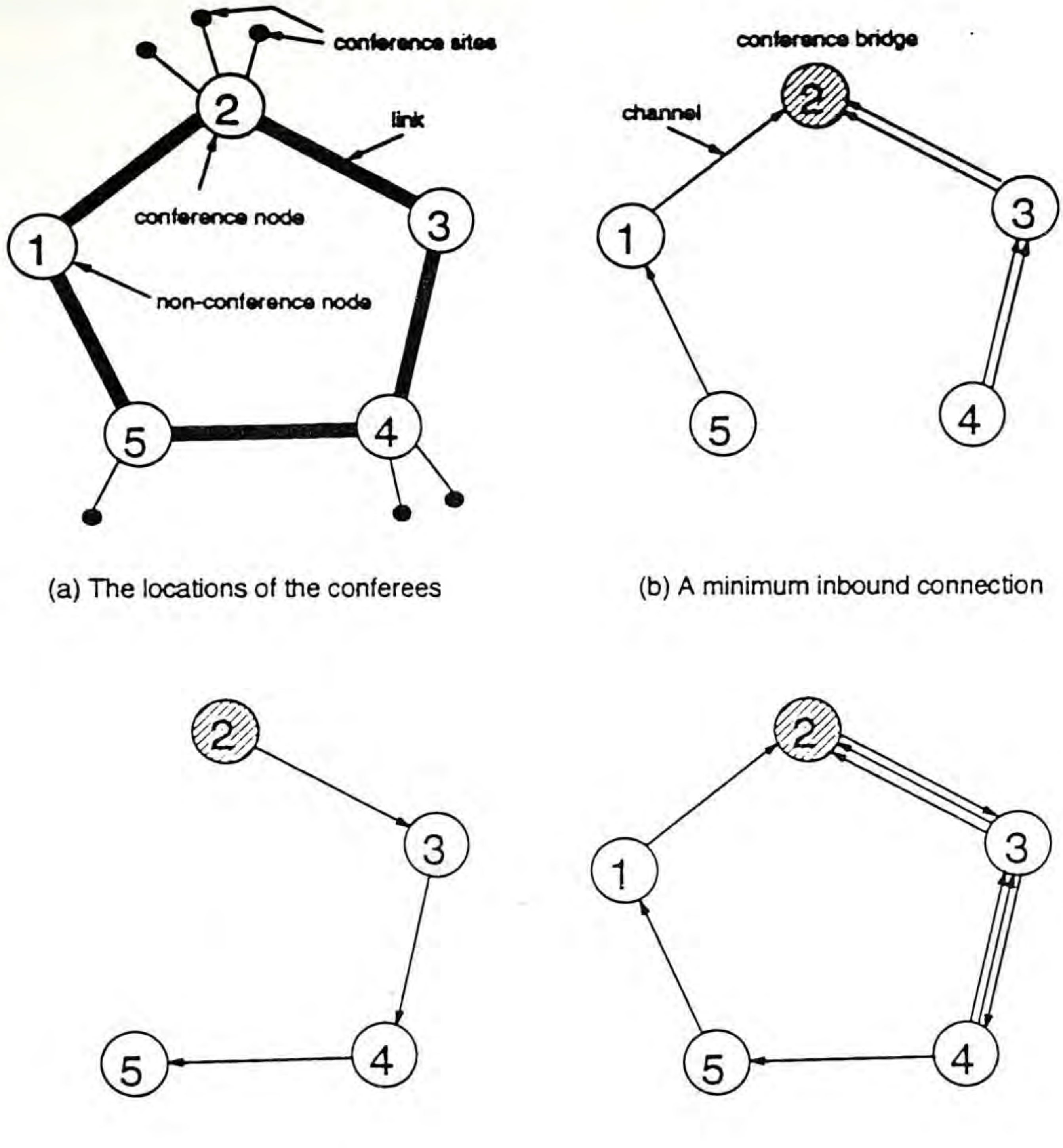
Subroutine *CB(i)* is executed a total of $|\Im|$ times. In each execution a fixed number (equal to the number of conference nodes in the conference concerned) of shortest paths need to be found. If Dijkstra's shortest path algorithm of time complexity $O(N^2)$ [AHO 83] is used, the time complexity of the SM algorithm is $O(N^3)$ as $|\Im| \leq N$.

For on-demand calls, a fast determination of the optimal connection path at call set-up time is required. For a network with a large number of possible conference sites, it would not be feasible to store in tables all possible connection patterns. A combination of table look-up and on-line processing to reduce the computation time should therefore be used. Note that in the SM algorithm, the subroutine *CB(i)* finds the shortest path lengths between node i and all other conference nodes. These shortest paths can be computed off-line and stored in tables. For a network with N nodes, there are $(N^2 - N)/2$ shortest paths between all node pairs. With these tables available, the SM algorithm has a time complexity of $O(N)$.

B. For Common Media Conferences

For a common media conference, all conferees receive the same composite video. The inbound connection, through which the conferees forward video/voice packets to the conference bridge, is the same as that for selectable media conferences. The outbound connection, on the other hand, is a multicast tree (chapter 2) through which the conference bridge multicasts the processed packets to the conferees. Fig.4 shows an example of inbound and outbound connections for a common media conference when all nodes of the network are processing nodes.

The optimal conference bridge location must be found before determining the optimal connection paths. Note that a conference bridge need not be located at a conference node. For example, if all the nodes of the network shown in Fig.3 are processing nodes, the optimal conference bridge is located at node 1, which is not a conference node. Therefore, we need to check all processing nodes to determine the optimal location of the conference bridge. The search can be performed as follows. Let T^* be the minimum multicast tree connecting a given set of conference nodes D. As the topology of T^* is invariant when any node in $T^* \cap \Im$ assumes the role of conference bridge, the determination of the optimal conference bridge location can be divided into two stages. First, determine the multicast tree T^*, select the best conference bridge location in $T^* \cap \Im$ and compute the number of channels in the resulting inbound and outbound connections. Second, enumerate all processing nodes in $\Im - T^*$ (i.e. the processing nodes that have not been checked in stage 1) for conference bridge locations. In each enumeration, determine the multicast tree connecting the conference nodes and the conference



(a) The locations of the conferees

(b) A minimum inbound connection

(c) A minimum outbound connection

(d) The optimal connection path

Fig.4: Inbound and outbound connections for a common media conference

bridge and compute the number of channels in the resulting inbound and outbound connections.

With these two stages completed, the optimal conference bridge location is determined for the construction of the optimal connection path. Let the subroutine *Steiner(D)* find a minimum multicast tree connecting the given set of conference nodes D and output the set of multicast tree nodes T_{node} and edges T_{edge}. The algorithm summarizing the above procedures is called the CM algorithm and is given below:

<u>CM Algorithm</u>

Input: locations of the conference sites
Outputs: the optimal conference bridge location C,
sets of nodes T_{node} and edges T_{edge} in the outbound connection

compute D;
call *Steiner(D)*;
$min \leftarrow \infty$;
FOR i=1 TO N DO
    IF node $i \in \Im \cap T_{node}$ THEN
        IF $CB(i)<min$ THEN
            $min \leftarrow CB(i)$;
            $P^{*} \leftarrow i$;
        ENDIF;
    ENDIF;
ENDFOR;
$min \leftarrow |T_{edge}| + min$;
$T_{min} \leftarrow (T_{node}, T_{edge})$;
$C \leftarrow P^{*}$;
FOR i=1 TO N DO
    IF node $i \in \Im - T_{node}$ THEN
        $D \leftarrow D \cup \{\text{node } i\}$;
        call *Steiner(D)*;
        $x \leftarrow |T_{edge}| + CB(i)$;
        IF $x<min$ THEN
            $\min \leftarrow x$;
            $T_{min} \leftarrow (T_{node}, T_{edge})$;
            $C \leftarrow i$;
        ENDIF;

```
        D←D −{node i};
    ENDIF;
ENDFOR.
```

As the determination of minimum multicast tree is NP-complete [GARE 79], finding optimal solution is feasible only when the number of conference nodes is small. In this case, the optimal algorithm proposed in chapter 2 can be used for *Steiner(D)*. When the number of conference nodes is large, a good heuristic should be used. In particular, a fast heuristic should be used for on-demand conferences but a more elaborate heuristic can be used for prescheduled conferences. After the CM algorithm has been executed, the inbound connection can be constructed by connecting the conference sites to node C by the shortest paths and the outbound connection is given by the sets T_{node} and T_{edge}.

Subroutine *CB(i)* is executed $|T_{node}|$ times in the first stage and $|\Im - T_{node}|$ times in the second stage. Since $|T_{node}| \leq N$ and $|\Im - T_{node}| \leq N$, the total execution time involving *CB(i)* is $O(N^3)$. Subroutine *Steiner(D)* is executed once in the first stage and $|\Im - T_{node}|$ times in the second stage. If Prim's minimum spanning tree heuristic of time complexity $O(N^2)$ [AHO 83] is used for *Steiner(D)*, the total execution time involving *Steiner(D)* is $O(N^3)$. Combining, the time complexity of the CM algorithm is found to be $O(N^3)$.

As mentioned before, for on-demand calls, a combination of table look-up and on-line processing can be used. Recall from before that subroutine *CB(i)* gives the sum of the shortest path lengths connecting node i and the conference sites. There are $(N^2 - N)/2$ shortest paths between all node pairs and they can easily be computed and stored in tables. Subroutine *Steiner(D)* gives a multicast tree connecting all the involved conference nodes and the processing node. Let R be the maximum number of involved conference nodes in any conference. Then ulting number of multicast trees is $\sum_{i=2}^{R+1}\binom{N}{i}$. Table 1 shows this number for different R

For most practical cases, R should not be very large, say 5 or less. Therefore computing ring all multicast trees should not be a problem provided N is no larger than say, 30. If multicast trees for connecting a small number of nodes are stored, larger size multicast an be obtained from fast heuristic such as heuristics B or C proposed in chapter 2.

R	Number of multicast trees			
	N=20	N=30	N=40	N=50
3	6.175×10^3	3.190×10^4	1.021×10^5	2.511×10^5
4	2.168×10^4	1.744×10^5	7.601×10^5	2.370×10^6
5	6.044×10^4	7.682×10^5	4.598×10^6	1.826×10^7
6	1.380×10^5	2.804×10^6	2.324×10^7	1.181×10^8
7	2.639×10^5	8.657×10^6	1.001×10^8	6.550×10^8
8	4.319×10^5	2.296×10^7	3.736×10^8	3.160×10^9
9	6.166×10^5	5.301×10^7	1.221×10^9	1.343×10^{10}
10	7.846×10^5	1.076×10^8	3.533×10^9	5.079×10^{10}

Table 1: Number of multicast trees

It is interesting to note that the solutions given by Heuristic A in chapter 2 can be prosively improved by increasing the value of the parameter k. Hence, using this heuristic and g off-line computation, the optimality of the multicast trees stored in memory can be ressively improved.

Performance Analysis

The analysis of multipoint videoconference service in a general network appears to be very difficult and no known effective analytical technique is available. In this section, we restrict our attention to fully connected networks where all nodes are processing nodes. In the analysis the usual Poisson arrival of calls and the exponentially distributed holding time assumptions are made.

For a selectable media conference, it can be proved that the optimal conference bridge is located at the node with the largest number of conferees attached. For a common media conference, the minimum multicast tree is constructed by connecting the conference nodes directly and hence the optimal conference bridge is also located at the conference node with the largest number of conferees. We denote the strategy which places the conference bridge at its optimal location the *optimal strategy*. For comparison, we consider also an *alternate strategy* in which the conference bridge is located at the call initiating node.

We model the conference calls as customers, the network links as facilities, and the channels in each link as servers. Since the number and locations of conferees in each conference is random, the arrival of a conference call would mean a customer requesting a simultaneous possession of a random number of servers from several facilities. Kelly [KELL 86] investigated the blocking probability in circuit switched networks where each customer requests a fixed number of channels from a set of links. By decomposing the conference traffic (as explained in the next subsection), Kelly's result can be applied to allocate a random number of servers. But Kelly's model requires the enumerations of all possible paths in the networks, which is upper bounded by 2^x where x is the total number of links in the network. Therefore, for any non-trivial network this approach cannot lead to numerical results. We observe that a properly designed videoconference service should have a low blocking probability, say no more than 10^{-3} [LIAO 87]. Under this condition, the link occupancies can be approximated to be independent [WHIT 85] and the maximum number of state variables can be reduced from a set of 2^x variables to x independent sets of (M-1) variables, where M is the maximum allowable number of conferees in a conference. The following is a derivation of the blocking probabilities for the reduced state space model.

In the network, let $b_1, b_2, \cdots$ be the total number of conference subscribers at node 1, node 2, $\cdots$, and let the b_i's be sufficiently large so that the arrivals of conference calls can be well approximated by a Poisson process just like that in the analysis of a telephone system. When the channel facilities are not all available for a conference call, this call is blocked and does not return. Define conference size $\tilde{W}$ as the total number of conferees in a conference and let the distribution of $\tilde{W}$ be denoted as

$$w_i = P[\tilde{W} = i] \qquad i = 2, 3, \cdots, M$$

and be known. Knowledge of this distribution is necessary for a complete specification of input traffic.

A. Analysis of The Alternate Strategy for Selectable Media Conferences

A-1. Decomposition of Arrival Traffic

The conferences initiated at a particular node, say node p, may involve conferees at the other nodes and thus may demand channels on the links connected to node p. Let λ_p be the arrival rate of conferences initiated at node p, $Poisson[\lambda_p]$ denotes a Poisson process with rate λ_p, and let $\beta_q(i)$ be the probability that a conference initiated at node p involves i conferees at node q. We decompose $Poisson[\lambda_p]$ into M-1 processes $Poisson[\lambda_p\beta_q(1)]$, $Poisson[\lambda_p\beta_q(2)]$, ..., $Poisson[\lambda_p\beta_q(M-1)]$ [KOBA 78], where each arrival of $Poisson[\lambda_p\beta_q(i)]$ requests i channels on link pq. This decomposition is illustrated in Fig. 5.

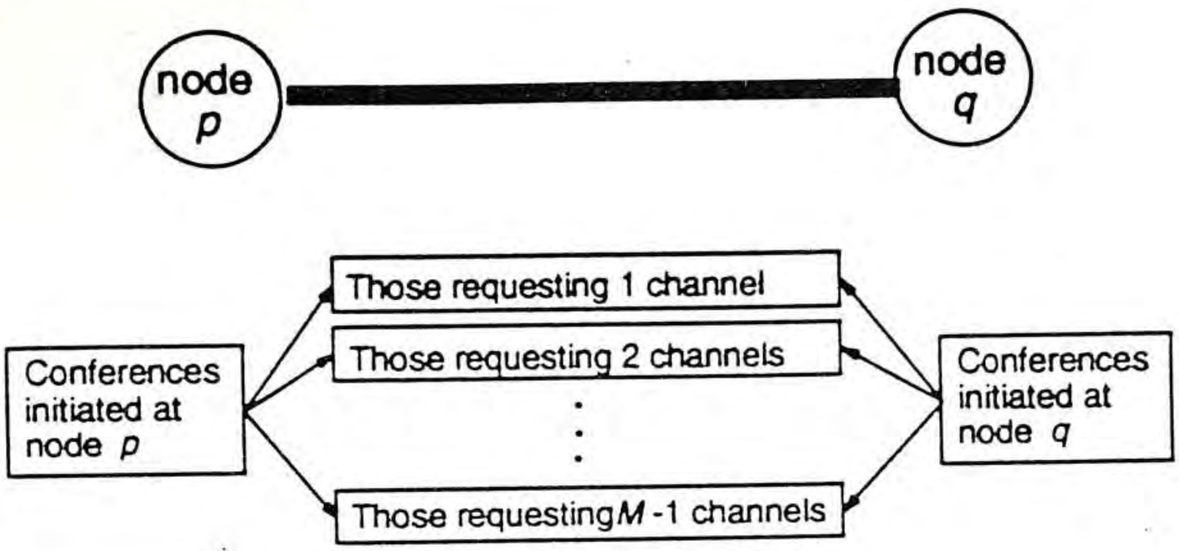


Fig.5: Decomposition of conference traffic

Since a conference may be initiated at node p or node q, both $Poisson[\lambda_p]$ and $Poisson[\lambda_q]$ would load traffic on link pq (Fig. 5). As arrivals of both $Poisson[\lambda_p\beta_q(i)]$ and $Poisson[\lambda_q\beta_p(i)]$ demand i channels on link pq, we can aggregate [KOBA 78] these two processes to form $Poisson[\lambda_{pq}(i)]$ where $\lambda_{pq}(i) = \lambda_p\beta_q(i) + \lambda_q\beta_p(i)$.

A-2. Derivation of $\beta_q(i)$

Let all conferees have equal community interest on all the others. Then the probability γ_q that a conferee is located at node q is:

$$\gamma_q = \frac{b_q}{\sum_{i=1}^{N} b_i}$$

The probability $\beta_q(i)$ is given by

$$\beta_q(i) = \begin{cases} \sum_{s=i+1}^{M} \binom{s-1}{i} \gamma_q^i (1-\gamma_q)^{s-1-i} w_s & i = 1,2,\cdots,M-1 \\ \sum_{s=2}^{M} (1-\gamma)^{s-1} w_s & i = 0 \end{cases}$$

where M is the maximum allowable number of conferees in a conference.

A-3. Link Occupancy Distribution

The channel occupancy on link pq can be described by the state vector $\tilde{n}_{pq} \equiv (n_1, n_2, \ldots, n_{M-1})$ where n_i is the number of ongoing conferences on link pq occupying i channels each. The dependency of n_i on p and q is dropped for simplicity. Let $1/\mu$ be the mean conference duration and L_{pq} be the number of logical full-duplex channels on link pq. Since the total number of occupied channels on link pq cannot be larger than L_{pq}, the set Λ_{pq} of admissible states on link pq is

$$\Lambda_{pq} = \left\{ \tilde{n}_{pq} \mid \sum_{j=1}^{M-1} j n_j \le L_{pq} \right\}$$

The state probability $P[\tilde{n}_{pq}]$ is given by a product form solution [KELL 86]

$$P[\tilde{n}_{pq}] = \begin{cases} C_{pq} \dfrac{\rho_1^{n_1}}{n_1!} \dfrac{\rho_2^{n_2}}{n_2!} \cdots \dfrac{\rho_{M-1}^{n_{M-1}}}{n_{M-1}!} & \tilde{n}_{pq} \in \Lambda_{pq} \\ 0 & \text{otherwise} \end{cases}$$

where ρ_i is defined as $\lambda_{pq}(i)/\mu$ (the dependency of ρ_i on p and q is also omitted here for simplicity)

and C_{pq} is a normalization constant given by

$$C_{pq}^{-1} = \sum_{k=0}^{L_{pq}} G_{pq}(k)$$

where

$$G_{pq}(k) = \sum_{\sum_{j=1}^{M-1} j n_j = k} \frac{\rho_1^{n_1}}{n_1!} \frac{\rho_2^{n_2}}{n_2!} \cdots \frac{\rho_{M-1}^{n_{M-1}}}{n_{M-1}!}$$

Let $\tilde{N}_{pq}$ be the channel occupancy on link pq. The distribution of $\tilde{N}_{pq}$ can be readily found as

$$P[\tilde{N}_{pq} = k] = \sum_{\sum_j j n_j = k} P[\tilde{n}_{pq}] = C_{pq} G_{pq}(k) \qquad (1)$$

For the special case where there are only two conferees in all conferences, equation (1) reduces to the Erlang B formula.

A-4. Blocking Probability

For a particular conference, let $\tilde{K} \equiv (\tilde{K}_1, \tilde{K}_2, \cdots, \tilde{K}_N)$ where $\tilde{K}_i$ is a random variable denoting the number of conferees located at node i. In addition, let $\mathbf{k} \equiv (k_1, k_2, \cdots, k_N)$ where the k_i's are non-negative integers. Define event Ψ_{pq} as

Ψ_{pq} = {the number of channels required by an incoming call on link pq is within what is available}

$= \{\tilde{K}_q \leq (L_{pq} - \tilde{N}_{pq})\}$

A conference call is blocked when at least one of the involved links does not have sufficient free channels. Given that a conference initiated at node p has s conferees, the blocking probability $B_p(s)$ can be found as

$$B_p(s) = 1 - P\left[\bigcap_{\substack{q=1\\q\neq p}}^{N} \Psi_{pq}\right]$$

$$= 1 - \sum_{\Omega_1} \prod_{\substack{q=1\\q\neq p}}^{N} P[\tilde{N}_{pq} \leq L_{pq} - k_q] P\left[\tilde{K} = k \mid \sum_{j=1}^{N} k_j = s \text{ and } k_p \geq 1\right]$$

where

$$\Omega_1 = \left\{k \mid \sum_{j=1}^{N} k_j = s \text{ and } k_p \geq 1\right\}$$

$$P\left[\tilde{K} = k \mid \sum_{j=1}^{N} k_j = s \text{ and } k_p \geq 1\right] = \binom{s-1}{k_1, k_2, \cdots, k_p - 1, \cdots, k_N} \gamma_p^{k_p - 1} \prod_{\substack{i=1\\i\neq p}}^{N} \gamma_i^{k_i}$$

Removing the conditionings on s and p, the blocking probability B is obtained as

$$B = \frac{\sum_{p=1}^{N} \lambda_p \left[\sum_{s=2}^{M} B_p(s) w_s\right]}{\sum_{j=1}^{N} \lambda_j}$$

B. Analysis of The Optimal Strategy for Selectable Media Conferences

Let Θ_p denote the event that node p is chosen as the conference bridge and let $\chi^i_{pq}(j)$ be a composite event defined as

$\chi^i_{pq}(j) = \{$node p and (i-1) other nodes all have the same largest number of conferees attached; Θ_p; and node q has j conferees attached$\}$

$P[\chi^1_{pq}(j)]$ can be expressed in terms of the joint distribution of K_i's as

$$P[\chi^1_{pq}(j)] = \sum_{\Omega_2} P[\tilde{K} = k]$$

where

$$\Omega_2 = \{k \mid (2 \le k_1 + k_2 + \cdots + k_N \le M) \text{ and } (k_p > k_i \text{ for } i \ne p) \text{ and } (k_q = j)\}$$

Similarly, $P[\chi^2_{pq}(j)]$ is given by

$$P[\chi^2_{pq}(j)] = \frac{1}{2} \sum_{\substack{u=1 \\ u \ne p}}^{N} \sum_{\Omega_3} P[\tilde{K} = k]$$

where

$$\Omega_3 = \{k \mid (2 \le k_1 + k_2 + \cdots + k_N \le M) \text{ and } (k_u = k_p \text{ and } k_i < k_p \text{ for } i \ne u \ne p) \text{ and } (k_q = j)\}$$

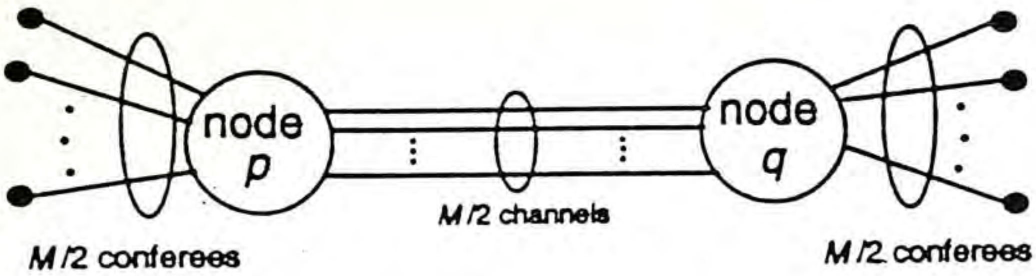
$P[\chi^3_{pq}(j)], P[\chi^4_{pq}(j)], \cdots$ can be found in a similar fashion. Finally, $P[\chi^N_{pq}(j)]$ is given by

$$P[\chi^N_{pq}(j)] = \frac{1}{N} P[\tilde{K} = (j, j, \cdots, j)]$$

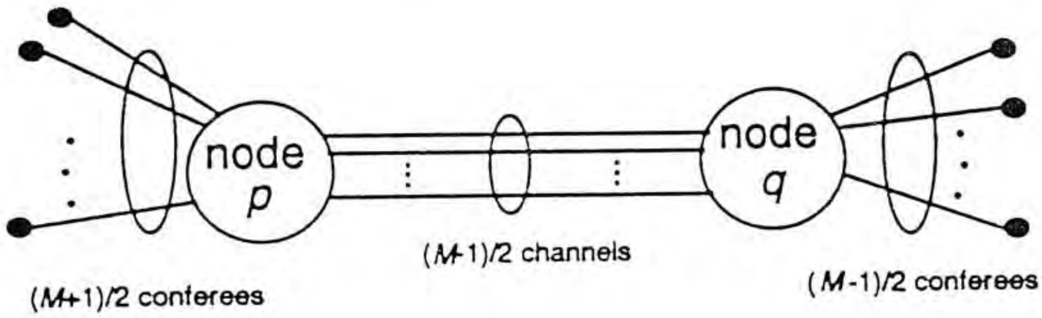
The probability $\sigma_{pq}(j)$ that a conference demands j channels on link pq is

$$\sigma_{pq}(j) = \sum_{i=1}^{N} \{P[\chi^i_{pq}(j)] + P[\chi^i_{qp}(j)]\} \tag{2}$$

Fig. 6 shows that, under the optimal strategy, the maximum number of channels required in any link is $\left\lfloor \frac{M}{2} \right\rfloor$. Since the total rate of conference arrivals λ_T is the sum of the rates at individual nodes and is given by $\lambda_T = \sum_{i=1}^{N} \lambda_i$, the traffic loaded onto link pq by $Poisson[\lambda_T]$ can therefore



(a) Even M



(b) Odd M

Fig.6: Maximum channel requirements, optimal strategy

be decomposed into $Poisson[\lambda_T\sigma_{pq}(1)], Poisson[\lambda_T\sigma_{pq}(2)], \cdots, Poisson\left[\lambda_T\sigma_{pq}\left(\left\lfloor \frac{M}{2} \right\rfloor\right)\right]$, where each arrival of $Poisson[\lambda_T\sigma_{pq}(i)]$ demands i channels on link pq. The state probabilities and link occupancies can be found in exactly the same manner as that for the alternate strategy in the last subsection. Given that a conference has s conferees, the blocking probability $B(s)$ is given by

$$B(s) = 1 - \sum_{\Omega_4} \sum_{p=1}^{N} \prod_{\substack{q=1 \\ q \neq p}}^{N} P[\tilde{N}_{pq} \leq L_{pq} - \tilde{K}_q \mid \tilde{K}_q = k_q] P\left[\tilde{K} = k \text{ and } \Theta_p \mid \sum_{n=1}^{N} k_n = s\right]$$

where

$$\Omega_4 = \left\{k \mid \sum_{n=1}^{N} k_n = s\right\}$$

$$P\left[\tilde{K} = k \text{ and } \Theta_p \mid \sum_{n=1}^{N} k_n = s\right]$$

$$= P\left[\Theta_p \mid \sum_{n=1}^{N} k_n = s \text{ and } \tilde{K} = k\right] P\left[\tilde{K} = k \mid \sum_{n=1}^{N} k_n = s\right]$$

$$= \begin{cases} \dfrac{1}{j} \dbinom{s}{k_1, k_2, \cdots, k_N} \displaystyle\prod_{i=1}^{N} \gamma_i^{k_i} & \text{if node } p \text{ and } j-1 \text{ other nodes all have the same largest number of conferees and } \displaystyle\sum_{n=1}^{N} k_n = s \\ 0 & \text{otherwise} \end{cases}$$

Removing the condition on s, the blocking probability B is given by

$$B = \sum_{s=2}^{M} B(s) w_s$$

C. Analysis of the Optimal Srategy for Common Media Conferences

For common media conferences, simplex channels instead of full duplex channels are allocated. If a common media conference uses node p as the conference bridge and has k_q conferees at node q, then it needs k_q channels on link qp for transmitting the video/voice packets from the conferees to the conference bridge and one channel on link pq for transmitting the composite video/voice packets from the conference bridge to the k_q conferees at node q. In general, the traffic loaded onto link pq can be depicted as in Fig.7. The probability $u_{pq}(j)$ that a conference using node p as conference bridge has j conferees at node q is

$$u_{pq}(j) = \sum_{i=1}^{N} \chi^{i}_{pq}(j)$$

Then the probability $v_{pq}(j)$ that a common media conference demands j channels on link pq can be expressed as:

$$v_{pq}(j) = \begin{cases} u_{qp}(1) + \sum_{i=1}^{\lfloor M/2 \rfloor} u_{pq}(i) & j = 1 \\ u_{qp}(j) & j > 1 \end{cases}$$

The traffic loaded onto link pq by $Poisson[\lambda_T]$ can then be decomposed into $Poisson[\lambda_T v_{pq}(1)]$, $Poisson[\lambda_T v_{pq}(2)], \cdots, Poisson\left[\lambda_T v_{pq}\left(\left\lfloor \frac{M}{2} \right\rfloor\right)\right]$, where each arrival of $Poisson[\lambda_T v_{pq}(i)]$ demands i channels on link pq. The state probabilities and link occupancies can be found in exactly the same manner as that for selectable media conferences. Given that a conference has s conferees, the blocking probability $B(s)$ is given by

$$B(s) = 1 - \sum_{\Omega_4} \sum_{p=1}^{N} \prod_{\substack{q=1 \\ q \neq p}}^{N} P[\tilde{N}_{pq} \le L_{pq} - \delta(k_q) \mid \tilde{K}_q = k_q] \cdot P[\tilde{N}_{qp} \le L_{qp} - k_q \mid \tilde{K}_q = k_q] \cdot$$
$$P\left[\tilde{K} = \tilde{k} \text{ and } \Theta_p \mid \sum_{n=1}^{N} k_n = s\right]$$

where

$$\delta(x) = \begin{cases} 1 & x > 0 \\ 0 & \text{otherwise} \end{cases}$$

Removing the condition on s, the blocking probability B is given by

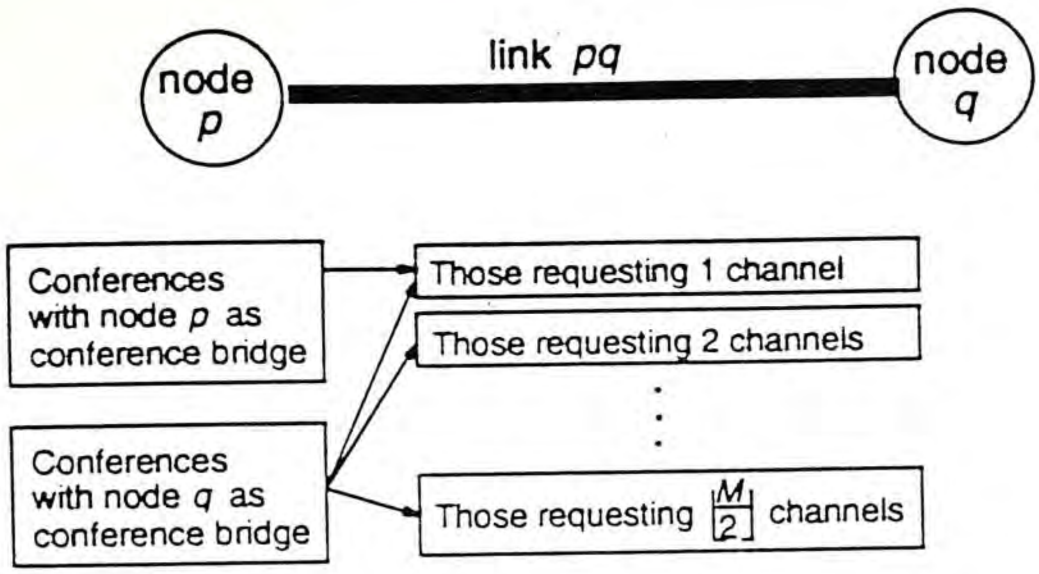


Fig.7: Traffic loaded onto link pq by common media conferences

$$B = \sum_{s=2}^{M} B(s) w_s$$

D. Examples and Discussions

In all the following examples, a five-node fully connected network is considered and $\mu = 1$, $\lambda_i = \lambda$ and $L_{ij} = L$ for all i and j are assumed.

Fig.8 shows the blocking probabilities of selectable media conferences using the optimal and the alternate strategies against the total offered load λ_T/μ assuming the conference size distribution is of the truncated geometric type (Fig.9) with a mean of 3. We see that in general the optimal strategy gives about one to two orders of magnitude smaller blocking than the alternate strategy. At $B=10^{-3}$ and for $L=30$, the network using the alternate strategy has a throughput of 89 Erlangs while using the optimal strategy it increases to 119 Erlangs. This 33.7% increase of network throughput should well justify the use of optimal connection paths for setting up conferences.

To study the sensitivity of blocking probability to the conference size distribution, consider the three hypothetical distributions shown in Fig.9. For the same mean conference size, the truncated geometric distribution has the largest variance and the delta distribution has the smallest (zero) variance. Fig.10 shows the throughput-blocking characteristics of selectable media conferences for these three distributions assuming the optimal strategy is used for placing conference bridges. It is seen that when the mean conference size is increased from 3 to 4, the maximum load the network can take (or the network capacity) is decreased from 165 Erlangs to 105 Erlangs for the geometric distribution at $B=10^{-3}$. The variance of the conference size distribution, on the other hand, is seen to have a small but non-negligible effect on the network

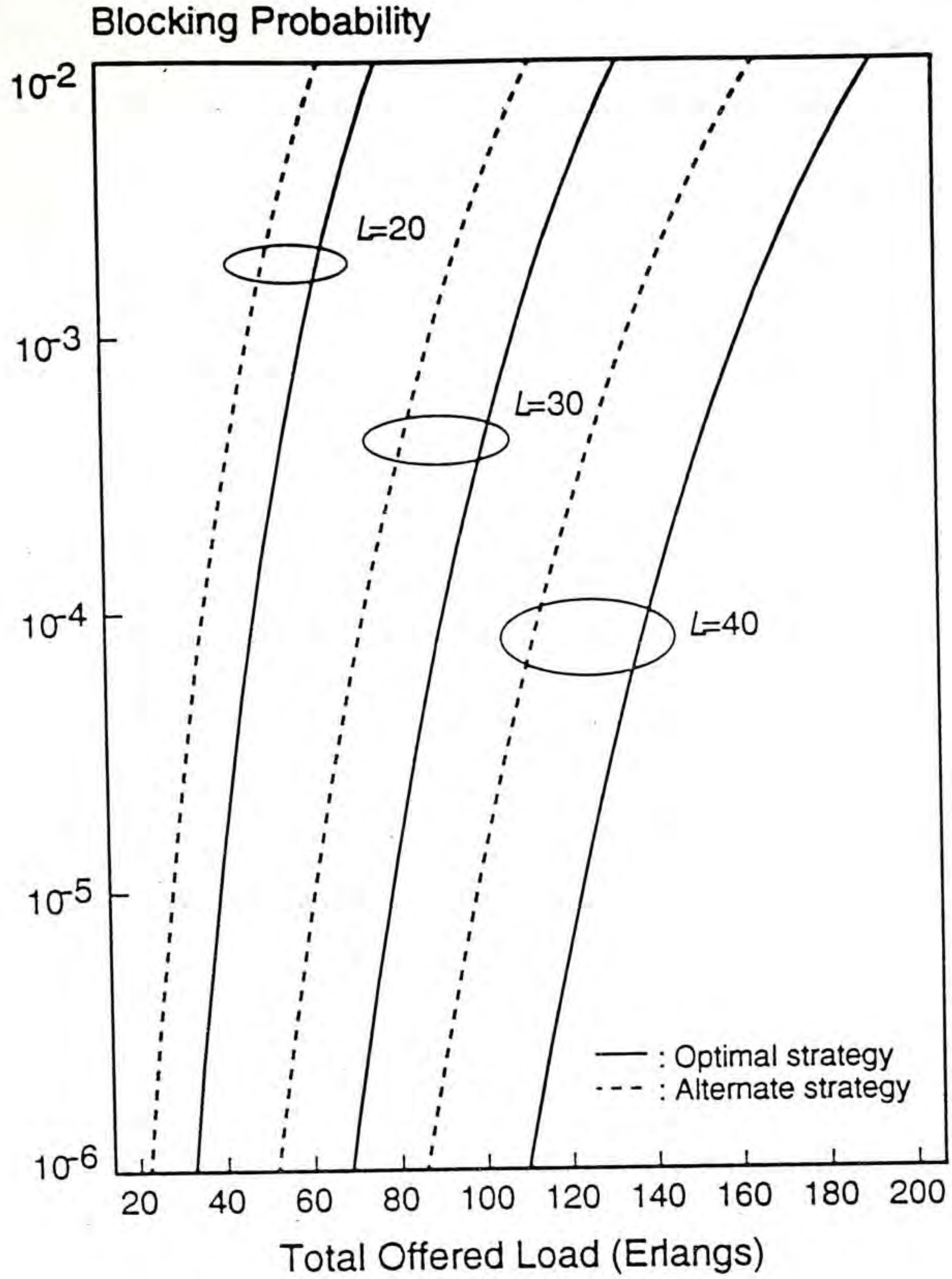
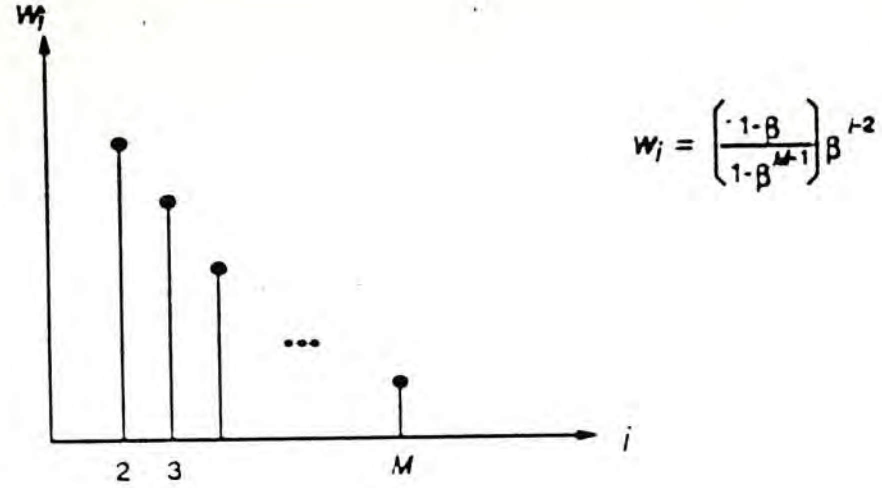
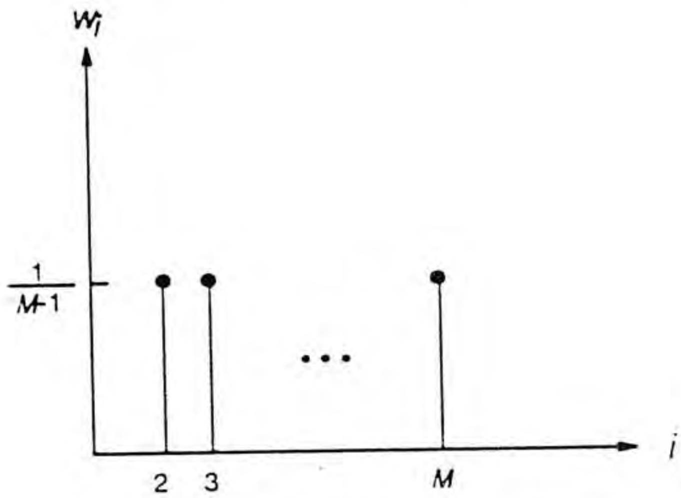


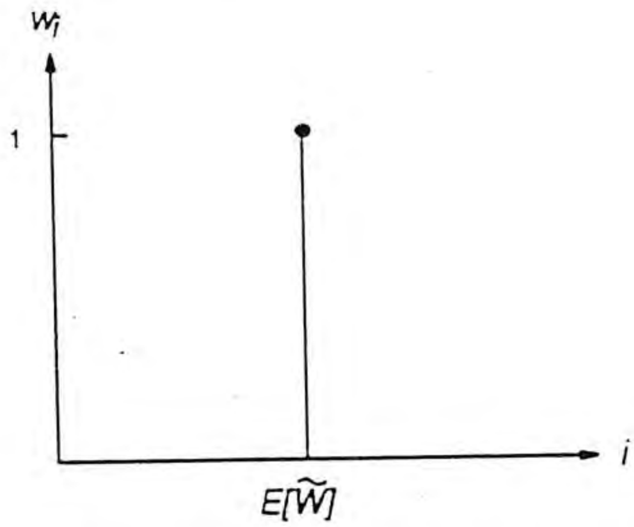
Fig.8: Comparison of the optimal and the alternate strategies $E[\widetilde{W}]=3$



(a) Truncated geometric distribution



(b) Uniform distribution



(c) Delta distribution

Fig.9: Three hypothetical conference size distributions

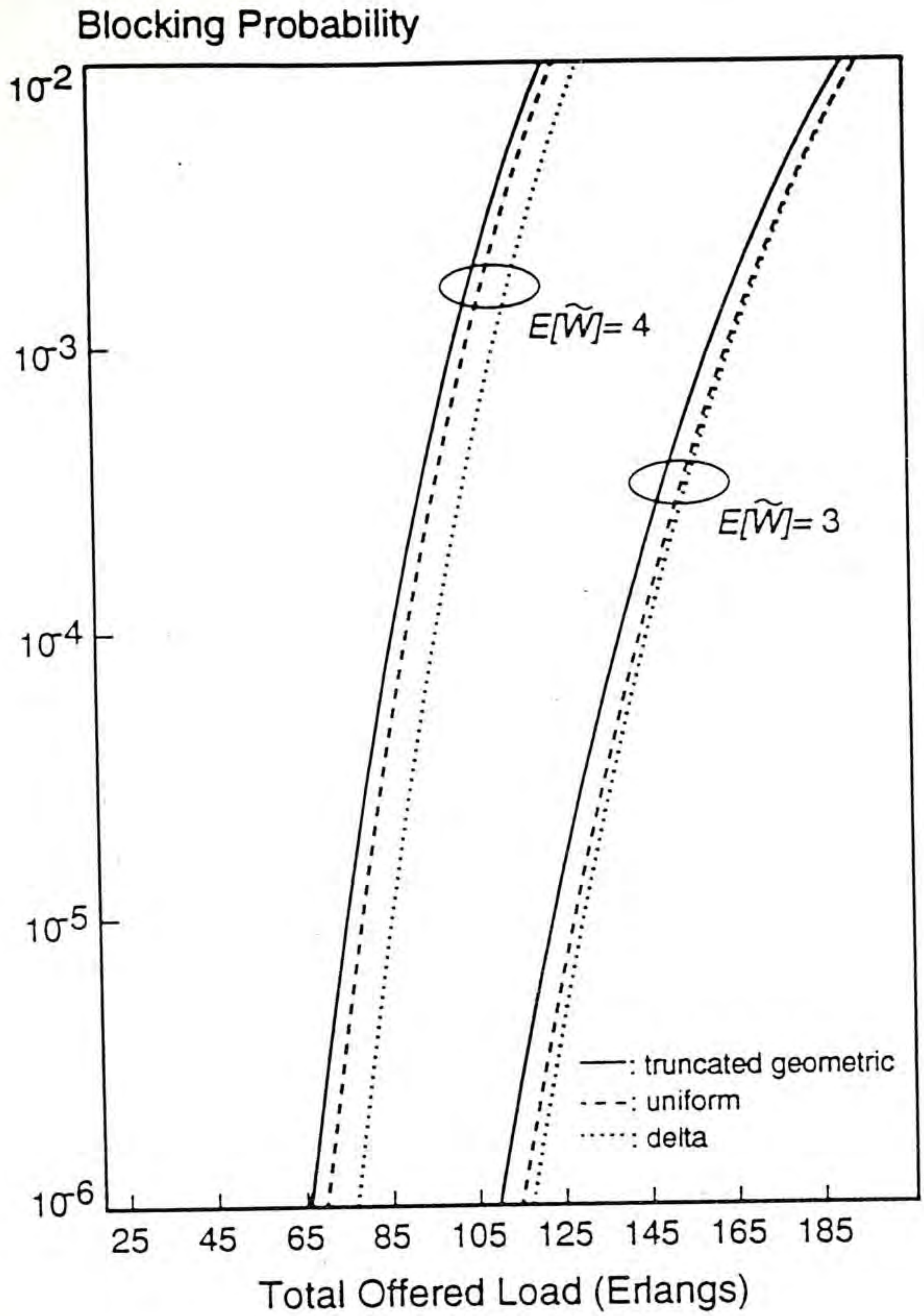


Fig.10: Comparison of different conference distributions (L=40)

throughput. For a given blocking requirement and a given mean conference size, the larger is the variance, the smaller is the throughput. The conference size distribution therefore is as essential as the conference traffic rate in traffic engineering.

Fig.11 compares the performance of selectable media and common media conferences. The optimal strategy is used for placing conference bridges and conference size is assumed to be a constant and equals 4, 5 and 6 for the three sets of curves. The common media conferences require less channel resources from the network than selectable media conferences because sharing of channels in the outbound connection is possible. The amount of sharing, moreover, increases with the conference size. Thus at $B=10^{-3}$ and conference size equal to 5, the maximum network throughput for common media and selectable media conferences are 83 and 78 Erlangs respectively. This represents a 6.4% larger throughput for common media conferences. When the conference size is 6, a 15.5% increase of throughput is observed for common media conferences. It should be noted that in fully connected networks, only conferees connected to the same node can share a channel. In a general network, those connected to different nodes may also be able to share channels in the outbound connection (i.e. the multicast tree). Hence, in a general network, the network throughput for common media conferences is expected to be much higher than that for selectable media conferences.

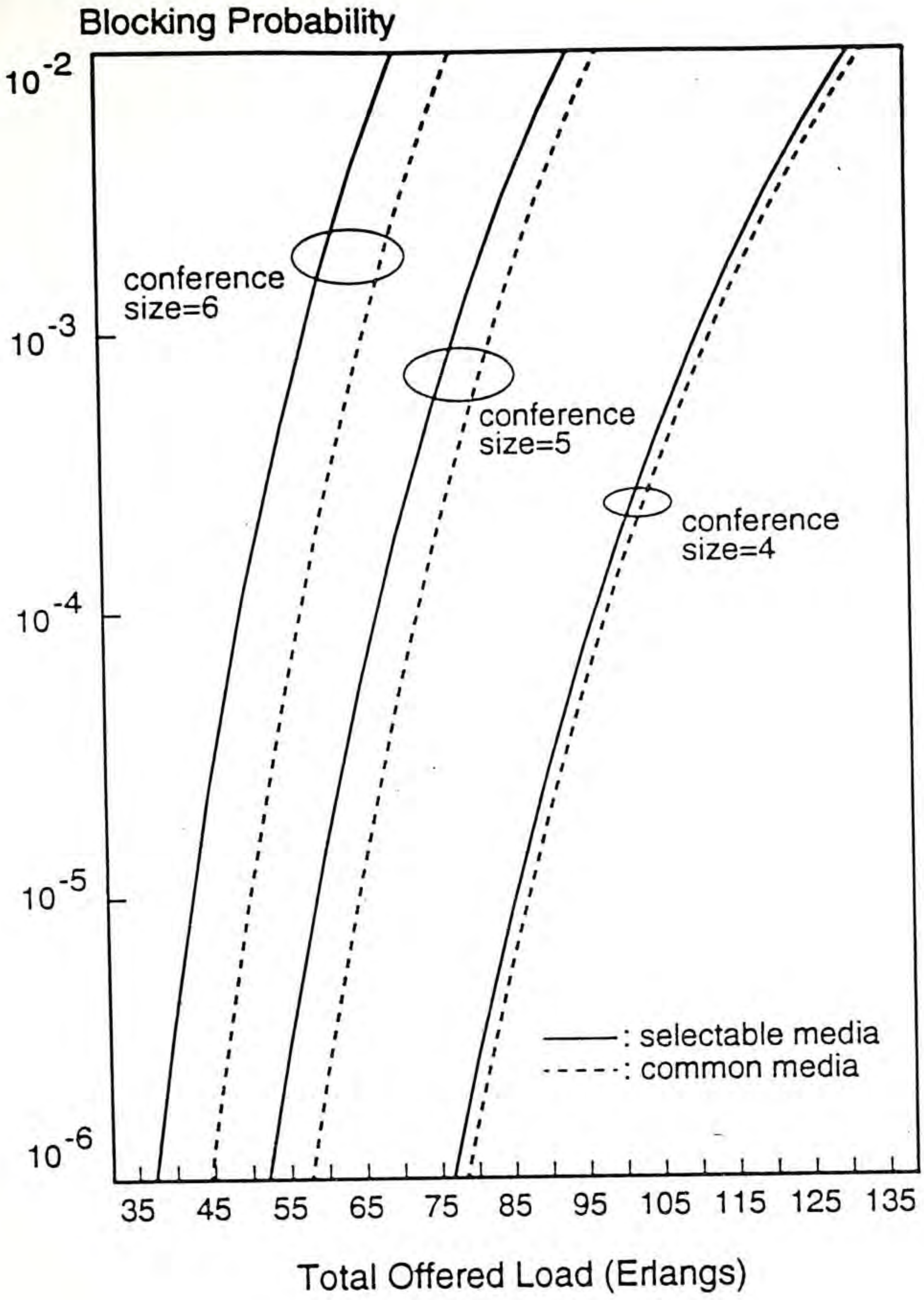


Fig.11: Performance comparison of selectable media and common media conferences (L=40)

3.5 Chapter Summary

The connection path problems for two types of multipoint videoconferences, called selectable media conferences and common media conferences, were formulated and algorithms were designed to find the connection paths with minimum number of channels. For on-demand calls which request immediate seizure of the transmission resources, a fast determination of the optimal connection path at call set-up time is required. A method was proposed that uses a combination of table look-up and on-line processing to reduce the computation time. The blocking probabilities as a function of network traffic for selectable media and common media conferences in fully connected networks were derived and compared. The sensitivity of network throughput to conference size distribution is also studied. It is found that, for a given mean conference size, the variance of the conference size distribution has a small but non-negligible effect on the network throughput.

Chapter 4
A TDM-based Multibus Packet Switch

A new packet switch architecture using two sets of time division multiplexed buses is proposed and analyzed. The *horizontal buses* collect packets from the input ports while the *vertical buses* distribute the packets to the output ports. The two sets of buses are connected by a set of switching elements which coordinate the connections between the horizontal buses and the vertical buses so that each vertical bus is connected to only one horizontal bus at a time. The switch has the advantages of: (1) very simple control circuit, (2) 100% potential throughput under heavy traffic, (3) internally unbuffered, and (4) adding input and output ports without increasing the bus and I/O adaptor speed. A combined analytical-simulation method is used to obtain the packet delay and packet loss probability. Numerical results show that for satisfactory performance the buses need to run about 30% faster than the aggregate input ports rate. With this speedup, even at a utilization factor of 0.9, the input queue can give a packet loss rate of 10^{-6} with only 31 buffers per input adaptor. The output queue behaves essentially as an M/D/1 queue.

4.1 Background

To support a wide variety of broadband communication services, a broadband network should be able to accommodate a large volume of traffic having different characteristics and transmission requirement. Due to the large throughput requirements, switching can no longer be done satisfactorily in the conventional uniprocessor share memory machines. In recent years, considerable effort has been made in developing high speed packet switches for broadband communications. Recent survey and taxonomy can be found in [TOBA 90, AHMA 89, HUI 89]. It is generally agreed that high speed packet switches can be classified into three broad types: shared-memory based, shared-medium based and space-division based. Designing broadband packet switches for the next generation communication networks requires a broad knowledge base and foresight in technological development. The shared-medium based switch has among its advocates the IBM's PARIS switch designers [CIDO 88a, CIDO 88b], the NEC's ATOM switch designers [SUZU 89] and the Fujisu's Bus Matrix Switch designers [NOJI 87]. We describe these switches briefly as follows.

A. PARIS Switch [CIDO 88a, CIDO 88b]

The PARIS switch is designed for private network and its design objectives are to use a simple architecture and a simple protocol for real-time as well as non-real-time packet switching. Fig.1 shows the architecture of the PARIS switch. It uses a high-speed shared bus to connect the input ports to the output ports. The capacity of the shared bus is equal to or larger than the aggregate capacity of all input ports. The switch can handle variable size packets ranging from 32 bits to a maximum of 8 kbits. An exhaustive round robin policy is adopted for bus arbitration. Cidon *et.al* [CIDO 88b] showed that: (a) this policy can provide a guaranteed delay bound for real-time packets and a small average delay for non-real-time packets; (b) using 4 buffers in each input adaptor can already ensure no packet loss in the input adaptors; and (c) using a small number of buffers in the output adaptors can already give a very small probability of packet loss in the output adaptors.

B. ATOM Switch [SUZU 89]

Fig.2 shows the architecture of an ATOM switch. All packets arriving on the input ports are first converted into parallel bit streams and then they are synchronously multiplexed onto a time-division high-speed bus. The capacity of the bus is equal to the aggregate capacity of the input ports. Each output port is connected to the bus via an interface consisting of a packet

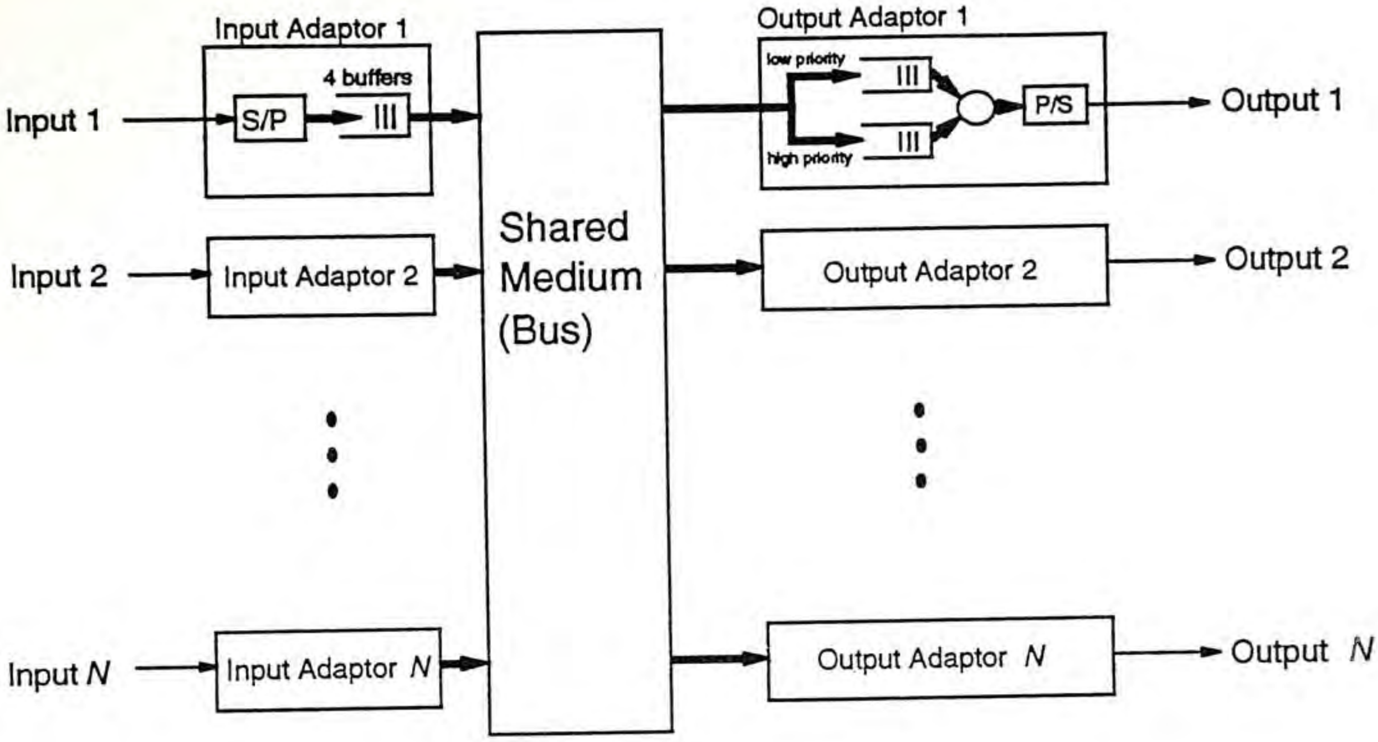


Fig.1: PARIS Switch

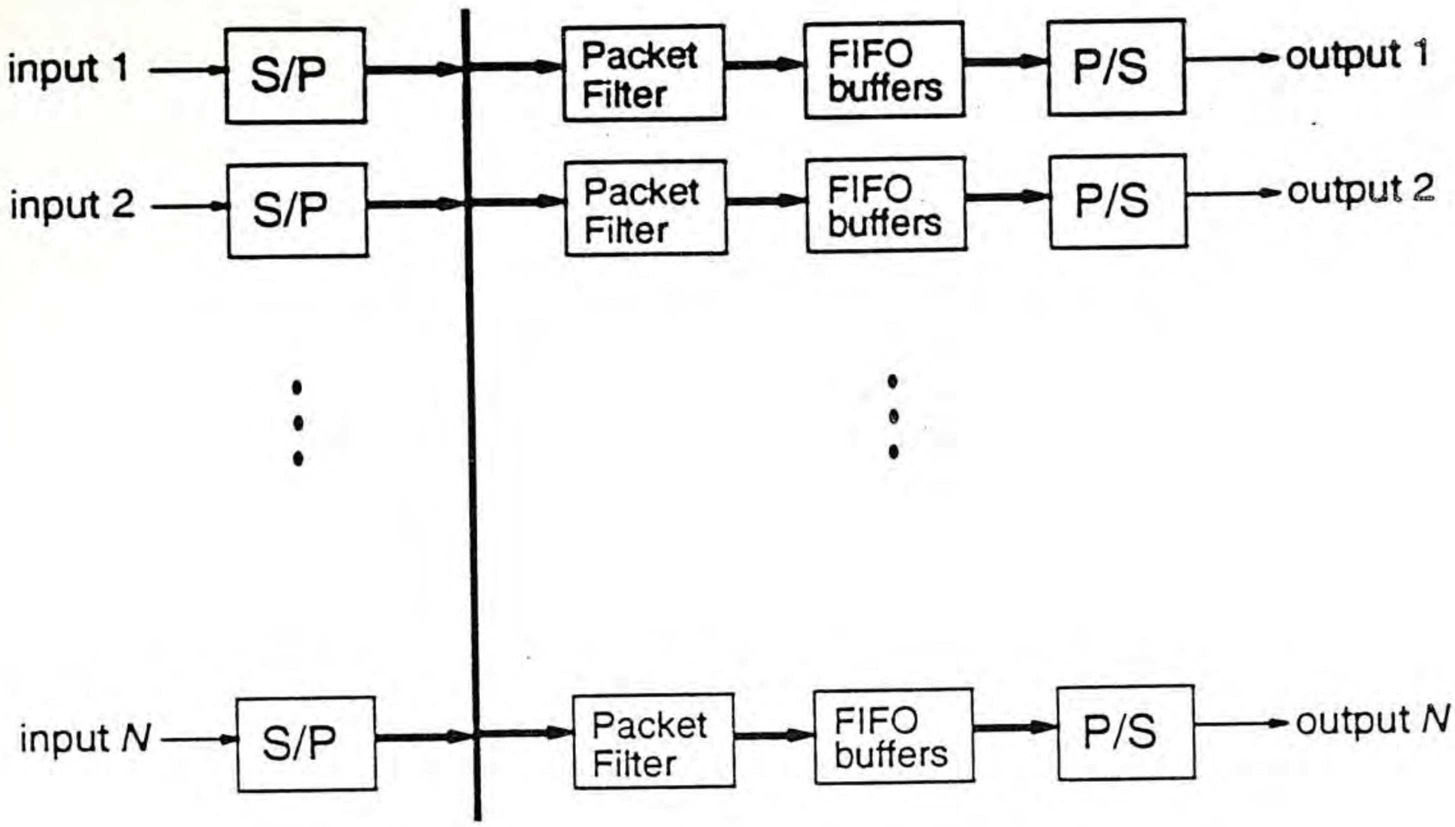


Fig.2: ATOM Switch

filter, FIFO buffers and a parallel-to-serial converter. The packet filter receives all packets transmitted on the bus. By inspecting the output address in the header of each packet, it determines whether the currently transmitting packet on the bus is to be written into the FIFO buffers. The packets stored in the buffer are first converted into serial bit streams and then they are forwarded to the output port.

To implement a high speed ATOM switch, we can use either a single bus with high speed bus interface circuits connected or a bit slice organization as shown in Fig.3. Incoming serial bit stream of each input port is converted into P serial streams, each feeding one of the P parallel ATOM sub-switches. Within a sub-switch, each serial bit stream is again converted into parallel streams for multiplexing onto the bus. Since the address in each packet is "broken" into P pieces, the FIFO buffers in each sub-switch must be informed when to store the "sub-packets" transmitted on the bus. A centralized address controller is used to process the headers of all incoming packets and inform the FIFO buffers in each sub-switch when to store the sub-packet transmitted on the bus. This is done by extracting the headers of all incoming packets and routing them to the address controller. The address controller can then determine which FIFO buffers should the incoming packets be written into and sends the appropriate write control signals to the sub-switches.

To construct a large switch, a multistage organization is proposed [SUZU 89] and is shown in Fig.4. Store and forward of packets, however, is needed at every stage.

C. Bus Matrix Switch [NOJI 87]

Fig.5 shows the architecture of a Bus Matrix switch. It uses multiple shared buses to construct a large switch. The shared buses are connected in matrix form with memory located

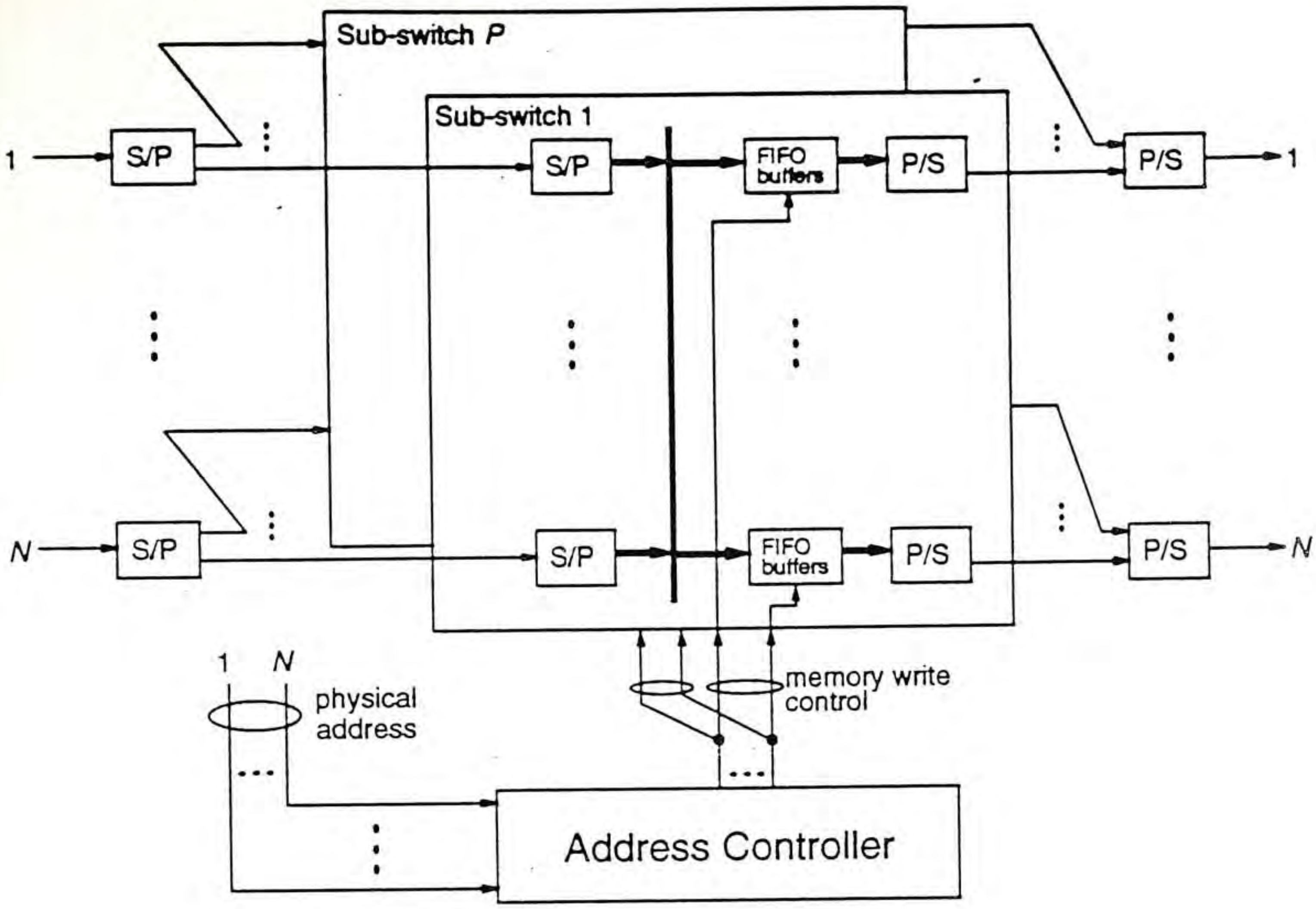


Fig.3: Bit slice organization of the ATOM switch

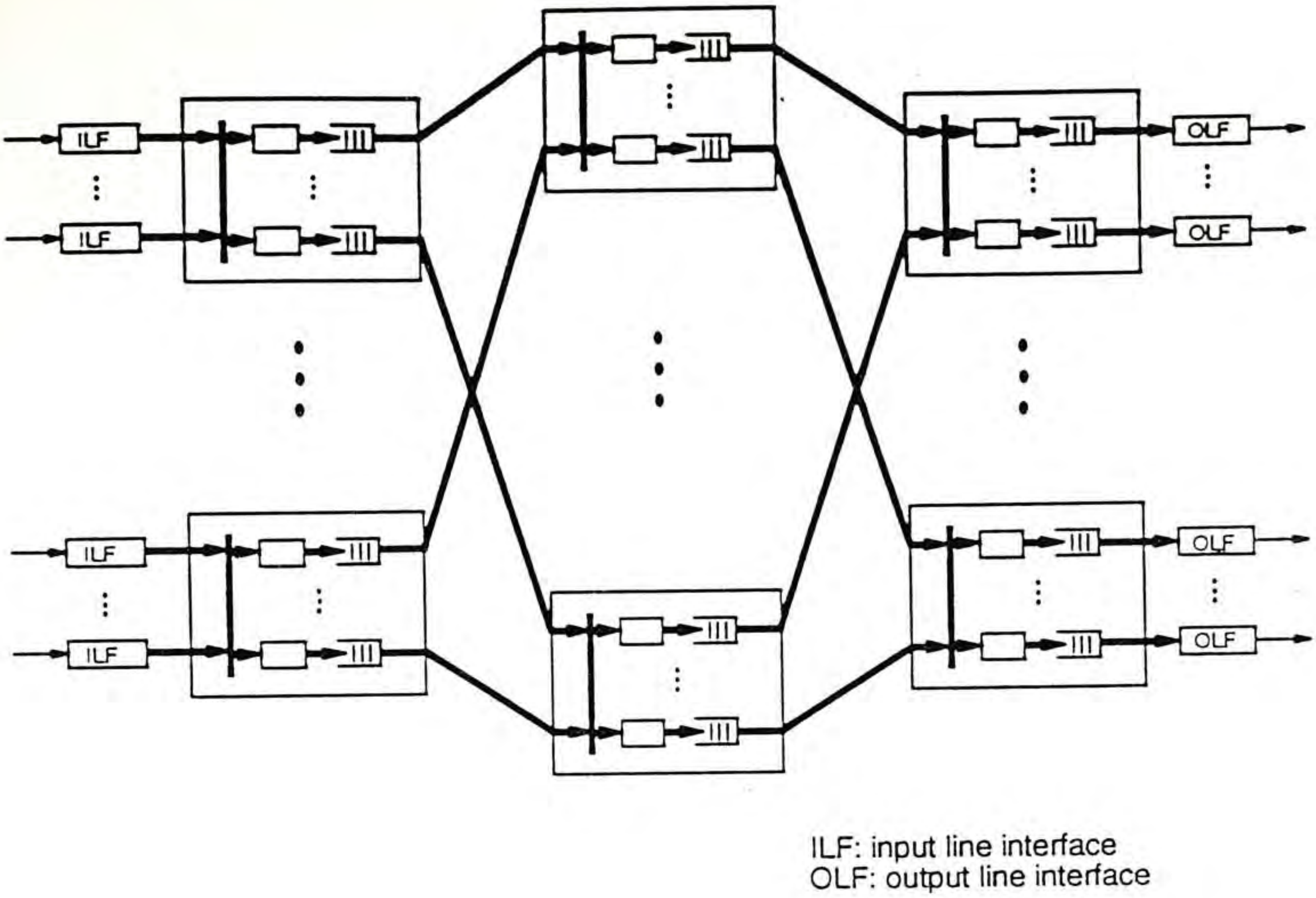


Fig.4: ATOM's multistage configuration

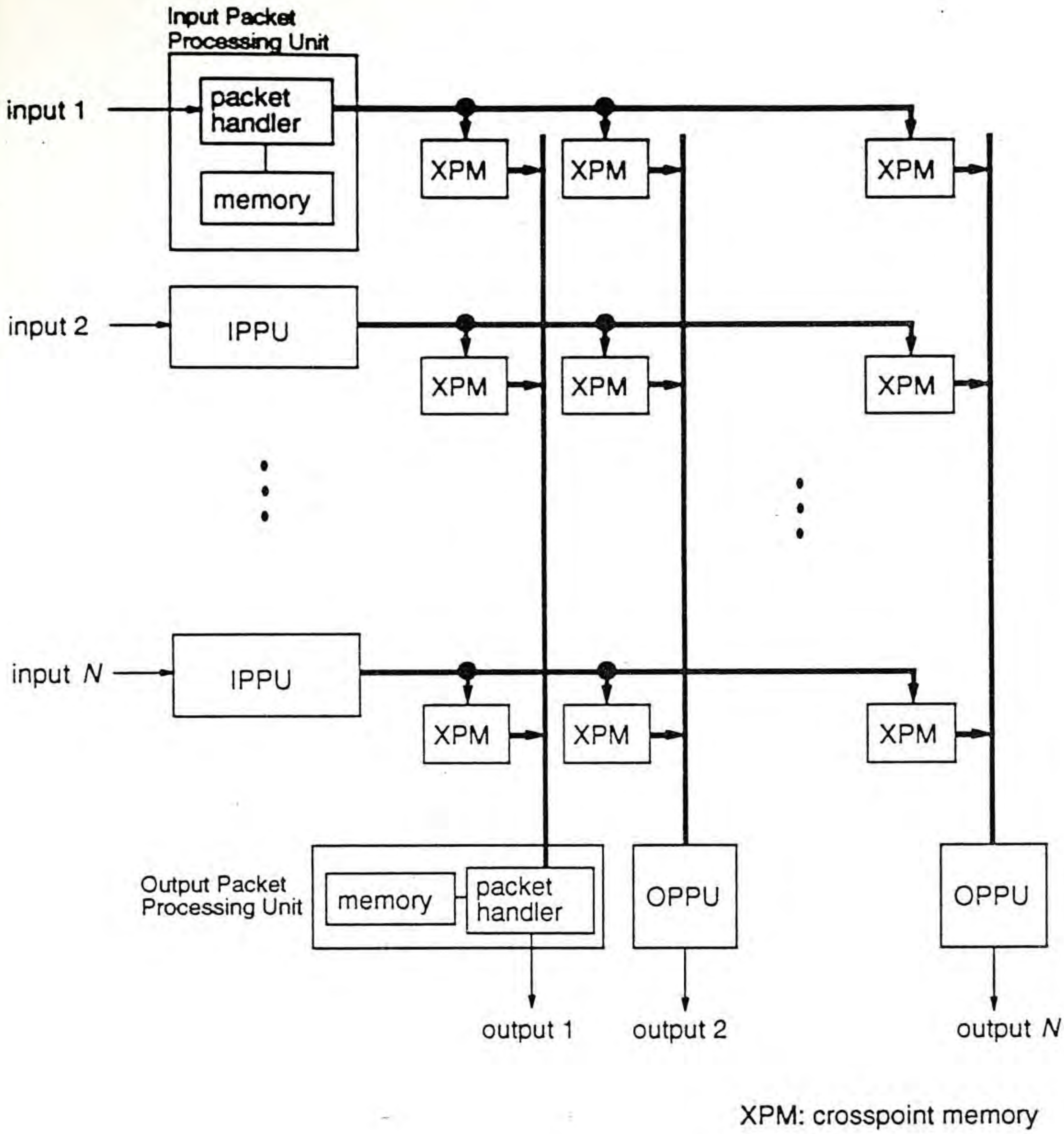


Fig.5: Bus Matrix Switch

at each crosspoint of the buses. Packets contending for access to the same bus are stored in the crosspoint memories connected to this bus. Arbiters (not shown in Fig.5) scan the crosspoint memories and remove packets from them.

In this chapter, we study a new switch architecture using multiple shared buses. This switch has the advantages of: (1) very simple control circuit, (2) 100% potential throughput under heavy traffic, (3) internally unbuffered, (4) adding input and output ports without increasing the bus and I/O adaptor speed.

4.2 Architecture of the Multibus Switch

Fig.6 shows the architecture of an $N \times N$ multibus switch. Each input port and each output port are connected to the switch through an input adaptor and an output adaptor respectively. Fig.7 shows the internal structure of an input adaptor and an output adaptor. The input adaptor receives packets from the input port, performs a serial to parallel conversion and queues the packets in a set of buffers. The output adaptor receives packets from the vertical bus through a packet filter, which identifies packets destined for it, and queues them in a set of buffers for transmission.

The N input and output ports are partitioned in M groups of L ports each. Here we restrict the choice of N such that $N=ML$. Group i input adaptors are connected to *horizontal bus* HB_i and group j output adaptors are connected to *vertical bus* VB_j. The M horizontal buses are connected to the M vertical buses in bus matrix form, with a total of M^2 *switching elements* at the crosspoints of the vertical and horizontal buses. The switching element placed at the crosspoint of HB_i and VB_j is identified as $S_{i,j}$. Fig.8(a) shows the schematic of the switching element. It connects the horizontal input bus to either the horizontal output bus or the vertical bus. Fig.8(b) shows the circuit realization of the switching element, using $2b$ *relays* (b is the

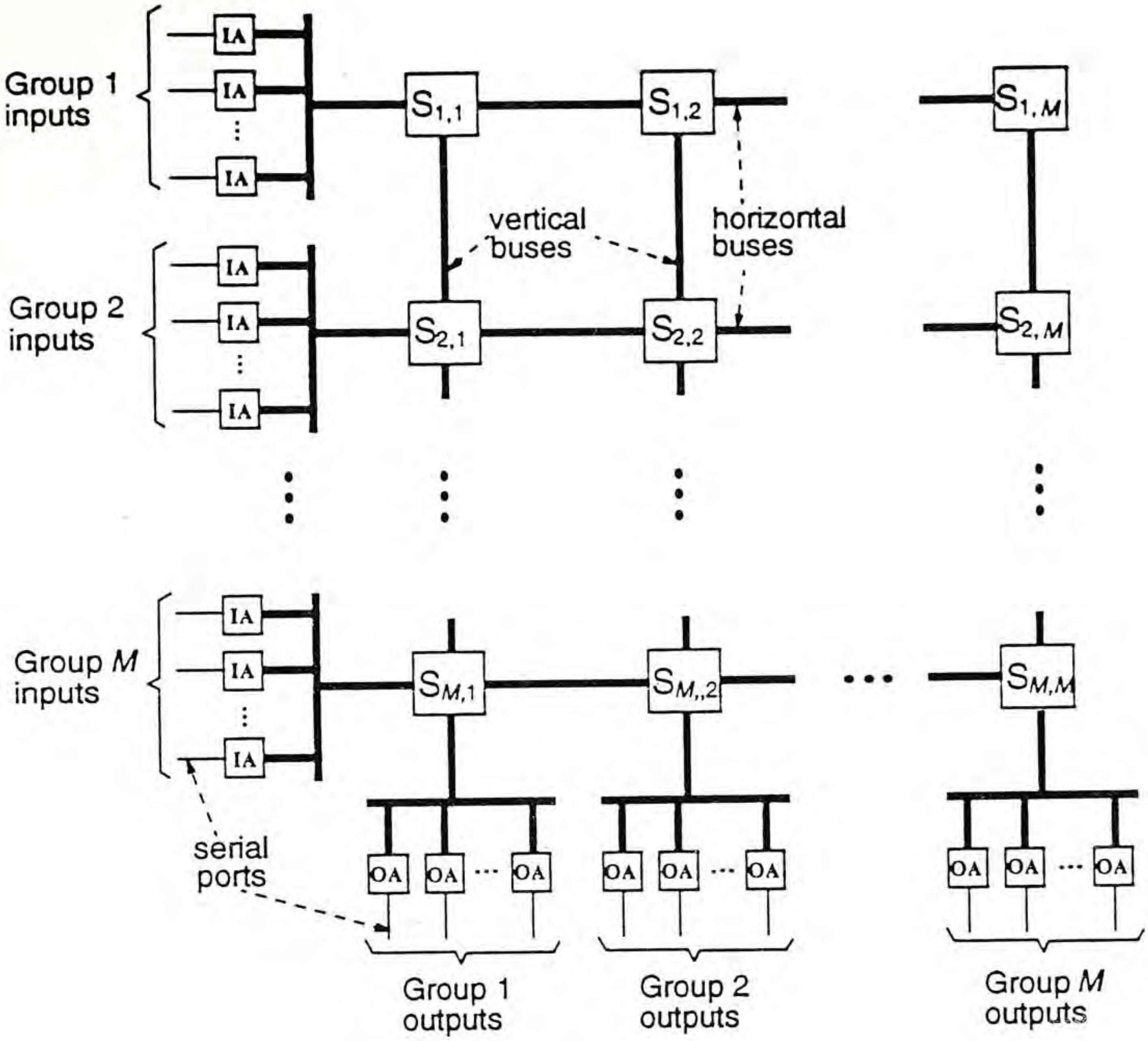


Fig.6: The multibus packet switch

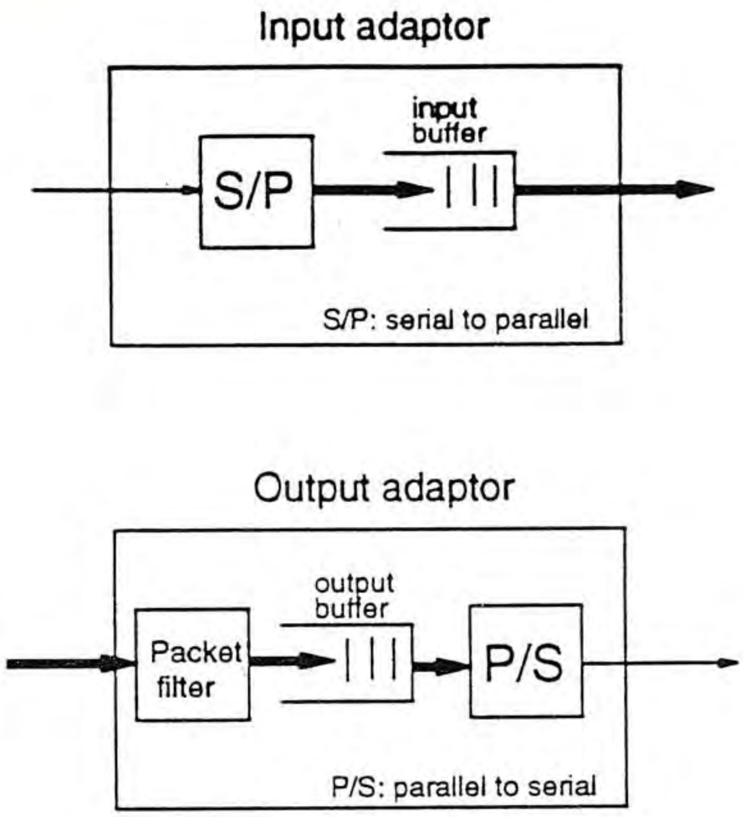
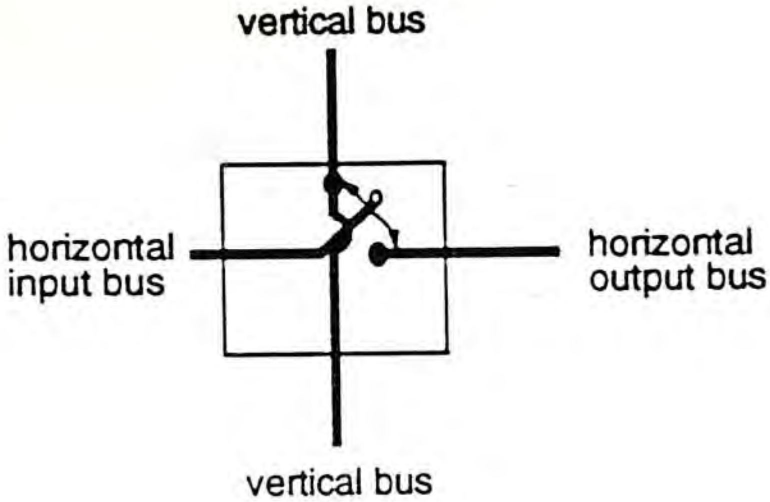
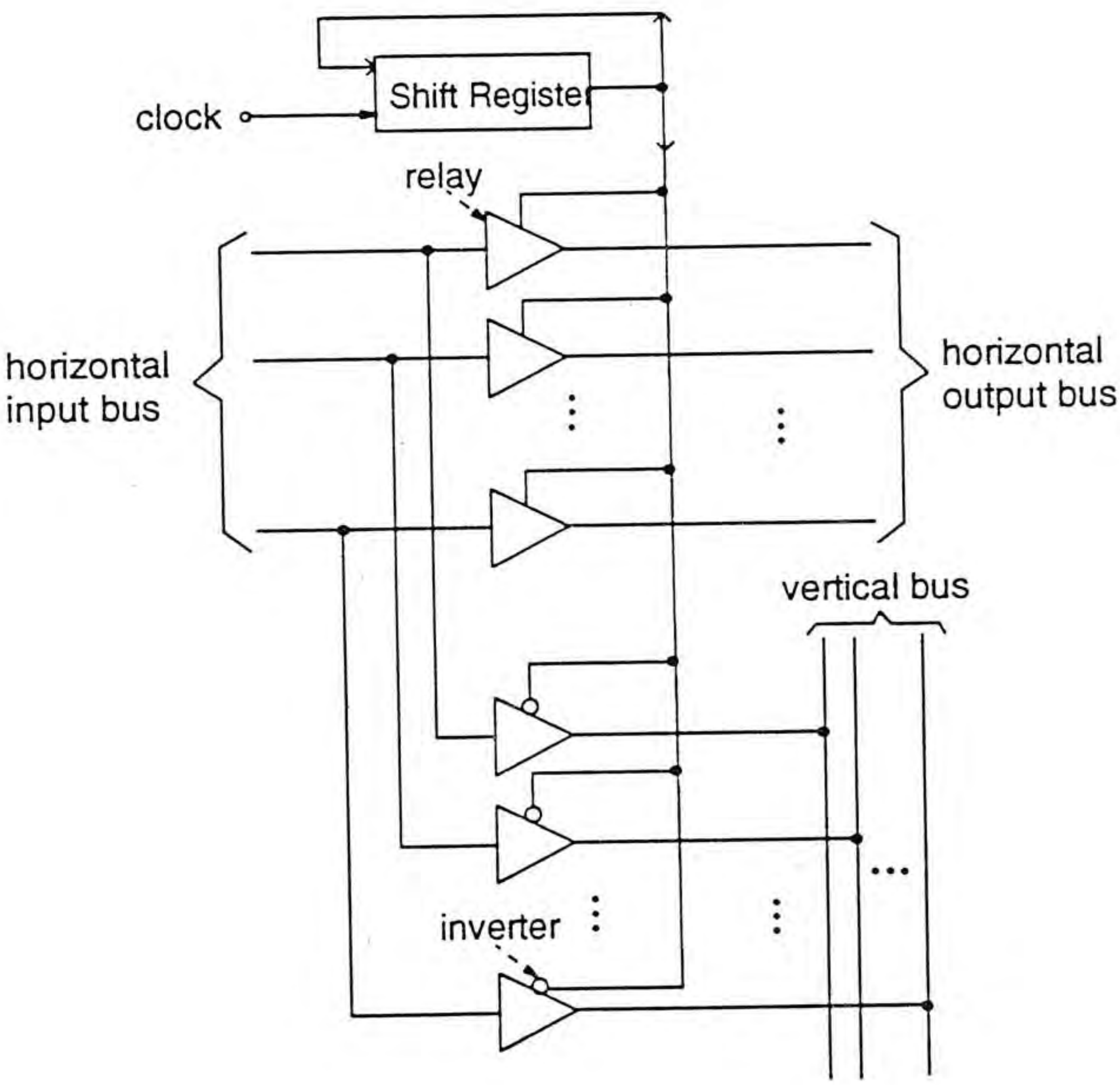


Fig.7: Input and output adaptors



(a) Schematic of the switching element



(b) Circuit realization of the switching element

Fig.8: The switching element

bus width), b inverters and one shift register. The relay is a high input impedance two-states elements. When its control input is HIGH, it copies the content in the input to the output; when its control input is LOW, it disconnects the output from the input. For prototyping, the set of relays are available as an off the shelf IC chip (e.g. Motorala's SN54LS). For actual implementation, we can design ASIC chips with multiple switching elements per chip. Since the circuitry in each switching element is very simple, the number of switching elements per chip is only limited by the pin size. The shift register in $S_{i,j}$ stores a bit pattern which determines when to connect the horizontal input bus to the vertical bus. When a clock pulse arrives, the last bit is shifted out to the relays. If this bit is "1", the horizontal input bus is connected to the horizontal output bus; otherwise, the horizontal input bus is connected to the vertical bus. The connection patterns of the switching elements are chosen such that one vertical bus is connected to only one horizontal bus at a time.

4.3 Operation of the Multibus Switch

The control of the packet switch is divided into *cycles* of equal length. Each cycle is sub-divided into M *subcycles* of equal length (Fig.9). In the i^{th} subcycle, group j input adaptors are connected to vertical bus $VB_{f(i,j)}$ where

$$f(i,j) = \begin{cases} (i+j-1) \bmod M & (i+j-1) \bmod M \neq 0 \\ M & (i+j-1) \bmod M = 0 \end{cases} \tag{1}$$

(Note that if we would have labelled the vertical buses as $0,1,2,\cdots,M\text{-}1$ instead of $1,2,\cdots,M$, (1) will be simplified to $f(i,j) = (i+j) \bmod M$. But doing so would complicate the subsequent discussion). Thus in the i^{th} subcycle packets from group j input adaptors are switched to group $f(i,j)$ output adaptors. Hence, only the switching elements $S_{j,f(i,j)}$ (j=1,2,...,M) connect the horizontal buses HB_j to the vertical buses $VB_{f(i,j)}$ (j=1,2,...,M) while all the others connect the horizontal input buses to the horizontal output buses. Fig.9 shows an example of this transmission

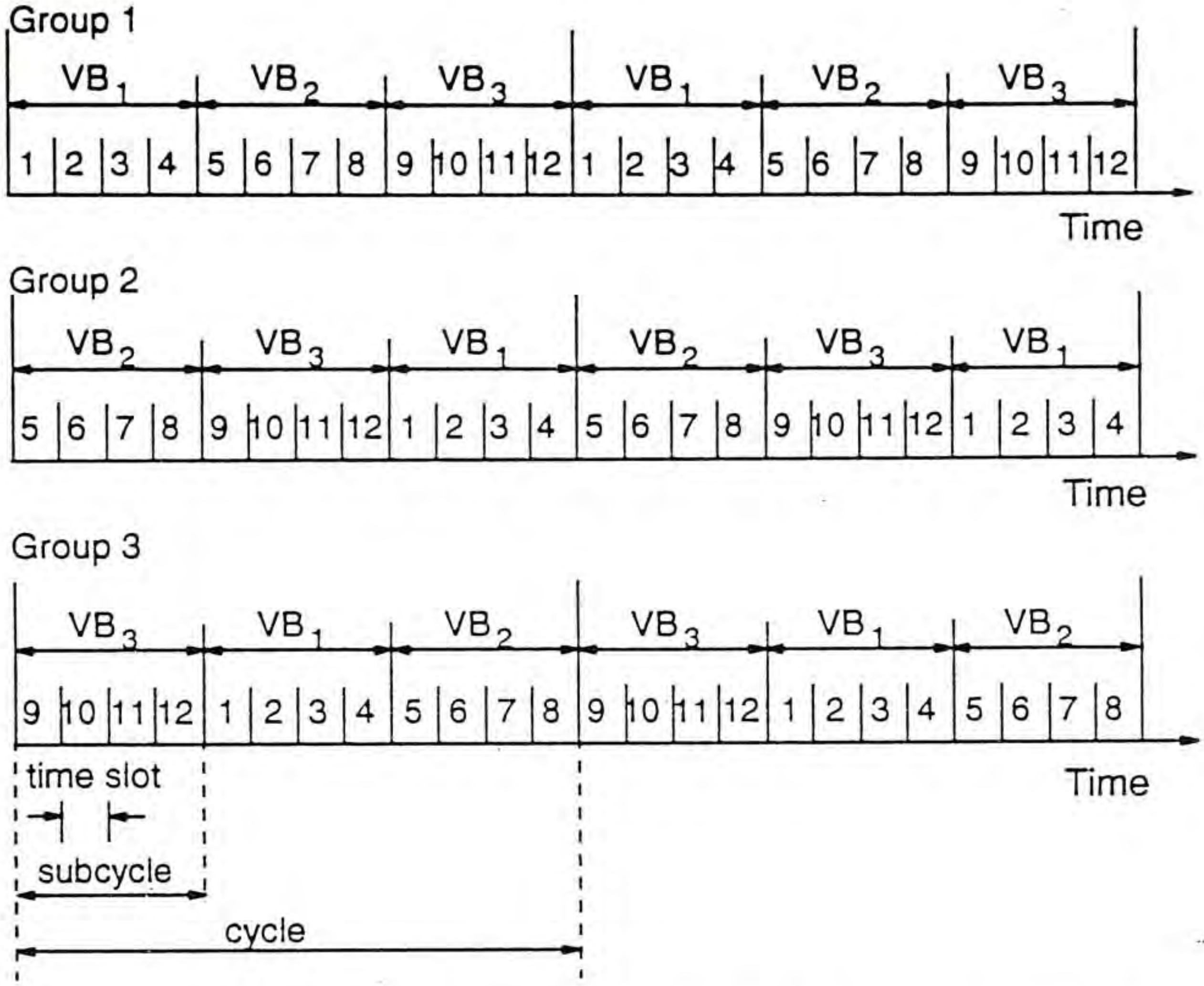


Fig.9: Transmission cycles, subcycles and time slots;M=3, N=4.

arrangement when M=3 and L=4. This transmission arrangement ensures that in each subcycle there is a unique one-to-one connection from every group of input adaptors to every group of output adaptors. This means that the M groups of input adaptors can simultaneously transmit packets to the M groups of output adaptors through the bus matrix.

To resolve the bus contention among the L input adaptors in each group, each subcycle is further divided into L time slots where each time slot is dedicated to one input adaptor. The duration of a time slot is set to equal to one packet transmission time on the bus. Each adaptor can therefore transmit one packet in each subcycle.

Global timing is used to ensure that all transmissions are properly synchronized. This requires all the input adaptors and switching elements be triggered by a common clock.

4.4 Performance Analysis

A. Queueing Model

The input port is a synchronous data link with time being divided into *link-slots*. A packet arrives in a link-slot with probability ρ. Let α_{ij} be the probability that an incoming packet from input port i is destined for output port j. Then the traffic rate β_{ik} from input port i to group k output ports is:

$$\beta_{ik} = \rho \sum_{j=(k-1)L+1}^{kL} \alpha_{ij} \tag{2}$$

The packets in an input adaptor are logically organized into M queues such that queue j contains all packets destined for group j output ports. For convenience, we let $Q_{i,j}$ denotes queue j in input adaptor i. Let B_1 and B_2 be the buffer size in each input and output adaptor respectively. The queues $Q_{i,j}$, j=1,2,...,M, share these buffers by the complete sharing strategy [KAMO 80]. Packets queued at the output adaptor are transmitted in a first-in-first-out order.

Without loss of generality, let us consider the delay of the packets departing from group 1 output ports. As group 1 output adaptors only get packets from VB_1, we shall model VB_1 as a *bus server*. We shall, for convenience, call the subsystem up to and after the bus server as the *input queueing system* and *output queueing system* respectively. As seen from Fig.10 there are altogether N input queues $Q_{1,1}, Q_{2,1}, \cdots, Q_{N,1}$ feeding packets to the bus server VB_1. The output queueing system consists of L queues corresponding to the L output adaptors in group 1.

The switching elements connecting the horizontal buses and the vertical buses are operated in such a way that each input queue has a fixed dedicated slot for transmitting a packet in every cycle. All input queues are therefore independent. Let the transmission time of a packet on the bus be τ. Then the cycle length is $N\tau$. All input queues are served once every $N\tau$ seconds with service time τ. Let K be the number of link-slots in a cycle. Analysis of the input queues is given in the next subsection.

The arrival process to the output queueing system is the superposition of the departure processes of all the input queues in the input queueing system. To characterize this arrival process, we must first characterize the departure processes of the input queues. The bus server visits an input queue every $N\tau$ seconds and removes one packet from the queue when the queue is not empty. As far as the characterization of the departure process is concerned, the service time in the input queue can be considered as equal to $N\tau$ seconds. For input queue $Q_{1,1}$, packet departure occurs at time epochs that are integer multiples of $N\tau$. Therefore the durations of the busy and idle periods of the departure process of $Q_{1,1}$ are both integer multiples of $N\tau$. The input queues $Q_{2,1}, Q_{3,1}, \cdots$ are served by the bus server in a similar manner as for $Q_{1,1}$ except for a time lag of τ, 2τ, ... seconds respectively. The probability mass functions of the idle period and busy period durations for input queue $Q_{m,n}$ are derived in section 4.4.D. In general, the departure epoch for the input queue $Q_{i,1}$ occurs at $(kN + i - 1)\tau$ for k=1,2,$\cdots$. The departure process of each of the input queues is characterized by these time epochs and the distributions of the idle and

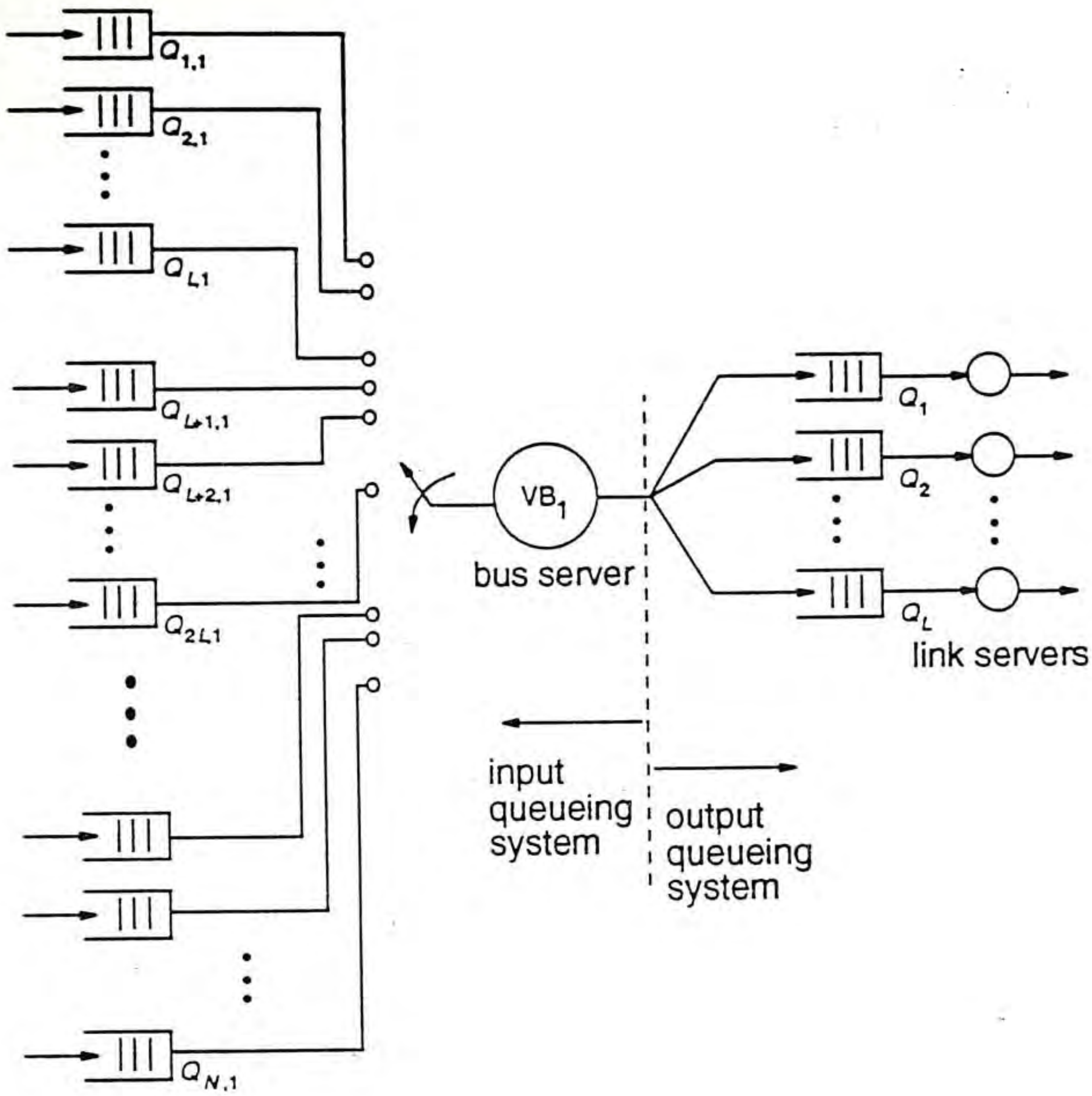


Fig.10: Queueing model for vertical bus VB_1

busy periods. The arrival process to the output queueing system is the superposition of the departure processes from all the input queues. Fig.11 shows an example of the departure process from the input queueing system with $M=L=2$.

Each departure from $Q_{i,1}$ may join output queues j in group 1 output adaptors with probability α_{ij}. Hence the departure process from $Q_{i,1}$ to output queue j is just the α_{ij} bifurcation of the departure process from $Q_{i,1}$. The arrival process to output queue j is the superposition of the N bifurcated departure processes from $Q_{i,1}$, $i=1,2,\cdots,N$, to output queue j. With such a complicated arrival process, we have to resort to computer simulation to obtain the queueing delay. As only a single server queue is simulated, very accurate delay and packet loss statistics can be obtained.

B. Expected Delay in the Input Queue

The buffer size B_1 in the input adaptor must be chosen such that the packet loss probability is very small. For example, a packet loss requirement of 10^{-6} is required for data packets to give a good throughput performance and a packet loss requirement of 10^{-9} is required for video packets to maintain an acceptable video quality [HONG 91]. Then we can approximate the expected delay with a finite buffer by the delay with an infinite buffer. As all input queues are similar, we choose to analyze a particular one with input rate β and use the imbedded Markov chain analysis technique. The input queue is served once every cycle for τ seconds. The imbedded points are chosen at the time instances at which the bus server has just visited the input queue. Let π_i be the probability that there are i packets in an input queue at the imbedded points and $G_Q(z)$ be the generating function $\sum_{i=0}^{\infty}\pi_i z^i$. In addition, let $\bar{q}_i$ be the number of customers in the input queue at the i^{th} imbedded point. Then $\bar{q}_{i+1}$ is related to $\bar{q}_i$ by:

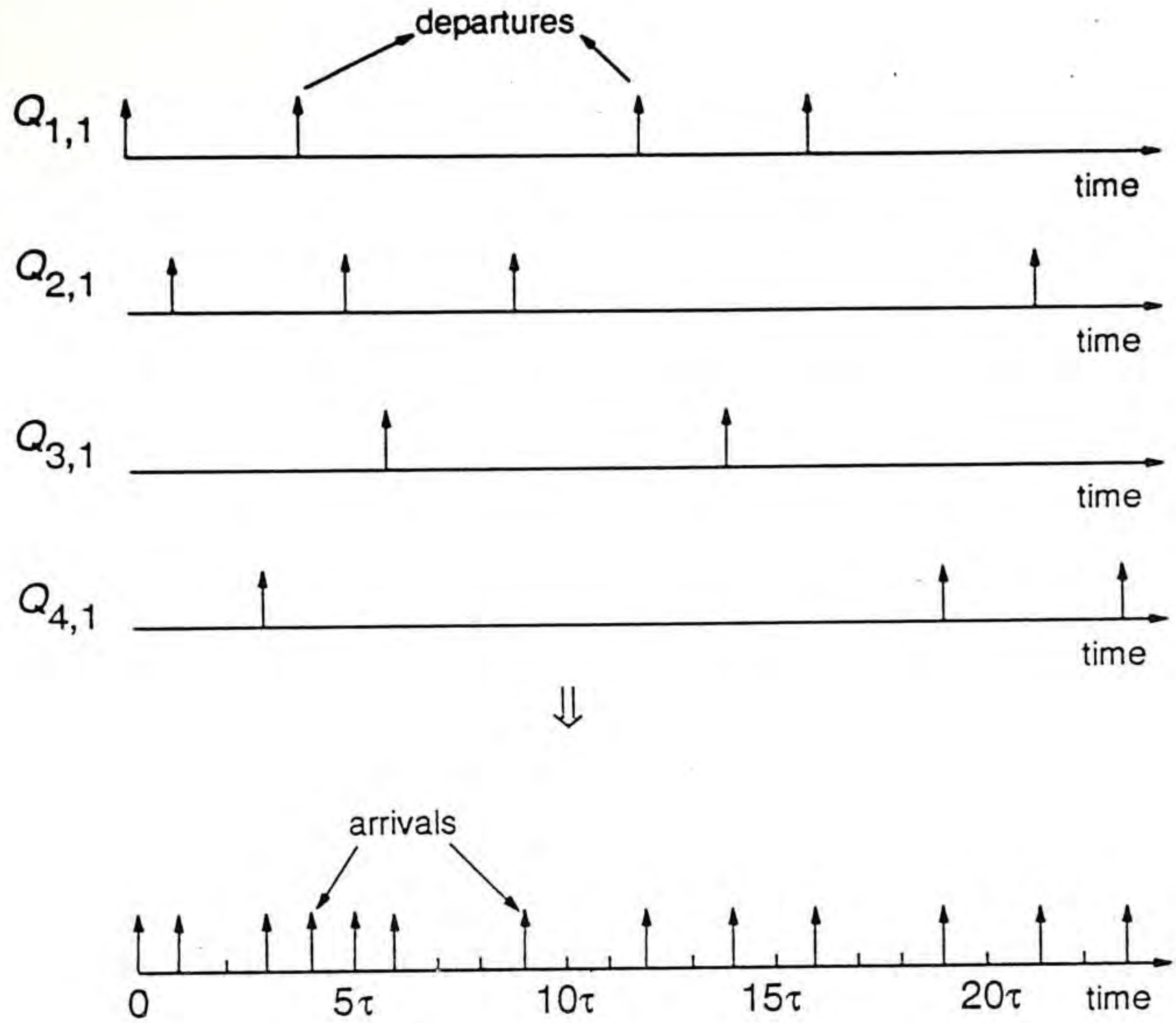


Fig.11: Arrival process to output queueing system is the superposition of the departure processes

$$\tilde{q}_{i+1} = \begin{cases} \tilde{q}_i + \tilde{a} - 1 & \tilde{q}_i > 0 \\ \tilde{a} - 1 & \tilde{a} > 0, \tilde{q}_i = 0 \\ 0 & \tilde{a} = 0, \tilde{q}_i = 0 \end{cases} \tag{3}$$

where $\tilde{a}$ is a random variable denoting the number of arrivals in one cycle. Since at most K packets arrive at an input port in a cycle, $\tilde{a}$ cannot be larger than K. Taking z-transform, we have

$$\begin{aligned} E\left[z^{\tilde{q}_{i+1}}\right] &= E\left[z^{\tilde{q}_{i+1}} \mid \tilde{q}_i > 0\right] P[\tilde{q}_i > 0] + E\left[z^{\tilde{q}_{i+1}} \mid \tilde{q}_i = 0\right] P[\tilde{q}_i = 0] \\ &= E\left[z^{\tilde{q}_i + \tilde{a} - 1} \mid \tilde{q}_i > 0\right] P[\tilde{q}_i > 0] + \\ &\qquad \{ E[z^0 \mid \tilde{a} = 0] P[\tilde{a} = 0] + E[z^{\tilde{a}-1} \mid \tilde{a} > 0] P[\tilde{a} > 0] \} \; P[\tilde{q}_i = 0] \\ &= (1 - \pi_0) z^{-1} E[z^{\tilde{a}}] \; E\left[z^{\tilde{q}_i} \mid \tilde{q}_i > 0\right] + \pi_0 \{ (1-\beta)^K + [1 - (1-\beta)^K] z^{-1} E[z^{\tilde{a}} \mid \tilde{a} > 0] \} \end{aligned} \tag{4a}$$

where

$$\begin{aligned} E[z^{\tilde{a}}] &= \sum_{i=0}^{K} \binom{K}{i} \beta^i (1-\beta)^{K-i} z^i \\ &= (1 - \beta + \beta z)^K \end{aligned} \tag{4b}$$

$$\begin{aligned} E\left[z^{\tilde{q}_i} \mid \tilde{q}_i > 0\right] &= \sum_{n=1}^{\infty} P[\tilde{q}_i = n \mid n > 0] \; z^n \\ &= \sum_{n=1}^{\infty} \left(\frac{\pi_n}{1 - \pi_0} \right) z^n \\ &= \frac{G_Q(z) - \pi_0}{1 - \pi_0} \end{aligned} \tag{4c}$$

$$\begin{aligned} E[z^{\tilde{a}} \mid \tilde{a} > 0] &= \sum_{i=1}^{K} \left[\frac{\binom{K}{i} \beta^i (1-\beta)^{K-i}}{1 - (1-\beta)^K} \right] z^i \\ &= \frac{(1 - \beta + \beta z)^K - (1-\beta)^K}{1 - (1-\beta)^K} \end{aligned} \tag{4d}$$

In steady state, $E[z^{\tilde{q}_{i+1}}] = E[z^{\tilde{q}_i}] = E[z^{\tilde{q}}]$. The generating function $G_Q(z)$ of the number of customers in the input queue can be obtained from equation (4) as:

$$G_Q(z) = E[z^{\tilde{q}}] = \frac{(1-K\beta)(1-z)}{(1-\beta+\beta z)^K - z} \tag{5}$$

The expected number of customers $E[\tilde{q}]$ at the imbedded points is given by:

$$E[\tilde{q}] = \left. \frac{dG_Q(z)}{dz} \right|_{z=1} = \frac{K(K-1)\beta^2}{2(1-K\beta)} \tag{6}$$

Consider the arrival of a tagged packet and let its arrival time be equally probable in any of the K link-slots. Let A_i be the event that the tagged packet arrives in link-slot i. Then the expected number of packets $E[\tilde{L}]$ arrived from the last imbedded point until the arrival of the tagged packet is:

$$\begin{aligned} E[\tilde{L}] &= \sum_{i=1}^{K} E[\tilde{L} \mid A_i] \cdot P[A_i] \\ &= \frac{1}{K} \sum_{i=1}^{K} (i-1)\beta \\ &= \frac{(K-1)\beta}{2} \end{aligned} \tag{7}$$

The number of packets in the queue averaged over a cycle, denoted as $E[\tilde{m}]$, is:

$$E[\tilde{m}] = E[\tilde{q}] + \frac{K\beta}{2} \tag{8}$$

The rate of packet arrival to an input adaptor is $K\beta/N\tau$. Therefore, by Little's formula, the expected delay D is:

$$D = \frac{E[\tilde{m}]}{\lambda} = \frac{(1-\beta)N\tau}{2(1-K\beta)} \tag{9}$$

C. Packet Loss Probability at the Input Adaptor

The M logical queues in each input adaptor share the B_1 buffers by complete sharing strategy which has the best blocking performance [KAMO 80]. When all the buffers are occupied, any incoming packets will be lost. In this section, we derive an approximate expression of P_L as a function of the buffer size B_1. This expression is an upper bound of P_L. Firstly, we derive the probability mass function of the number of packets in the input queue with infinite buffers. Equation (5) can be rewritten as:

$$G_Q(z) = (1 - K\beta)(1 - z)H(z) \tag{10}$$

where

$$H(z) = \sum_{i=0}^{\infty} h_i z^i = \frac{1}{(1 - \beta + \beta z)^K - z} \tag{11}$$

From equation (10), π_i is given by:

$$\pi_i = \begin{cases} \dfrac{-(1 - K\beta)}{(1 - \beta)^K} & i = 0 \\ (1 - K\beta)(h_i - h_{i-1}) & i > 0 \end{cases} \tag{12}$$

where h_i can be evaluated as follow. Differentiate $H(z)$ with respect to z, we have

$$\frac{dH(z)}{dz} = [1 - K\beta(1 - \beta + \beta z)^{K-1}]\, H^2(z)$$

By Leibnitz's rule of differentiation, we have

$$\frac{d^{i+1}H(z)}{dz^{i+1}} = \sum_{l_1+l_2+l_3=i} \binom{i}{l_1, l_2, l_3} \frac{d^{l_1}[1 - K\beta(1 - \beta + \beta z)^{K-1}]}{dz^{l_1}} \frac{d^{l_2}H(z)}{dz^{l_2}} \frac{d^{l_3}H(z)}{dz^{l_3}} \tag{13a}$$

where

$$\frac{d^{l_1}[1-K\beta(1-\beta+\beta z)^{K-1}]}{dz^{l_1}} = \begin{cases} 1-K\beta(1-\beta+\beta z)^{K-1} & l_1=0 \\ -\dfrac{K!\beta^{l_1+1}}{(K-l_1-1)!}(1-\beta+\beta z)^{K-l_1-1} & 0<l_1\le K-1 \\ 0 & l_1>K-1 \end{cases} \qquad (13b)$$

With equation (13), $d^iH(z)/dz^i|_{z=0}$ can be evaluated recursively. h_i can then be found as:

$$h_i = \frac{1}{i!}\frac{d^iH(z)}{dz^i}\bigg|_{z=0}$$

Consider a tagged packet that arrives at the j^{th} link-slot after an imbedded point. The probability $\Lambda(i,j)$ that the number of packets seen by this tagged packet is i is given by:

$$\Lambda(i,j) = \sum_{p=0}^{\min\{i,j-1\}} P[p \text{ packet arrivals in the first } j-1 \text{ link-slots}] \cdot$$

$$P[\text{there are } i-p \text{ packets in the queue at the last imbedded point}]$$

$$= \sum_{p=0}^{\min\{i,j-1\}} \left[\binom{j-1}{p}\beta^p(1-\beta)^{j-1-p}\right]\pi_{i-p}$$

The probability Π_i that there are i packets in an input queue with infinite buffer is given by:

$$\Pi_i = \sum_{j=1}^{K} \text{Prob[the tagged packet sees } i \text{ packets in the input queue} \mid A_j] \cdot \text{Prob}[A_j]$$

$$= \frac{1}{K}\sum_{j=1}^{K}\Lambda(i,j) \qquad (14)$$

When the buffer size B_1 is finite and complete sharing strategy is employed, the M input queues in an input adaptor are not independent. However, for a well designed fast packet switch, the buffer size B_1 is chosen such that the probability of packet loss is very small (say less than 10^{-6}). In this case, the input queues can be approximated to be independent. The probability of packet loss P_L in an input adaptor can be approximated as:

$$P_L = 1 - \sum_{\sum_{j=1}^{M} i_j \le B_1} \Pi_1(i_1)\Pi_2(i_2)\cdots\Pi_M(i_M) \tag{15}$$

D. Probability Mass Functions of the Idle and Busy Periods

In this section, we derive the probability mass functions of the length of the idle and busy periods of the departure process of an input queue. These functions characterize the departure processes of the input queues and are used to generate the arrival process to the output queue in the simulation experiments. Let $\tilde{X}$ and $\tilde{Y}$ be the duration of the idle and busy periods in unit of cycles. Then, $x_i = P[\tilde{X} = i]$ is found to be:

$$\begin{aligned} x_i &= P[\text{no packet arrival for consecutive } i \text{ cycles and a packet arrives} \\ &\qquad \text{at cycle } i+1 \mid i \ge 1] \\ &= \frac{[(1-\beta)^K]^i \, [1-(1-\beta)^K]}{(1-\beta)^K} \end{aligned} \tag{16}$$

$y_i = P[\tilde{Y} = i]$ can be found as follow. An idle period will terminate at the end of a cycle when there is at least one packet arrival in this cycle. Let there be $\tilde{w}$ packet arrivals in this cycle. Each of these $\tilde{w}$ packets generates a sub-busy period [KLEI 74]. $\tilde{Y}$ can then be expressed as:

$$\tilde{Y} = \tilde{S}_1 + \tilde{S}_2 + \cdots + \tilde{S}_{\tilde{w}}$$

where $\tilde{S}_i$'s are i.i.d. random variables and $\tilde{S}_i$ gives the duration of the i^{th} sub-busy period. Since at least one packet is transmitted in each sub-busy period, the sub-busy period $\tilde{S}_1$ is at least equal to one. In any sub-busy period, let $\tilde{v}$ be the number of packet arrivals when the input queue is transmitting the first packet to the output queueing system. Each of these packets will in turn generate a sub-busy period [KLEI 74]. Then $\tilde{S}_1$ can be expressed as:

$$\tilde{S}_1 = 1 + \tilde{s}_1 + \tilde{s}_2 + \cdots + \tilde{s}_{\tilde{v}}$$

where $\tilde{s}_i$'s are i.i.d. random variables having the same distribution as $\tilde{S}_i$'s. Let $G_Y(z)$ and $G_S(z)$ be the probability generating functions of the busy period and sub-busy period respectively. $G_S(z)$ can be found as follow:

$$\begin{aligned} G_S(z) &= E\left[E\left[z^{1+\tilde{s}_1+\tilde{s}_2+\cdots+\tilde{s}_{\tilde{v}}} \mid \tilde{v}\right]\right] \\ &= zE\left[\left(E\left[z^{\tilde{s}_1}\right]\right)^{\tilde{v}}\right] \\ &= zE[(G_S(z))^{\tilde{v}}] \\ &= z\sum_{i=0}^{K}\binom{K}{i}\beta^i(1-\beta)^{K-i}[G_S(z)]^i \\ &= z[1-\beta+\beta G_S(z)]^K \end{aligned}$$

$G_Y(z)$ can then be found as follow:

$$\begin{aligned} G_Y(z) &= E[z^{\tilde{Y}}] \\ &= E\left[E\left[z^{S_1+S_2+\cdots+S_{\tilde{w}}} \mid \tilde{w}\right]\right] \\ &= E\left[\left(E\left[z^{S_1}\right]\right)^{\tilde{w}}\right] \\ &= E[(G_S(z))^{\tilde{w}}] \\ &= \sum_{i=1}^{K}\left[\frac{\binom{K}{i}\beta^i(1-\beta)^{K-i}}{1-(1-\beta)^K}\right][G_S(z)]^i \\ &= \frac{[1-\beta+\beta G_S(z)]^K-(1-\beta)^K}{1-(1-\beta)^K} \\ &= \left[\frac{z^{-1}G_R(z)-(1-\beta)^K}{1-(1-\beta)^K}\right] \end{aligned}$$

where

$$G_R(z) = \sum_{i=1}^{\infty} r_i z^i = z[1-\beta+\beta G_R(z)]^K \tag{18}$$

From equations (17) and (18), y_i can be found to be

$$y_i = \frac{r_{i+1}}{1-(1-\beta)^K}$$

$$= \frac{\frac{1}{(i+1)!} \left. \frac{d^{i+1} G_R(z)}{dz^{i+1}} \right|_{z=0}}{1-(1-\beta)^K} \tag{19}$$

where

$$\left. \frac{d^n G_R(z)}{dz^n} \right|_{z=0} = \sum_{\substack{1+l_2+l_3+\cdots+l_{K+1}=n \\ 0 \le l_2, l_3, \ldots, l_{K+1} \le n-1}} \binom{n}{1, l_2, l_3, \cdots, l_{K+1}} \prod_{i=2}^{K+1} \left. \frac{d^{l_i}}{dz^{l_i}} [1-\beta+\beta G_R(z)] \right|_{z=0} \tag{20}$$

which can be evaluated recursively.

4.5 Numerical Results

Consider a 1024×1024 switch with inputs divided into 8 equal groups. Let the packet transmission time in any input link be normalized to one time unit. We define the *speedup factor* (SF) as the ratio of the sum of the data rates of all the vertical buses to the sum of the data rates of all the input links. SF is therefore equal to M/K.

Fig.12 shows the average queueing delay in the input adaptor. As SF increases, the queueing delay becomes smaller because the input queues are served at a faster rate. At 30% speedup, very small delay is obtained even at $\rho = 0.9$. Fig.13 shows the packet loss probability at the input adaptor for various buffer sizes at $\rho = 0.9$. When SF=1, the required buffer size to achieve a packet loss probability of 10^{-6} is found to be about 150. However, with SF=1.3, the required buffer size is reduced to only 31. In the input queue only packets to a certain destination

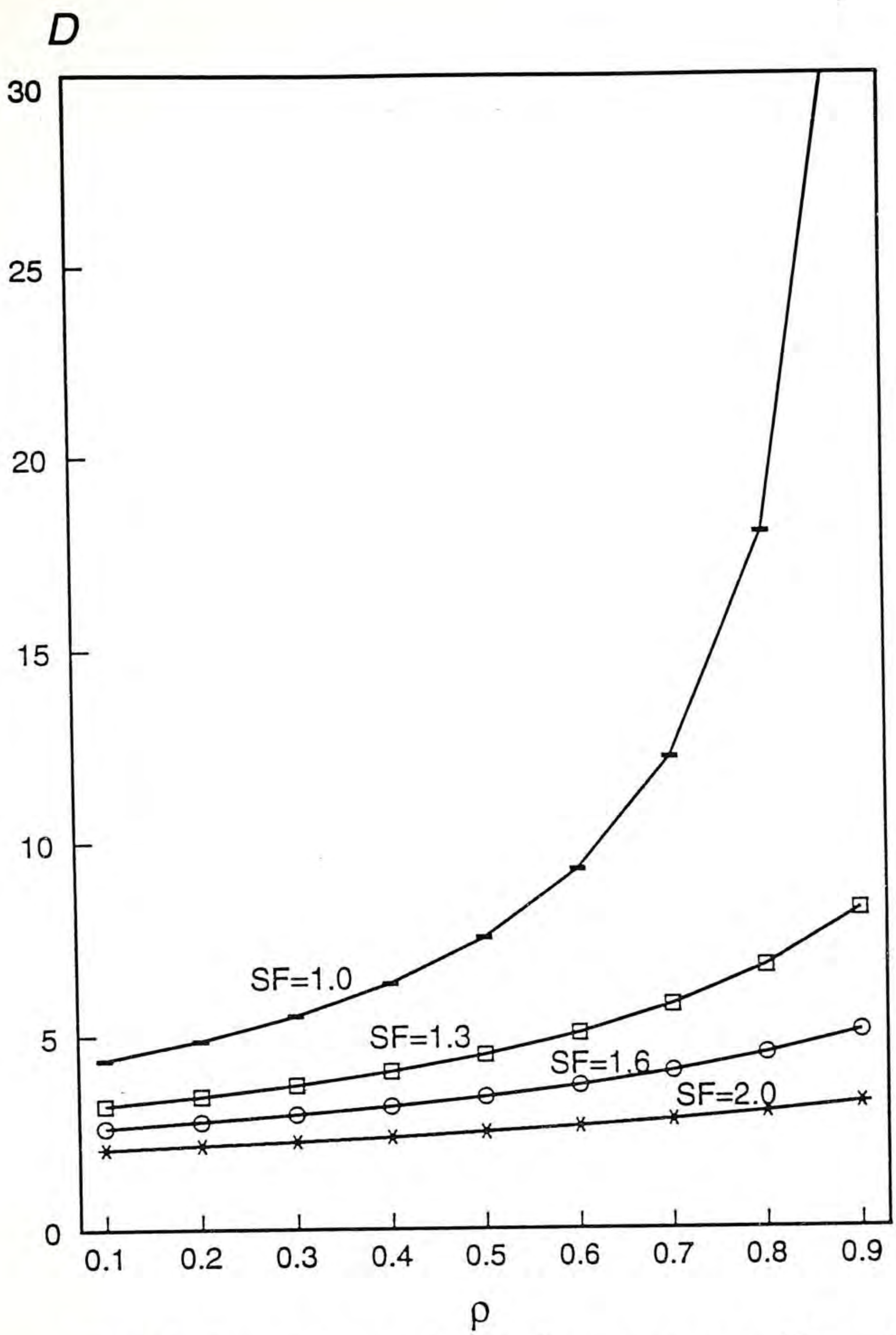


Fig.12: Input queueing delay versus ρ ($B_T=\infty$)

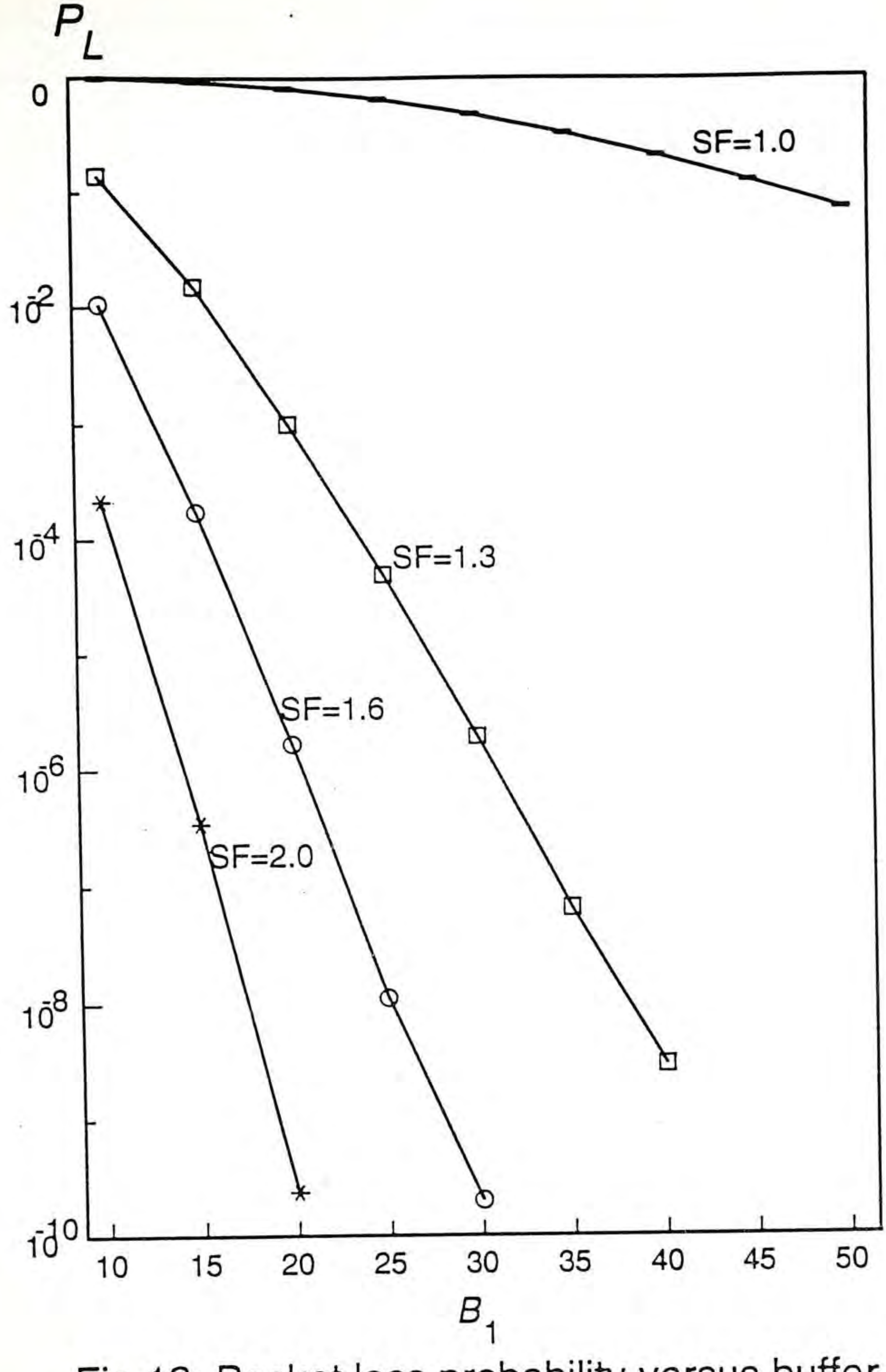


Fig.13: Packet loss probability versus buffer size in input adaptor (ρ=0.9)

can be served at a certain time. As all packet destinations are assumed to be independent and uniformly distributed, this "randomness" makes the speeding-up of the bus rate necessary for satisfactory performance.

Fig.14 shows the average queueing delay in the output adaptor. Here, to the contrary, a larger SF gives larger delay at the output adaptor. The difference, however, is only apparent at ρ very large (a difference of 1 time unit at $\rho = 0.9$). Moreover, all SF≥1.3 cases give almost identical delay characteristics. This phenomenon can be explained as follow. When SF>>1, there is essentially no queueing at the input adaptor. All packets to a certain output port will immediately appear at the output queue. Moreover, the input process to the output queue is a superposition of N Bernoulli processes. For N=1024, that process should be indistinguishable from a Poisson process. Thus the output queue is just a simple M/D/1 queue. In fact, the M/D/1 delay characteristics coincide with the SF=8 curve in Fig.14. What is interesting to note is that for SF=1.0, the delay at the output queue is smaller than that of the M/D/1 queue and for SF=1.3, the delay is essentially that of the M/D/1 queue. Fig.15 shows the packet loss probability versus the buffer size B_2 in the output adaptor when SF=1.3. As can be seen, at $\rho = 0.9$ a packet loss probability of 10^{-4} can be achieved with a buffer size of 30 and 10^{-5} can be achieved with buffer size of 43.

4.6 Discussions

In the above design, the bus bandwidth is allocated to the input adaptors in a fixed cyclic order. Under very heavy traffic, the switch has 100% potential throughput. However, if the traffic is not heavy and is highly asymmetric, then dynamic bandwidth allocation policies can be used to improve the delay performance. In this section, we propose a dynamic bandwidth allocation policy for this purpose.

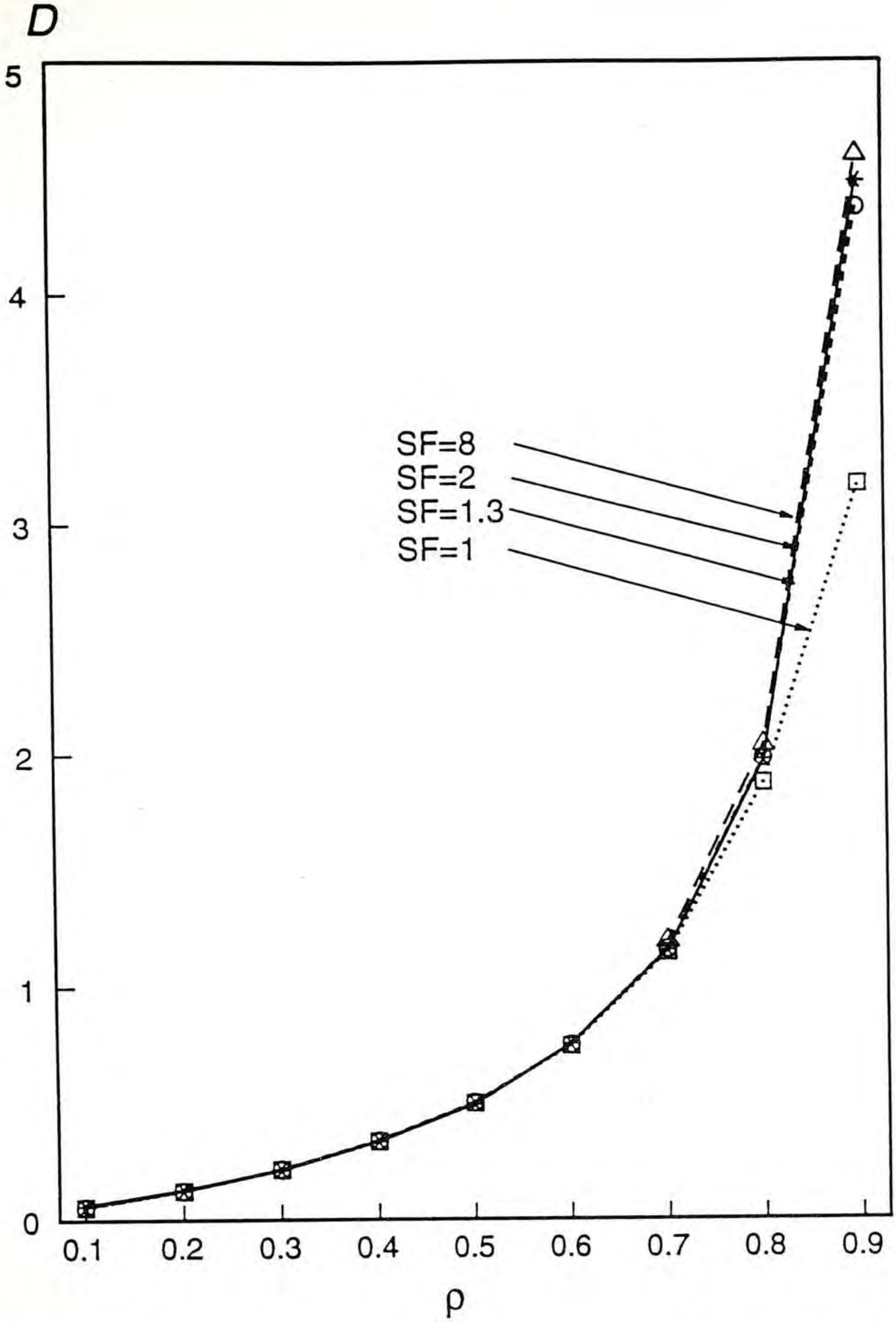


Fig.14: Output queueing delay versus ρ (B_2=40)

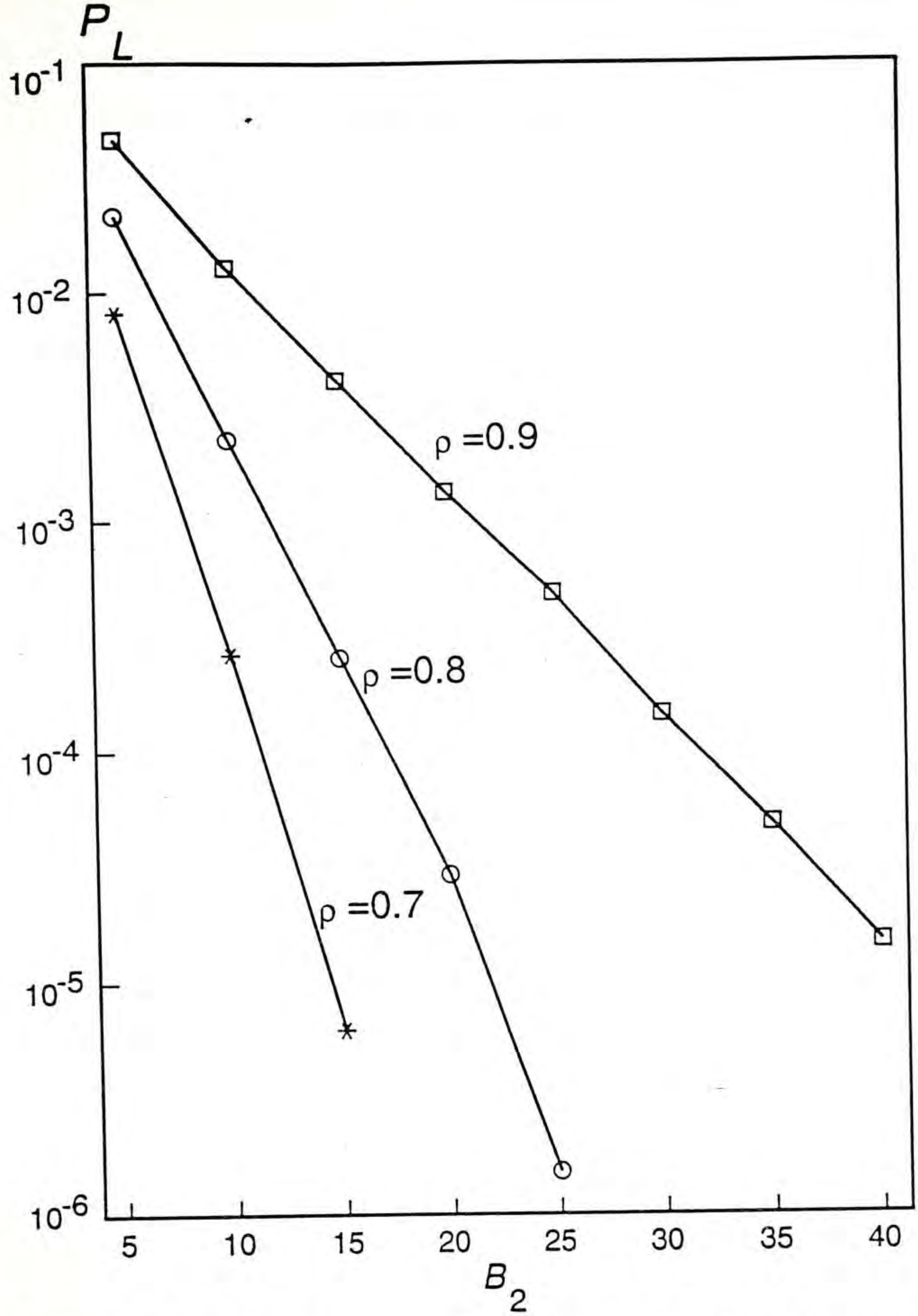


Fig.15: Packet loss probability in output adaptor versus buffer size (SF=1.3)

Transmission time is divided into cycles; each cycle is further divided into M subcycles and each subcycle consists of a fixed number of slots. In the i^{th} subcycle, group i input adaptors are connected to vertical bus $VB_{f(i,j)}$ (where $f(i,j)$ is defined by equation (1)) and share the available bandwidth (or time slots) in $VB_{f(i,j)}$ in this subcycle.

Within each input group, there are M tokens but only one of them is active at a time. Consider any subcycle and let input group i be connected to vertical bus j in this subcycle. Only token j circulates among the group i input adaptors. When an input adaptor in group i seizes token j, then it has a chance to transmit a packet to vertical bus j in the next slot. If it has one or more packets destined for group j outputs, it holds the token until the next slot. In the next slot, it immediately releases token j to its neighbour and starts to transmit a packet. On the other hand, if it has no packet for group j outputs, it immediately passes token j to its neighbour. At the end of this subcycle, the input adaptor holding token j keeps on holding this token for one cycle (i.e. until input group i is connected to vertical buses j again). If input group i is connected to vertical bus k in the next subcycle, then the input adaptor holding token k releases this token and arbitration for vertical bus k starts.

Using this policy, the input adaptors in the same group can share the available bandwidth in each subcycle by round robin policy and hence the delay performance can be improved. Moreover, using this strategy can preserve the nice features of the multibus packet switch (i.e. very simple control circuit within the switch fabric, 100% potential throughput under heavy traffic, internally unbuffered and adding input and output ports without increasing the bus and I/O adaptor speed). One of the possible extensions of the present work is to analyze the performance of this dynamic bandwidth allocation strategy under non-uniform traffic.

4.7 Chapter Summary

There are various approaches to the design of broadband packet switches. Using high speed shared buses, packet switching can be done very simply by individual stations through filtering of unwanted packets. In fact, it is believed that shared medium based packet switches are potential candidates for future broadband switching. In this chapter, we designed and analyzed a TDM-based multibus switch. In addition to modular growth, this design has the advantages of internally unbuffered, very simple control circuit and 100% throughput under heavy traffic. For satisfactory performance, the buses need to speedup 30% relative to the aggregate input ports rate.

Chapter 5
A Modular Shared Media Video Switch

In chapter 4, we proposed a TDM based multibus packet switch where the inputs and outputs are grouped for buses sharing and packets are switched from the inputs to the outputs using a $M \times M$ bus matrix without memory. Based on this multibus design, we propose in this chapter a switch architecture for distributing broadcast and switched videos. The switch consists of a set of concentration buses (or input buses), a TDM-based bus matrix and a set of distribution buses (or output buses). Dedicated time slots in a frame are reserved for the broadcast videos. The remaining time slots are allocated to the switched videos on a first-come-first-serve basis. Videos are switched via time slot assignments which determine the connections within the bus matrix. Two slot assignment algorithms are designed, one for point-to-point transmissions and the other for point-to-multipoint transmissions. This architecture has three advantages. First, multirate video channels can be accommodated. This can accommodate a variety of video services that have different bit rate requirements. Second, videos can easily be broadcast or multicast to the customers through the shared media. Hence, multipoint communication services (e.g. videoconferencing) can be provided. Third, it can be used as a building block for constructing large video distribution networks.

5.1 System Architecture and Operation

Fig.1 shows a video distribution network serving a population of users. It consists of a *video multiplexer*, a *video switch*, a *slot assignment processor* and a set of *output buses* where video customers are connected to. We assume that each video source has a fixed bit rate. For example, compressed HDTV video without degradation has a bit rate of 44 Mb/s [TOBA 91] and compressed NTSC video with VCR quality has a bit rate of 1.5 Mb/s [TOBA 91]. Let all

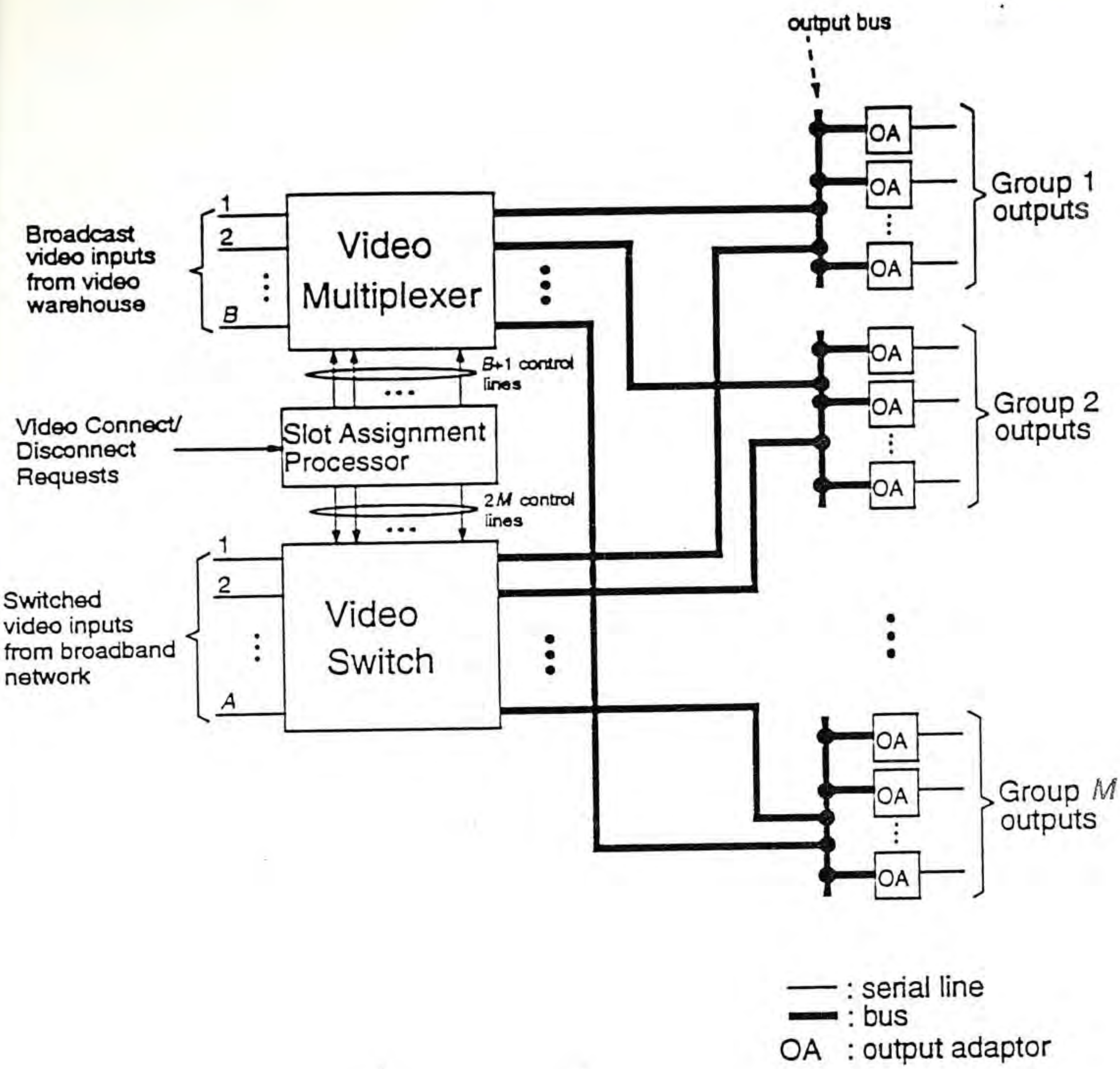


Fig.1: A shared media video distribution network

videos be packetized into the same fixed size packets before entering the network. The transmission of videos in the network occurs in fixed size *frames*. A frame is further divided into g slots of ***broadcast video subframe*** and h slots of ***switched video subframe***. Each slot can accommodate one video packet. Fig.2 shows the framing structure used in the network. Video sources can operate on multiple rates. The basic rate in this network is one packet per frame. A higher rate video can be accommodated by assigning multiple slots per frame for its use. In the rest of this chapter, the slot positions for broadcast videos and switched videos are in reference to the broadcast video subframe and the switched video subframe respectively. In the following, we describe the functions of each component of the video distribution network in details.

A. Video Multiplexer

The video multiplexer combines up to B broadcast videos by time division multiplexing and feeds them onto the M output buses. Fig.3 shows the block diagram of a video multiplexer. It consists of B *broadcast video adaptors* and M *bus relays*. Every broadcast video is connected to a broadcast video adaptor, which converts the serial input into parallel bit streams, stores the packets in the buffer and transmits a packet when its control input is HIGH (i.e. "1"). The bus relay is a high input impedance two-states device which copies the content in the input bus to the output bus when its control input is HIGH and disconnects the output bus from the input bus when its control input is LOW. Because these relays have high input impedance, they can ensure a certain fanout at the output buses.

The B+1 control lines from the slot assignment processor are divided into B transmission enable lines and 1 relay enable line. Each transmission enable line is connected to the control input of a broadcast video adaptor and the relay enable line is connected to the control inputs of all the bus relays. In the broadcast video subframe, the relay enable line is HIGH so that all the broadcast video adaptors can transmit to all the output buses. Let the i^{th} broadcast video

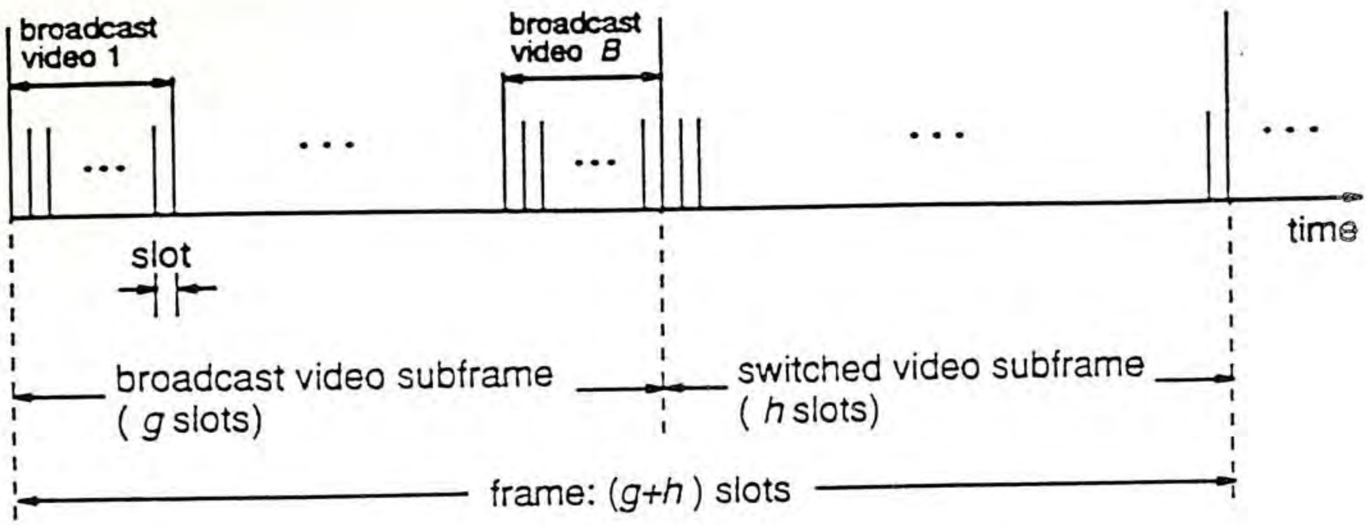


Fig.2: Transmission frame

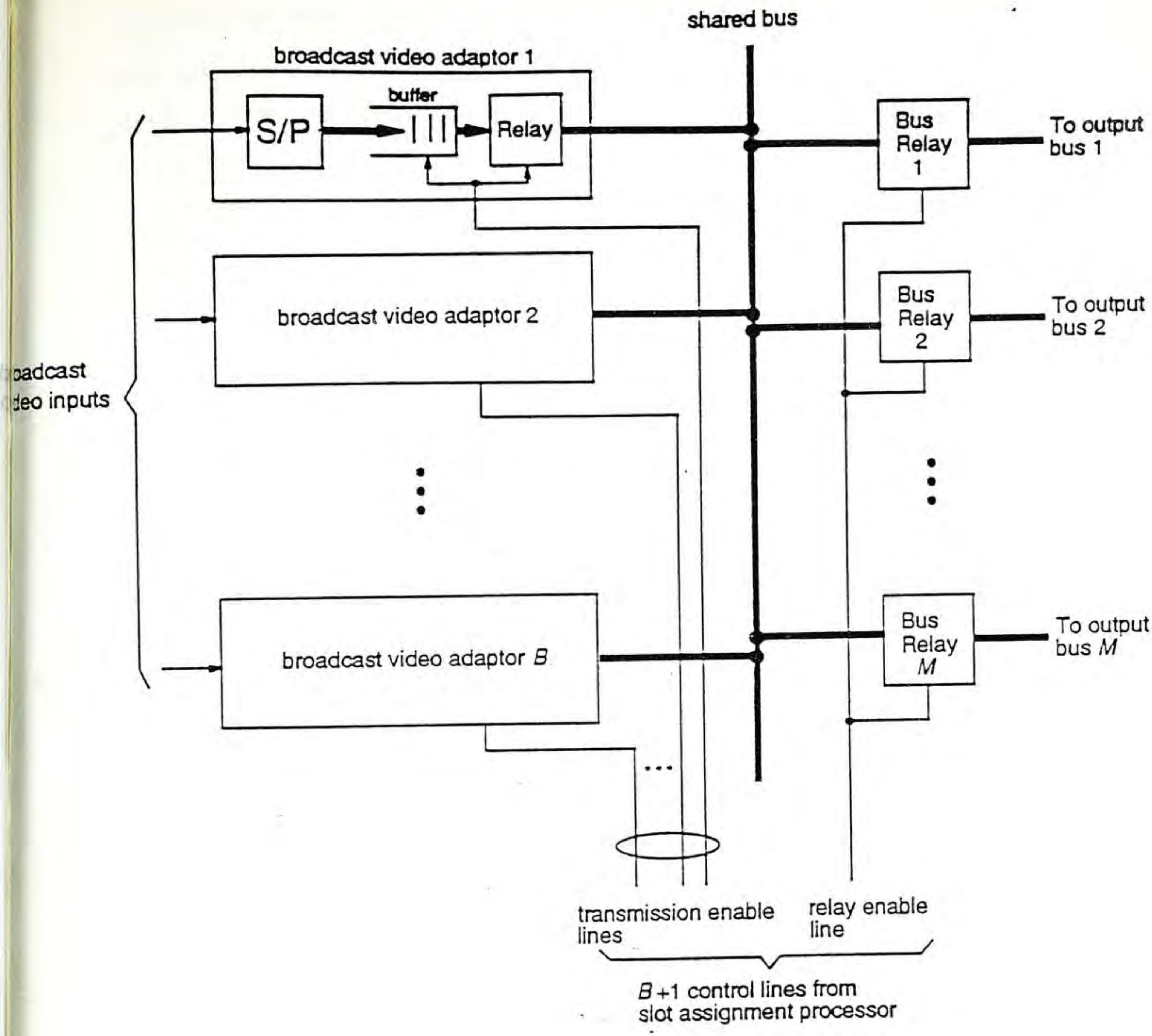


Fig.3: Video multiplexer

require b_i slots per frame. In the first b_1 slots of every broadcast video subframe, the 1^{st} transmission enable line is HIGH to enable broadcast video adaptor 1 to transmit packets to the output buses. Similarly, in slots b_1+1 to b_1+b_2, the 2^{nd} transmission enable line is HIGH, and etc.

B. Video Switch

Fig.4 shows the video switch. The switched video inputs are partitioned into M groups of L video inputs each. Every switched video is connected to a *switched video adaptor* via an input port. Group i adaptors are connected to the i^{th} horizontal bus, and the j^{th} vertical bus is connected to group j outputs. The M horizontal buses are connected to the M vertical buses in bus matrix form, with a total of M^2 *switching elements* at the crosspoints of the vertical and horizontal buses. The switching element placed at the crosspoint of the i^{th} horizontal bus and the j^{th} vertical bus is identified as $S_{i,j}$ and it can connect the i^{th} horizontal bus to the j^{th} vertical bus. A common clock line is connected to all the switched video adaptors and switching elements (this is not shown in Fig.4 for clarity). Videos are switched to the outputs via time slot assignment that controls the connections of all the switching elements. The slot assignment problems are formulated and solved in sections 5.2 and 5.3.

When the current slot assignments are changed, the slot assignment processor sends the new slot assignments at the beginning of a frame to all the switched video adaptors and switching elements through $2M$ control lines (Fig.4). These new slot assignments will be executed in the next frame. Fig.5(a) shows the slot assignment signal unit format for group i switched video adaptors. w_{ij} is a bit map of the slot positions assigned to the j^{th} input port of group i. Thus if h=10 and w_{34}=0001001000, the 4^{th} input port of group 3 can transmit at slots 4 and 7. When new slot assignments arrive at a switched video adaptor, it removes the first h bits and passes the remaining bits to the next switched video adaptor. Fig.5(b) shows the slot assignment signal unit format for the switching elements on the i^{th} horizontal bus. $t_{ij}^{(k)}$ consists of 1 bit and $t_{ij}^{(k)} = 1$

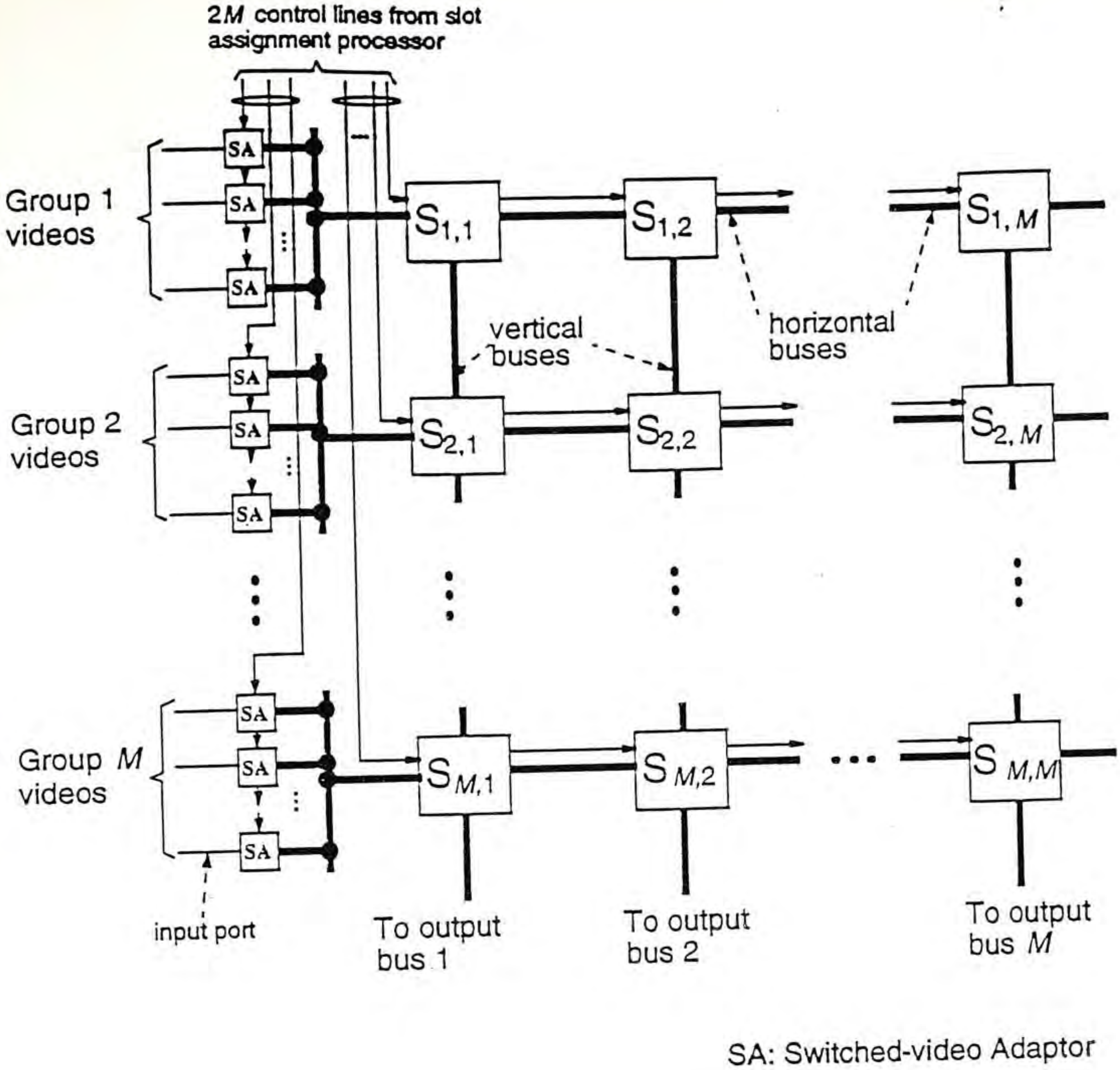
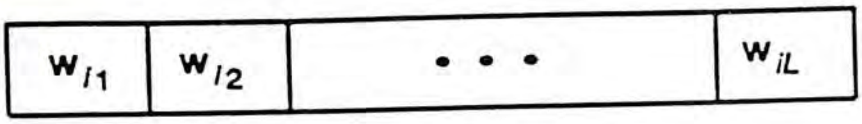
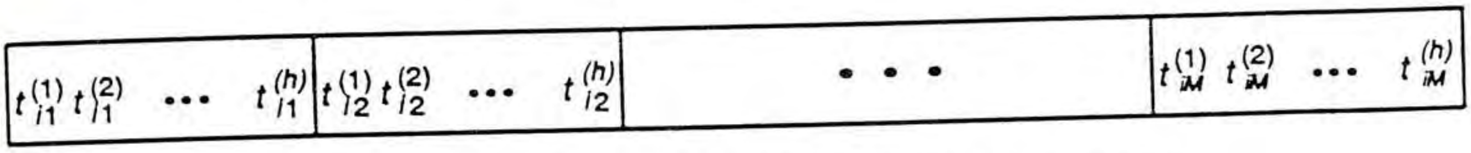


Fig.4: Video switch



(a) For group *i* input adaptors



(b) For switching elements on the i-th horizontal bus

Fig.5: Slot assignment signal unit formats

if S_{ij} connects the i^{th} horizontal bus to the j^{th} vertical bus in slot k and $t_{ij}^{(k)} = 0$ otherwise. When new slot assignments arrive at a switching element, it removes the first h bits and passes the remaining bits to the next switching elements. w_{ij}'s and $t_{ij}^{(k)}$'s are determined in section 5.2 and 5.3.

Fig.6 shows a switched video adaptor. It converts the serial input into parallel bit streams and stores the packets in the buffer. The controller receives slot assignments, removes and stores the first h bits and passes the remaining bits to the next switched video adaptor, and initiates packet transmissions in the slots assigned to it.

Fig.7 shows the realization of a switching element. The relay is a high input impedance two-states element. When its control input is HIGH, it copies the content in the input to the output; when its control input is LOW, it disconnects the output from the input. The high input impedance of these relays can ensure a certain a certain fanout at the output buses. The shift register stores a set of bits that determines when to connect the horizontal bus to the vertical bus. When a clock pulse arrives, all the bits shift to the left by one position and the least significant bit is shifted to the most significant bit position. If the current least significant bit is HIGH, the relays copies the content in the horizontal bus to the vertical bus. When new slot assignments arrives, the receiver extracts the first h bits, stores them in the register and passes the remaining bits to the next switching elements. At the end of the current frame, the receiver enables the relays connecting the two registers, allowing the updating of slot assignments in the shift register.

C. Slot Assignment Processor

The slot assignment processor records the number of slots required by every broadcast and switched video and records the output group numbers of the destinations of the switched videos. It coordinates broadcast video distribution through the B+1 control lines connecting it and the video multiplexer. When a customer requests a new switched video session, the slot

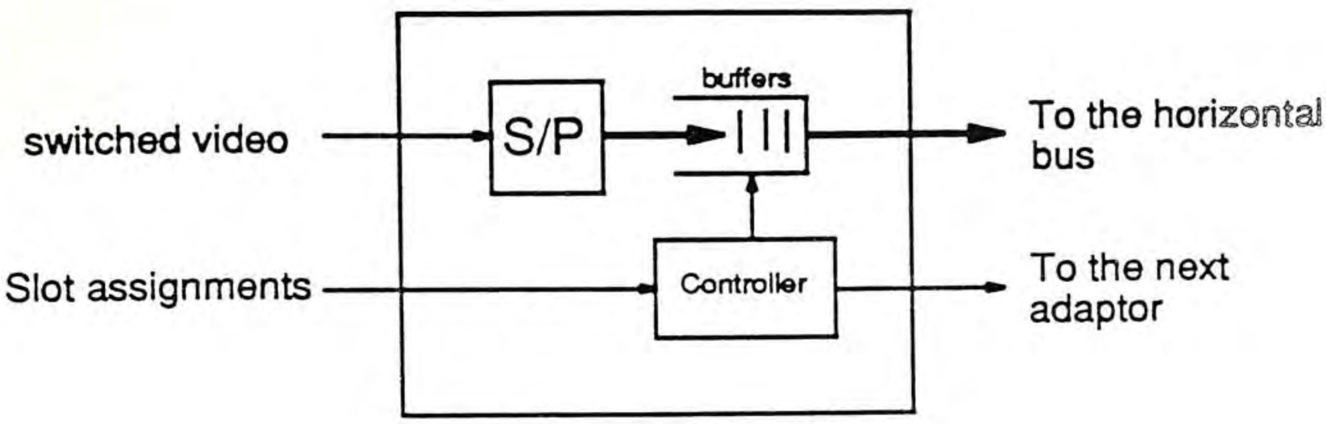


Fig.6: Switched video adaptor

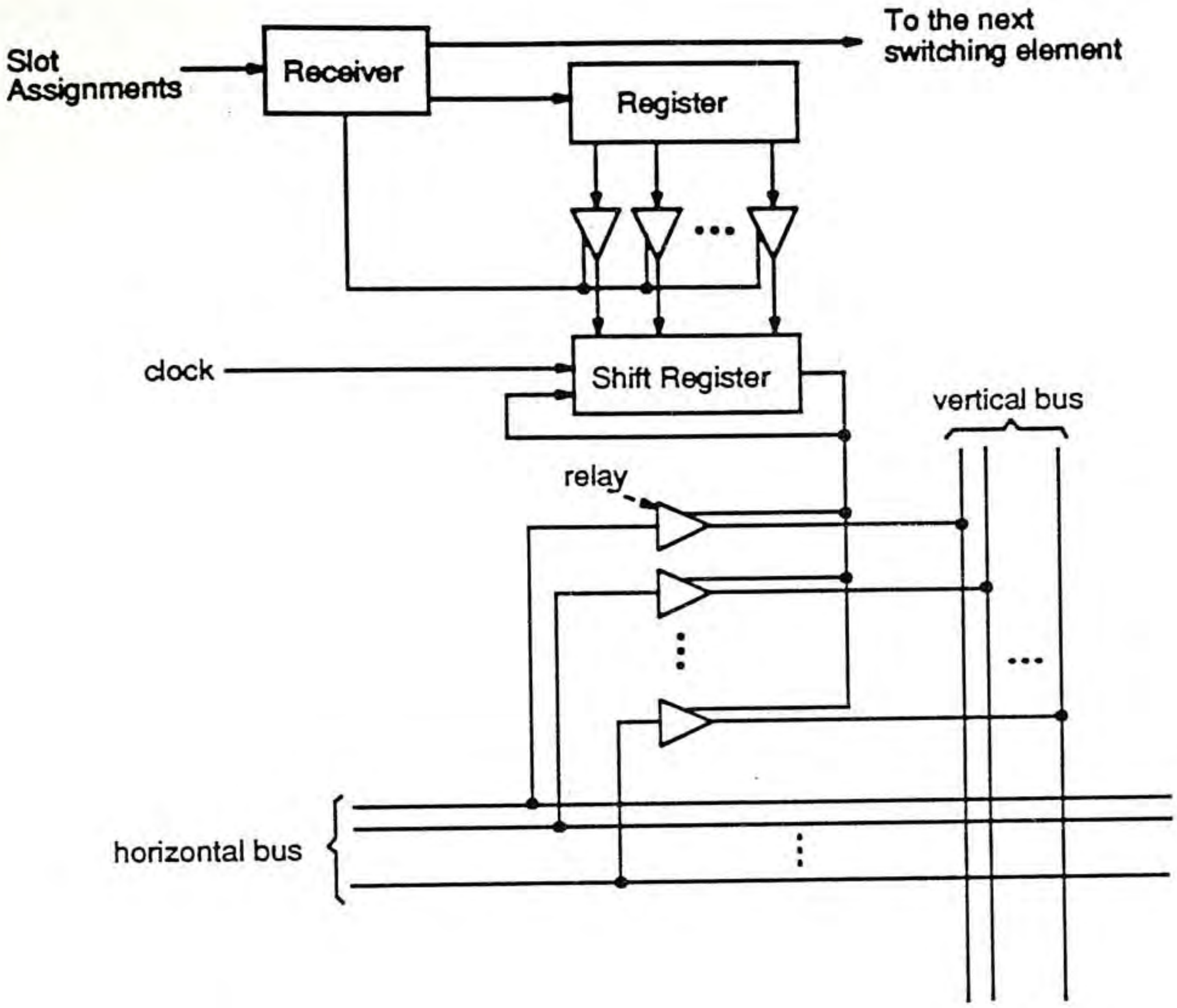


Fig.7: Switching element

assignment processor determines whether it can be admitted. If this new session can be admitted, it sends the new slot assignments to the video switch through the $2M$ control lines connecting it and the video switch.

D. Output Adaptor

Fig.8 shows the block diagram of an output adaptor. The bus receiver is a high input impedance device and it copies the content in its input to its output. The packet filter inspects the circuit number contained in the header of every incoming packet and accepts packets destined for it. The received packets are stored in the buffer and then converted into serial bit streams for transmission.

The fanout problem [STRA 84] limits the maximum number of output adaptors on an output bus. To relieve, the output bus can first be connected to a set of bus receivers as shown in Fig.9, and each bus receiver drives a subgroup of output adaptors in an output group.

E. Typical Circuit Board Layout

The entire distribution network can be packed on a single circuit board as shown in Fig.10. In one corner of the board, the $M \times M$ bus matrix is located. The switched video adaptor boards are inserted into connection slots attaching to the input of the bus matrix, the output adaptor boards are inserted into connection slots attaching to the outputs of the bus matrix, and the video multiplexer board is inserted into a connection slot attaching to the output adaptor board. The slot assignment processor and the clock distribution network (not shown) are placed on the back of the circuit board.

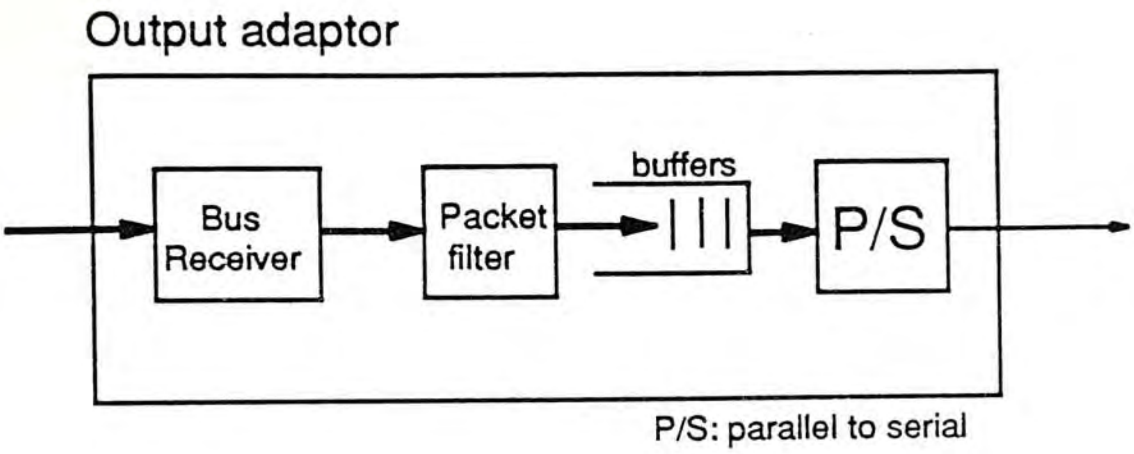


Fig.8: Output adaptor

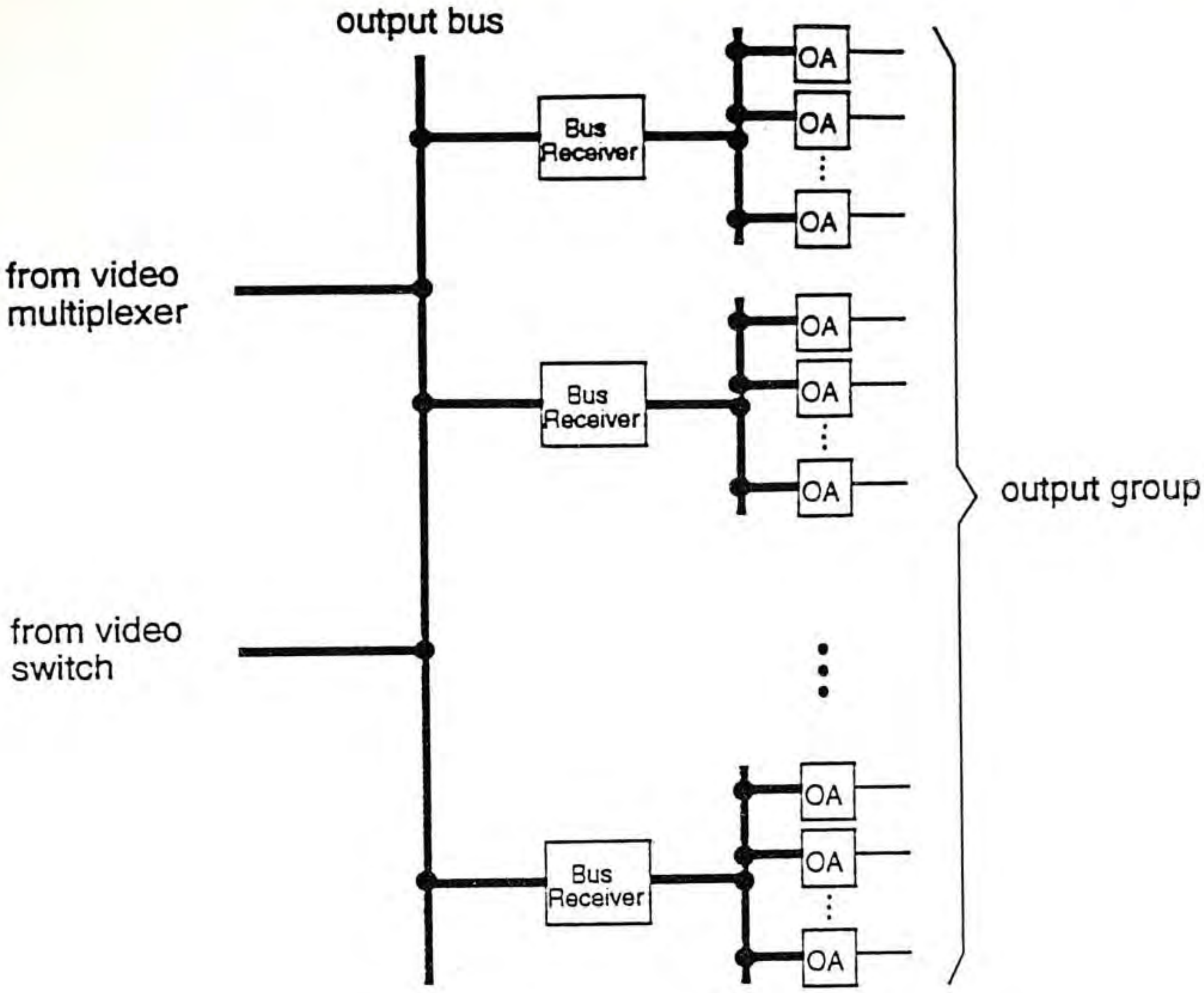


Fig.9: Method to increase the output fanout

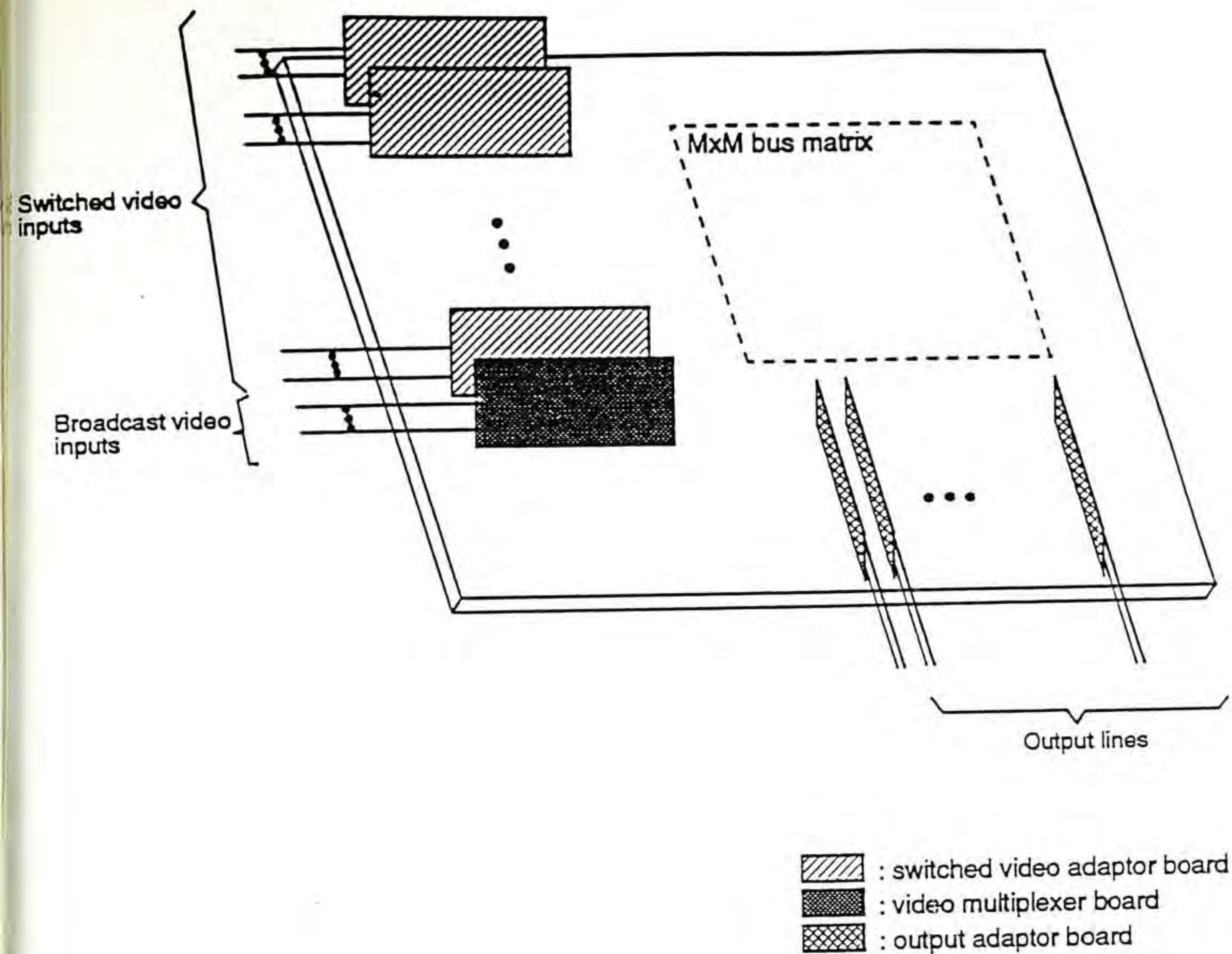


Fig.10: Compact arrangement of the four types of circuit boards

5.2 Slot Assignment for Point-to-Point Transmissions

In this section, we consider the case where every switched video packet is sent to one output group.

A. Data Structure

Let $\gamma(i,j;d)$ be the number of packets sent from the j^{th} input port in group i to group d outputs in each frame. The total number of packets sent from group i video to group d outputs is $c_{id} = \sum_{j=1}^{L} \gamma(i,j;d)$. We call the c_{id}'s put in matrix form a *group traffic matrix* C. Since only h slots per frame are available for switched videos, all the row sums and column sums of C cannot be larger than h. The m^{th} slot *transmission matrix* $T_m = [t_{id}^{(m)}]_{M\times M}$ describes the transmitting and receiving group pairs in slot m. Specifically, $t_{id}^{(m)} = 1$ indicates that group i videos can transmit a packet to group d outputs in slot m and $t_{id}^{(m)} = 0$ otherwise. Since each vertical or horizontal bus can transmit at most one packet in a slot, all the column sums and row sums of T_m cannot be larger than one.

B. Slot Assignment Algorithm

The slot assignment problem is to determine $\{T_i\}$ such that $T_1+T_2+\cdots T_h$=C. A related System of Distinct Representative problem [INUK 79] is to find a transmission matrix with a maximum number of non-zero elements or maximum rank. Its solution requires backtracking. The slot assignment problem studied here, however, does not require the transmission matrices to have maximum rank. Another related problem considered by Bonnccelli [BONU 89] is to find a minimum set of T_i's such that their sum is equal to C. In the slot assignment problem, h is a given parameter.

The following greedy algorithm called SA/PP (Slot Assignment/Point-to-Point traffic) computes T_1, T_2, $\cdots$ in turns. In the computation of T_1, let matrix B be a running "matrix variable" recording the part of C still to be processed. For the computation of T_2, B is first updated by subtracting T_1 from C. For T_3, B starts from C-T_1-T_2, etc. Let r_i and c_i be the number of non-zero elements in row i and column i of B respectively, and let $\boldsymbol{r} = [r_1 \quad r_2 \quad \cdots \quad r_M]$ and $\boldsymbol{c} = [c_1 \quad c_2 \quad \cdots \quad c_M]$. The idea of the SA/PP Algorithm is based on the fact that in any iteration, group i videos can be connected to one of the r_i possible output groups, and output group j can be connected to one of the c_j video groups. If the number of non-zero elements is smallest at row i or group i videos has the smallest number of connection choices, then we should connect group i videos to one of the r_i output groups specified by B in the present slot. Similarly, if the number of non-zero elements is smallest at column j or output group j has the smallest number of connection choices, then we should connect output group j to one of the c_j video groups specified by B in the present slot. When group i videos is connected to group j outputs in a slot, the former cannot be connected to other output groups and the latter cannot accept packet from other video groups in the present slot. Hence, the elements in row i and column j of B are set to zero. We prove in the appendix that SA/PP algorithm is efficient. The SA/PP algorithm given below has two loops. With each inner loop, a non-zero element of a transmission matrix is determined. A proper transmission matrix is found with the completion of each outer loop.

SA/PP Algorithm

$$\begin{bmatrix} \text{Inputs:} & M;\, h;\, \mathrm{C} \\ \text{Outputs:} & \mathrm{T}_1, \mathrm{T}_2, \cdots, \mathrm{T}_h \end{bmatrix}$$

1. FOR m=1 TO h DO
2. $\mathrm{T}_m \leftarrow 0$; $\mathrm{B} \leftarrow \mathrm{C}$;
3. FOR n=1 TO M DO
4. compute arrays $\boldsymbol{r}$ and $\boldsymbol{c}$ from B;
5. find the smallest non-zero entry r_p in $\boldsymbol{r}$ and the smallest non-zero entry c_q in $\boldsymbol{c}$;
6. IF $r_p \le c_q$ THEN select a non-zero element in row p of B and denote its column number as i; $t_{pi}^{(m)} \leftarrow 1$; and set all elements in row p and column i of B to zero;

   ELSE select a non-zero element in column q of B and denote its row number as i; $t_{iq}^{(m)} \leftarrow 1$; and set all elements in row i and column q of B to zero;

7. ENDFOR;

8. $\mathbf{C} \leftarrow \mathbf{C} - \mathbf{T}_m$;

9. ENDFOR.

The inner loop is executed M times, each with $O(M)$ steps. The outer loop is executed h times. Therefore, the time complexity of the SA/PP Algorithm is $O(hM^2)$.

Example 1

Let M=3, h=4 and the group traffic matrix $\mathbf{C}$ be:

$$\mathbf{C} = \begin{bmatrix} 2 & 0 & 2 \\ 1 & 1 & 2 \\ 1 & 3 & 0 \end{bmatrix}$$

The details of the first step is given below (the numbers below matrix $\mathbf{B}$ are the c_i's and the numbers to the right of $\mathbf{B}$ are the r_i's):

$$n = 1: \quad \mathbf{B} = \mathbf{C} = \begin{array}{cc} \begin{bmatrix} 2 & 0 & 2 \\ 1 & 1 & 2 \\ 1 & 3 & 0 \end{bmatrix} & \begin{matrix} 2 \\ 3 \\ 2 \end{matrix} \\ \begin{matrix} 3 & 2 & 2 \end{matrix} & \end{array} \quad \Rightarrow \quad \mathbf{T}_1 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$n = 2: \quad \mathbf{B} = \begin{array}{cc} \begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 3 & 0 \end{bmatrix} & \begin{matrix} 0 \\ 2 \\ 2 \end{matrix} \\ \begin{matrix} 2 & 2 & 0 \end{matrix} & \end{array} \quad \Rightarrow \quad \mathbf{T}_1 = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$n = 3: \quad \mathbf{B} = \begin{array}{cc} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 3 & 0 \end{bmatrix} & \begin{matrix} 0 \\ 0 \\ 1 \end{matrix} \\ \begin{matrix} 0 & 1 & 0 \end{matrix} & \end{array} \quad \Rightarrow \quad \mathbf{T}_1 = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

For the computation of $\mathbf{T}_2$, we start with $\mathbf{B} = \mathbf{C} - \mathbf{T}_1$. Following the same procedure, we obtain

$$\mathbf{T}_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad \mathbf{T}_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad \mathbf{T}_4 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

C. Slot Assignments

The slot assignments w_{ij}'s sent to the switched video adaptors can be found by matching the destination output group number of every switched video and the non-zero elements of the transmission matrices:

$$\begin{bmatrix} \text{Inputs:} & M; h; \mathrm{T}_1, \mathrm{T}_2, \cdots, \mathrm{T}_h \\ \text{Outputs:} & w_{ij} \text{ for } i = 1,2,\cdots,M \text{ and } j = 1,2,\cdots,L \end{bmatrix}$$

1. set all bits in w_{ij} to zero ($i = 1,2,\cdots,M$ and $j = 1,2,\cdots,L$);
2. FOR i=1 TO M, m=1 TO h and d=1 TO M DO
3. IF $t_{id}^{(m)} = 1$ and $\gamma(i,j;d) > 0$ for some $j \in \{1,2,\cdots,L\}$ THEN
4. set the m^{th} bit of w_{ij} to one;
5. $\gamma(i,j;d) \leftarrow \gamma(i,j;d) - 1$;
6. ENDIF;
7. ENDFOR.

5.3 Slot Assignment for Point-to-Multipoint Transmissions

In this section, we consider the general case where a switched video packet can be multicast to multiple output groups.

A. Data Structure

Let group i videos generate a total of v_i packets per frame. Label these v_i packets as packet 1, packet 2, $\cdots$, and packet v_i. The *group i traffic matrix* $\mathrm{C}_i=[c_{jd}^i]_{v_i \times M}$ is defined such that c_{jd}^i=1 if packet j in group i is sent to output group d and c_{jd}^i=0 otherwise. The j^{th} row sum of C_i gives the number of destined output groups of packet j in group i and the d^{th} column sum of C_i gives the total number of packets sent from group i videos to group d outputs.

The row sum of any transmission matrix (defined in Section 5.2) is now larger than one for multicast packets. Let α_{ij} denote the input port number of packet j in group i and let $r_i = [r_i^{(1)} \quad r_i^{(2)} \quad \cdots \quad r_i^{(m)}]$ $(i = 1,2,\cdots,M)$ be the *slot assignment array* of group i, where $r_i^{(1)}, r_i^{(2)}$, $\cdots$ denote the input port number in group i that can transmit a packet in slot 1, slot 2, $\cdots$.

B. Slot Assignment Algorithm

The slot assignment problem is to determine the transmission matrices such that $\sum_{m=1}^{h} t_{id}^{(m)}$ is equal to the d^{th} column sum of C_i for all $1 \le i, d \le M$. The following algorithm called SA/PM (Slot Assignment/Point-to-Multipoint traffic) can be used to find $\{T_m\}$ one by one starting from T_1. Consider the computation of the m^{th} slot transmission matrix T_m, which starts as a zero matrix. Let $x = [x_1 \quad x_2 \quad \cdots \quad x_M]$ be an array where x_i=1 indicates that group i videos can send a packet in slot m and x_i=0 otherwise, and let $y = [y_1 \quad y_2 \quad \cdots \quad y_M]$ be an array where y_i=1 indicates that output group i can receive a packet in slot m and y_i=0 otherwise. If $x_1 \vee x_2 \vee \cdots \vee x_M = 1$ and $y_1 \vee y_2 \vee \cdots \vee y_M = 1$, then some packets can be sent from the input groups to the free output groups in slot m. Let g_i be the input group with $x_{g_i} = 1$ and has a set of packets H_i that can complete the multicast in slot m. Let $G=\{g_i\}$.

T_m is found in the following steps. Compute the sets G and H_i for all $i \in G$. If G is not empty (i.e. at least one packet can be sent to all of its destinations in slot m), select C_i (where $i \in G$) such that, among the matrices C_p for all $p \in G$, C_i has the largest number of non-zero rows. Then select row j from C_i (where $j \in H_i$) such that the j^{th} row sum of C_i is the largest. The reasons for this choice are: (1) Group i videos has the largest number of outstanding packets; and (2) Packet j in group i has the largest multicast multiplicity and can complete the multicast in one slot. On the other hand, if G is empty, select C_i with x_i=1 and C_i has the largest number of non-zero rows. Then select packet j from group i such that packet j can be sent to the largest number of free output groups in slot m. Let D be a set of output groups that packet j in group i

can be sent to in slot m. Packet j in group i is then assigned to transmit in slot m, in other words, set $t_{jd}^{(m)} = 1$ for all $d \in D$. Next, set x_i=0 to indicate that group i videos cannot send other packet in slot m and set y_d=0 for all $d \in D$ to indicate that output group d for all $d \in D$ cannot receive other packet in slot m. Repeat the above steps until $x_1 \vee x_2 \vee \cdots x_M = 0$ or $y_1 \vee y_2 \vee \cdots \vee y_M = 0$ to completely determine $\mathbf{T}_m$. When $\mathbf{T}_1, \mathbf{T}_2, \cdots, \mathbf{T}_h$ are found with some $\mathbf{C}_i \neq 0$, then the call request cannot be accommodated by the current transmission schedule and is blocked. If computation time allows, backtracking can be used to search for other transmission schedules. However, our computational experience reveals that backtracking is seldom needed except when both v_i's and the total number of packets sent to each output group are nearly equal to h (i.e. under very heavy traffic load). The SA/PM algorithm is given below:

<u>SA/PM Algorithm</u>

$$\begin{bmatrix} \text{Inputs:} & h; M; v_i \text{ and } \mathrm{C}_i \text{ for } i = 1,2,\cdots,M; \alpha_{ij} \text{ for } i = 1,2,\cdots,M \text{ and } j = 1,2,\cdots v_i \\ \text{Outputs:} & \mathrm{T}_m \text{ for } m = 1,2,\cdots,h; \quad r_i \text{ for } i = 1,2,\cdots,M \end{bmatrix}$$

1. FOR m=1 TO h DO
2. $\mathrm{T}_m \leftarrow 0$; $r_i^{(m)} \leftarrow 0$ for $i = 1,2,\cdots,M$;
3. IF group i videos ($i = 1,2,\cdots,M$) has at least one outstanding packet THEN $x_i \leftarrow 1$ ELSE $x_i \leftarrow 0$;
4. IF at least one packet is sent to output group d ($d = 1,2,\cdots,M$) THEN $y_d \leftarrow 1$ ELSE $y_d \leftarrow 0$;
5. WHILE $x_1 \vee x_2 \vee \cdots \vee x_M = 1$ and $y_1 \vee y_2 \vee \cdots \vee y_M = 1$ DO
6. compute G; compute H_i for all $i \in G$;
7. IF G is not empty THEN
   among the matrices C_p for all $p \in G$, select C_i ($i \in G$) such that C_i has the largest number of non-zero rows; and select row j from C_i such that the j^{th} row sum of C_i is the largest
   ELSE select C_i with x_i=1 such that C_i has the largest number of non-zero rows; and select row j from C_i such that packet j of group i can be sent to the largest number of free output groups;
8. $x_i \leftarrow 0$;
9. IF $y_d \wedge c_{jd}^i = 1$ for $d = 1,2,\cdots,M$ THEN

$t_{id}^{(m)} \leftarrow 1;\ y_d \leftarrow 0;\ c_{jd}^{i} \leftarrow 0;\ r_i^{(m)} \leftarrow \alpha_{ij};$

10. ENDIF.

11. ENDWHILE;

12. ENDFOR;

The WHILE loop is executed at most M times, each with $O(LM)$ steps. The FOR loop is executed h times, each with $O(LM^2)$ steps. Therefore, the time complexity of the SA/PM algorithm is $O(hLM^2)$.

Example 2

Let M=4, h=4 and $\{C_i\}$ be:

$$C_1=\begin{bmatrix}0&1&1&0\\0&0&1&0\\0&1&0&0\end{bmatrix}\quad C_2=\begin{bmatrix}1&0&1&1\\1&0&0&0\end{bmatrix}\quad C_3=\begin{bmatrix}1&1&0&0\\0&0&0&1\\0&0&1&1\end{bmatrix}\quad C_4=\begin{bmatrix}1&1&0&0\\0&0&0&1\end{bmatrix}$$

The details of determining T_1 is given below:

Iteration 1:

$$C_1=\begin{bmatrix}0&1&1&0\\0&0&1&0\\0&1&0&0\end{bmatrix}\quad C_2=\begin{bmatrix}1&0&1&1\\1&0&0&0\end{bmatrix}\quad C_3=\begin{bmatrix}1&1&0&0\\0&0&0&1\\0&0&1&1\end{bmatrix}\quad C_4=\begin{bmatrix}1&1&0&0\\0&0&0&1\end{bmatrix}$$

$$y=[0\quad 0\quad 0\quad 0]\quad\Rightarrow\quad T_1=\begin{bmatrix}0&1&1&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\end{bmatrix}$$

Iteration 2:

$$C_2=\begin{bmatrix}1&0&1&1\\1&0&0&0\end{bmatrix}\quad C_3=\begin{bmatrix}1&1&0&0\\0&0&0&1\\0&0&1&1\end{bmatrix}\quad C_4=\begin{bmatrix}1&1&0&0\\0&0&0&1\end{bmatrix}\quad y=[0\quad 1\quad 1\quad 0]$$

$$\Rightarrow\quad T_1=\begin{bmatrix}0&1&1&0\\1&0&0&0\\0&0&0&0\\0&0&0&0\end{bmatrix}$$

Iteration 3:

$$\mathbf{C}_3=\begin{bmatrix}1&1&0&0\\0&0&0&1\\0&0&1&1\end{bmatrix}\quad \mathbf{C}_4=\begin{bmatrix}1&1&0&0\\0&0&0&1\end{bmatrix}\quad \mathbf{y}=[1\quad 1\quad 1\quad 0]\ \Rightarrow\ \mathbf{T}_1=\begin{bmatrix}0&1&1&0\\1&0&0&0\\0&0&0&1\\0&0&0&0\end{bmatrix}$$

Following the same procedure, $\mathbf{T}_2$, $\mathbf{T}_3$ and $\mathbf{T}_4$ are found to be:

$$\mathbf{T}_2=\begin{bmatrix}0&0&1&0\\0&0&0&0\\1&1&0&0\\0&0&0&1\end{bmatrix}\quad \mathbf{T}_3=\begin{bmatrix}0&1&0&0\\1&0&1&1\\0&0&0&0\\0&0&0&0\end{bmatrix}\quad \mathbf{T}_4=\begin{bmatrix}0&0&0&0\\0&0&0&0\\0&0&1&1\\1&1&0&0\end{bmatrix}$$

C. Slot Assignments

The slot assignments w_{ij}'s sent to the switched video adaptors can be determined as follow:

$$\begin{bmatrix}\text{Inputs:} & r_i \text{ for } i=1,2,\cdots,M\\ \text{Outputs:} & w_{ij} \text{ for } i=1,2,\cdots,M \text{ and } j=1,2,\cdots,L\end{bmatrix}$$

1. FOR m=1 TO h and i=1 TO M DO
2. the m^{th} bit of w_{ij} (where $j=r_i^{(m)}$) is set to one;

5.4 Network Design Example

A. Hierarchical Distribution of Videos

To provide video services to a large number of customers, it is possible to use multiple copies of the above designed distribution network and connects them in a hierarchical manner. Such an arrangement is shown in Fig.11. The hierarchical design consists of a central distribution hub and K local distribution subnets. The central distribution hub is located in a convenient place where there is access to the public broadband network and a video warehouse. It is responsible for distributing broadcast and switched videos to the K local distribution subnets

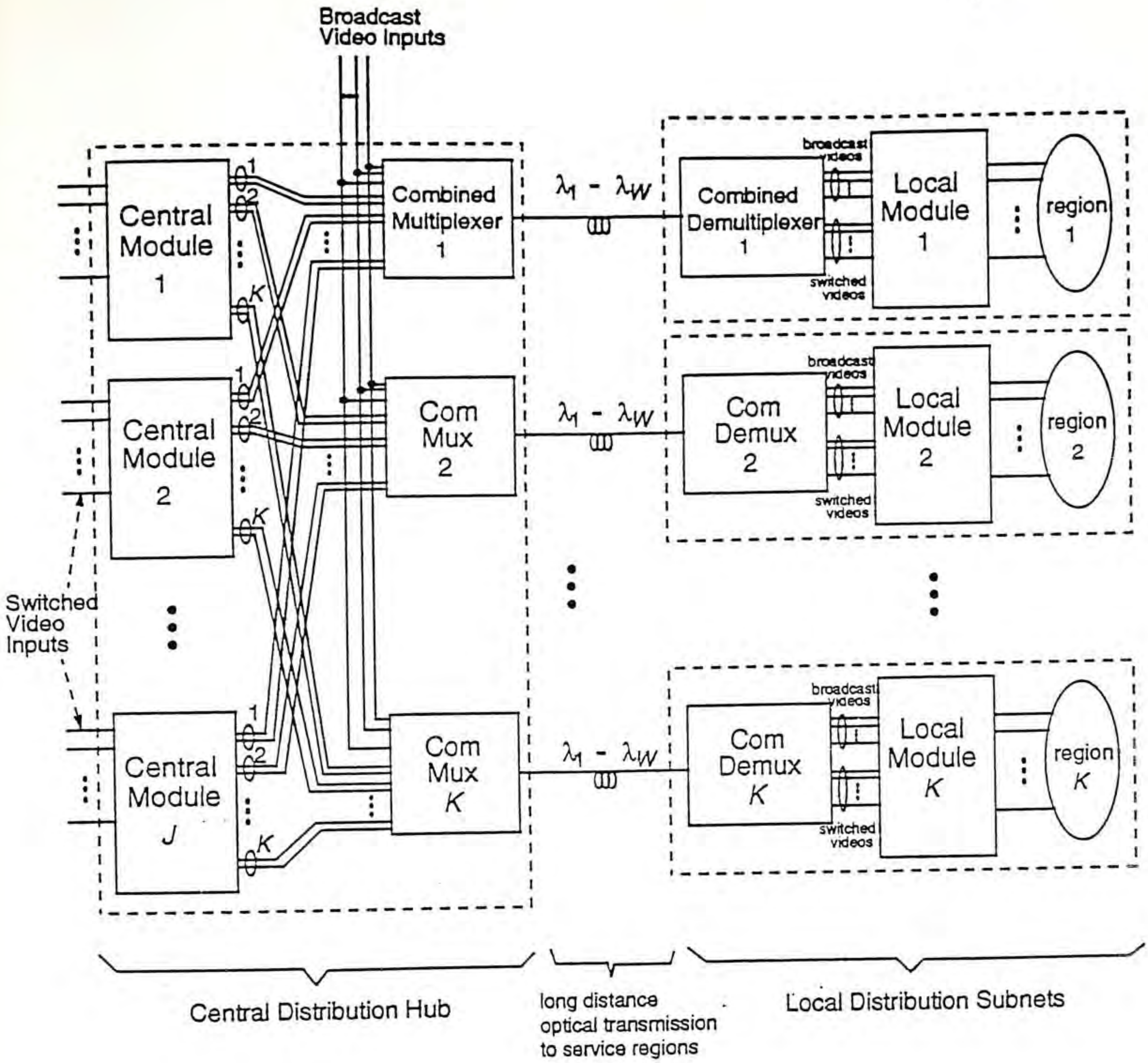


Fig.11: A hierarchical video distribution network

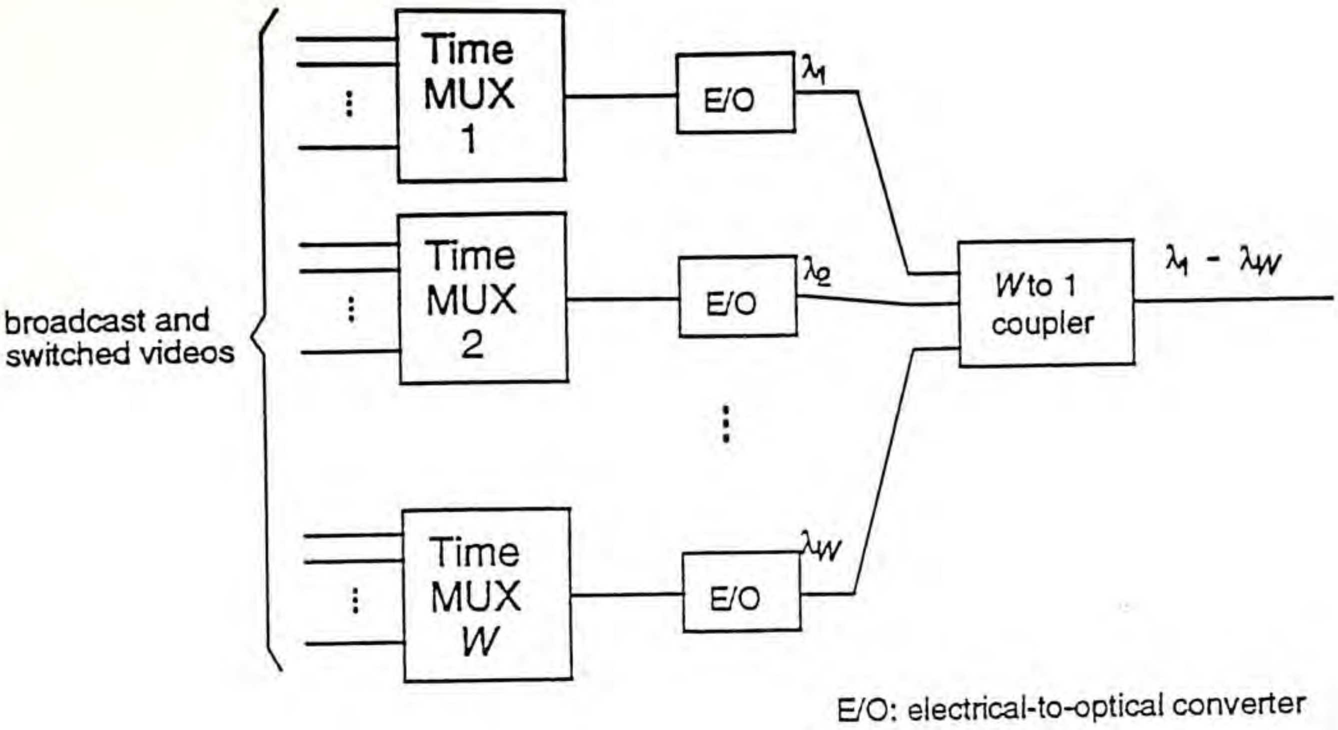
through optical carriers. Each local distribution subnet is located in a service region and distributes the received videos to the customers in its service region. This hierarchical design can replace a large switch (or bus matrix) by a network of smaller switches and reduce the overall circuit mileage.

A central distribution hub consists of J *central modules* and K *combined multiplexers*. A central module is a shared media switch as shown in Fig.1 but without a video multiplexer as only switched videos are connected. It therefore can devote all the slots in a frame to switched videos. The output lines from each central module are divided into K groups (one for each region) connected to K combined multiplexers. The broadcast video inputs are fed into all K combined multiplexers. Fig.12(a) shows a block diagram of a combined multiplexer. The broadcast and switched videos are further divided into W groups and each group is connected to a time multiplexer. The time multiplexed signals are then converted to optical signals at specific wavelengths. The W optical channels at wavelengths $\lambda_1, \lambda_2, \cdots, \lambda_W$ are coupled into an outgoing optical fiber for long distance transmission.

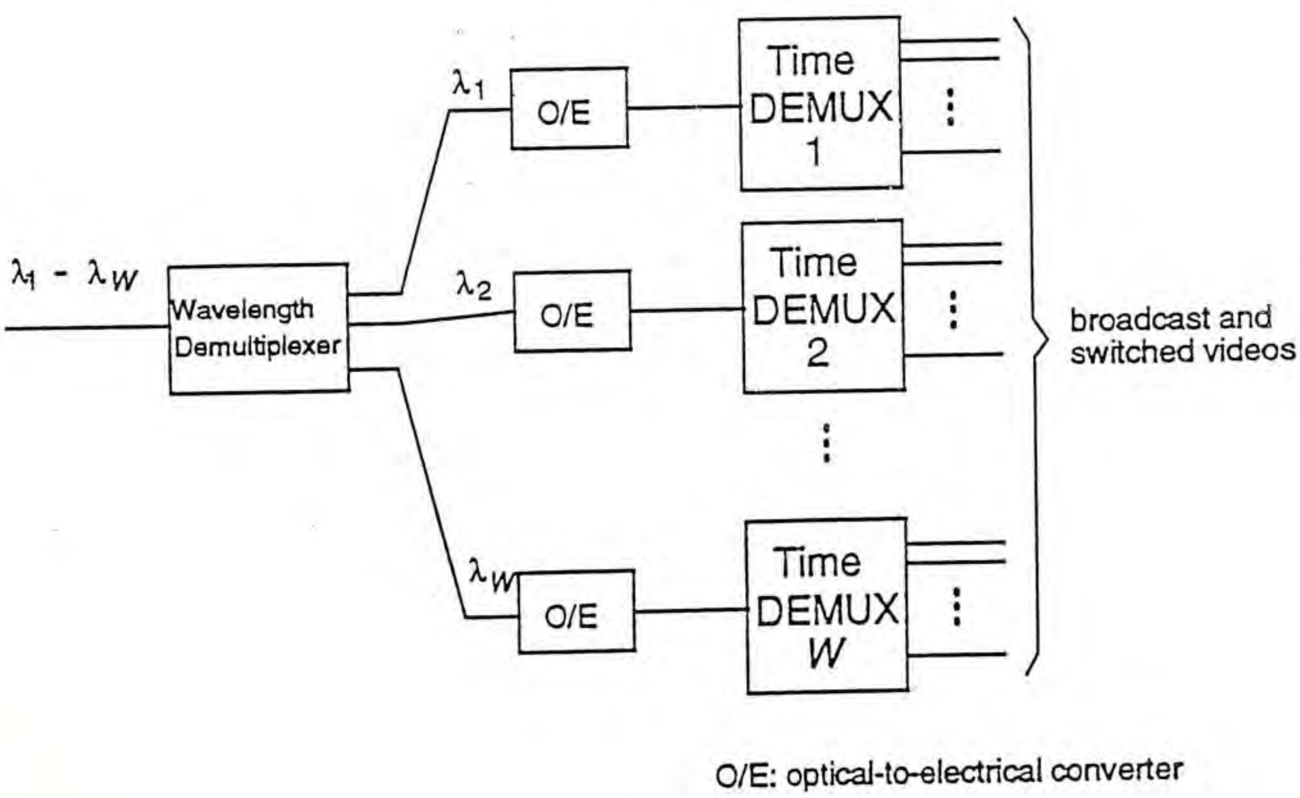
A local distribution subnet consists of a *combined demultiplexer* and a *local module*. A combined demultiplexer is shown in Fig.12(b). It demultiplexes the received optical signals into W optical channels. The optical signals of each channel are converted into electrical signals, which are then time-demultiplexed. The demultiplexed broadcast and switched videos are then connected to a local module, which is a shared media switch as shown in Fig.1, for further distribution to the customers in the local service region.

B. Blocking Probability

In the above design, a request for switched video may be blocked at a local distribution subnet (called local blocking) or at the central distribution hub (called central blocking). Based on the traffic model proposed by Yum [YUM 91], we derive the blocking probability for



(a) Combined Multiplexer



(b) Combined Demultiplexer

Fig.12: The combined multiplexer and demultiplexer

point-to-point transmission of switched videos. At the busiest hour of a day, let p be the probability that a customer requests a switched video session. A customer is equally likely to request a switched video from any one of the J central modules. Let the central module use a $R_C \times R_C$ bus matrix to switch a total of $R_C(g+h)$ videos to its output lines. Therefore, it can supply $Y = \lfloor R_C(g+h)/K \rfloor$ switched videos to each service region through a combined multiplexer and a local distribution subnet. The total number of customers N is divided into K service regions of $\lceil N/K \rceil$ customers each. Let the local module use a $R_L \times R_L$ bus matrix. Therefore, a local module has R_L output groups of $X = \lceil N/KR_L \rceil$ customers each.

Consider local blocking. Since each local module can supply at most h switched videos to each output group, blocking occurs when more than h customers in the same output group request switched videos. The local blocking probability B_L is given by

$$B_L = \sum_{n=h+1}^{X} \frac{n-h}{n} \binom{X}{n} p^n (1-p)^{X-n} \tag{1}$$

Consider central blocking. Since a central module can supply at most Y switched videos to each service region, blocking occurs when more than Y customers in the same service region request switched videos from a central module. Let $\tilde{Z}_i$ be the number of requests from group i customers in a specific service region to a specific central module. Then $P[\tilde{Z}_i = z]$ is given by

$$P[\tilde{Z}_i = z] = \begin{cases} \binom{X}{z} \left(\frac{p}{J}\right)^z \left(1-\frac{p}{J}\right)^{X-z} & z < h \\ \sum_{j=h}^{X} \binom{X}{j} \left(\frac{p}{J}\right)^j \left(1-\frac{p}{J}\right)^{X-j} & z = h \end{cases} \tag{2}$$

Let $\tilde{Z}_{SUM} = \tilde{Z}_1 + \tilde{Z}_2 + \cdots + \tilde{Z}_{R_L}$ be the total number of requests from a specific service region to a specific central module. Since all $\tilde{Z}_i$'s are i.i.d. random variables, the distribution of $\tilde{Z}_{SUM}$ is just the R_L-fold convolution of the distribution of $\tilde{Z}_1$ with itself. Central blocking occurs when $\tilde{Z}_{SUM} > Y$ and its probability B_C is given by

$$B_C = \sum_{n=Y+1}^{R_L h} \frac{n-Y}{n} P[\check{Z}_{SUM}=n] \tag{3}$$

The overall blocking probability B_o is B_L+B_C.

C. Optimal Network Dimensioning

The complexity of a distribution network can be grossly measured by the number of switching elements used. In the above hierarchical design, this is equal to $JR_C^2+KR_L^2$. The size of a network depends on the service requirement. Here, we stipulate that the overall blocking probability B_o must be less than a specified value B^*, say equal to 10^{-3}. The minimum complexity network configuration can be determined by minimizing $JR_C^2+KR_L^2$ with respect to J, K and X subject to $B_o \le B^*$. As an example of optimal design, we take g=10, h=30 and p=0.25 and show in Fig.13 the complexity of the optimal hierarchical networks as a function of N. Thus for N=5000 and B^*=10^{-3}, the complexity of the optimal hierarchical network is 508. The corresponding values of J, K, X, R_C and R_L are J^*=63, K^*=16, X^*=79, R_C^*=2 and R_L^*=4. In other words, the optimal hierarchical network consists of 63 2×2 bus matrix based central modules in the central distribution hub and 16 4×4 bus matrix based local modules for the 16 local distribution subnets. This reveals that small size bus matrices are preferred because the complexity of a bus matrix is equal to the square of its size. When the blocking requirement B^* is relaxed to 10^{-2}, the complexity of the optimal hierarchical network is decreased to 356, a 30% reduction from 508 for B^*=10^{-3}. The actual effect of B^* on the customers is subjective and is not the aim of this study. Also shown in Fig.13 is the complexity of the optimal single-stage network. It is seen that the complexity of the hierarchical network is only a small fraction of that of the single-stage network. For B^*=10^{-3}, this fraction decreases from 0.19 for N=3000 to 0.146 for N=5000.

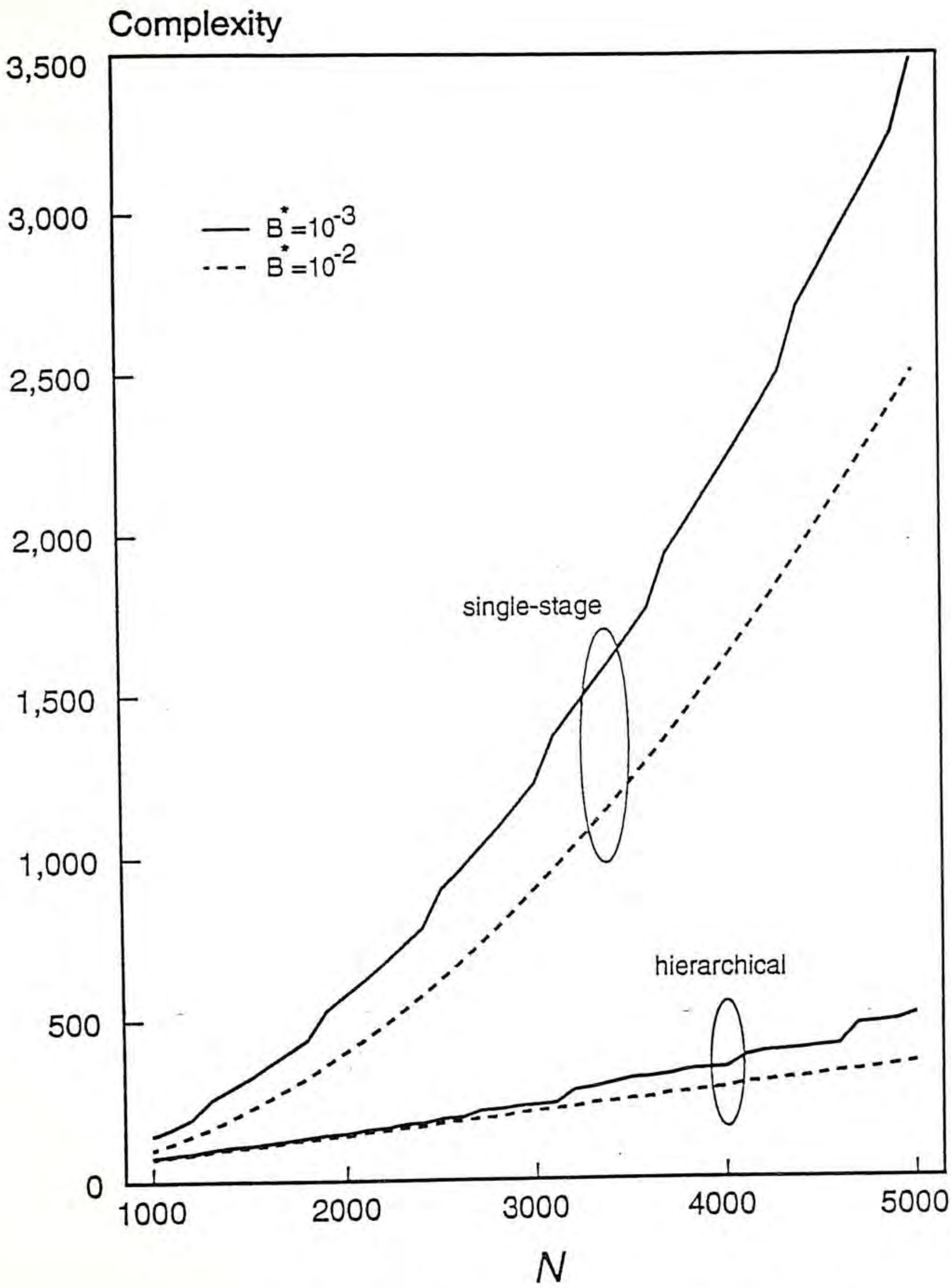


Fig.13: Complexity versus N

5.5 Chapter Summary

Videos will constitute the bulk of the traffic in broadband ISDN. It is therefore essential that videos are switched and transmitted to the customers in a cost effective manner. Shared medium based packet switches are potential candidates for future broadband communications. A shared media architecture was proposed in this chapter for distributing broadcast and switched videos using shared buses and a TDM-based bus matrix. Two slot assignment algorithms were designed, one for point-to-point transmissions and another for point-to-multipoint transmissions. A network design example was given to illustrate how a large video distribution network can be constructed and dimensioned using this architecture as building blocks.

Appendix

To show that SA/PP Algorithm can give the proper transmission matrices, it suffices to consider the worse case where all the row sums and column sums of the group traffic matrix C are equal to h. Let each execution of the inner loop of the SA/PP Algorithm be called an iteration. Just before a transmission matrix is computed, let ν be the number of entries in r that are equal to one.

Lemma 1

If a non-zero element b_{pq} in B is selected in an iteration, r_i and c_j ($i \neq p$ and $j \neq q$) are decremented by at most one in this iteration.

Proof

When b_{pq} is selected in an iteration, all the elements in row p and column q are set to zero. At the beginning of this iteration, if the element b_{pj} ($j \neq q$) on row p is zero, b_{pj} is not changed in this iteration and hence c_j is also unchanged; if $b_{pj} > 0$ ($j \neq q$), b_{pj} is set to zero in this iteration and hence c_j is decremented by one. The result that r_i, for all $i \neq p$, are decremented by at most one can be proved in a similar manner.

Lemma 2

If c_q=1 (r_p=1) and the non-zero element b_{pq} of B is selected in an iteration, r_i for all $i \neq p$ (c_j for all $j \neq q$) are not changed in this iteration.

Proof

Since c_q=1, only the element b_{pq} in column q of B is larger than zero and hence only b_{pq} in column q is set to zero in this iteration. Hence, only r_p in r becomes zero while r_i for all $i \neq p$ are not changed. The case where r_p=1 and c_j for all $j \neq q$ are not changed can be proved in a similar manner.

Lemma 3

After the ν^{th} iteration, at most one entry in $\boldsymbol{r}$ ($\boldsymbol{c}$) can be equal to one.

Proof

In the first ν iterations for computing a transmission matrix, only the non-zero elements in row p with r_p=1 and column q with c_q=1 of B are selected until $r_i \neq 1$ and $c_j \neq 1$ for all $1 \leq i,j \leq M$. After the ν^{th} iteration, the smallest non-zero entry in $\boldsymbol{r}$ and $\boldsymbol{c}$ remains larger than 1 until it is equal to two at the beginning of an iteration so that, by Lemma 1, the smallest non-zero entry may become one in this iteration. Let c_q=2 and the two non-zero elements in column q of B be b_{pq} and b_{uq}. Suppose b_{uq} is selected in this iteration. Only the elements b_{pq} and b_{uq} in column q of B are changed to zero. Hence, r_u becomes zero, r_p decrements by one while all other entries in $\boldsymbol{r}$ are not changed. If r_p>2 at the beginning of this iteration, then r_p>1 and hence all non-zero entries in $\boldsymbol{r}$ are larger than one at the end of this iteration; if r_p=2, then only r_p in $\boldsymbol{r}$ is equal to one at the end of this iteration. In the next iteration, if r_p=1 and the non-zero element b_{pv} in row p of B is selected, only r_p and c_v are set to zero (Lemma 2). Then the smallest non-zero entry in $\boldsymbol{r}$ and $\boldsymbol{c}$ is again equal to or larger than 2. By repeating the above arguments, it can be concluded that at most one entry in $\boldsymbol{r}$ is equal to one. The case that at most one entry in $\boldsymbol{c}$ is equal to one can be proved in a similar manner.

Theorem

A proper transmission matrix can be found in each execution of the outer loop of the SA/PP Algorithm.

Proof

A proper transmission matrix can be found iff only one entry in $\boldsymbol{r}$ and one entry in $\boldsymbol{c}$ become zero in any iteration such that one input group is connected to one output group in this iteration. In the first ν iterations, only the non-zero elements of row p with r_p=1 and column q with c_q=1 of B are selected until $r_i \neq 1$ and $c_j \neq 1$ for all $1 \leq i,j \leq M$. By Lemma 2, only one entry in $\boldsymbol{r}$

and one entry in c are set to zero in each of these iterations. Consider any one of the remaining iterations and let the non-zero element b_{pq} of B be selected such that r_p and c_q are set to zero in this iteration. It remains to show that no other non-zero entry in r and c will become zero in this iteration. If the smallest non-zero entry in r and c is larger than one, the non-zero entries excluding r_p and c_q will not become zero in this iteration because they only decrement by at most one in this iteration (Lemma 1). If the smallest non-zero entry in r and c is one, we distinguish two cases. First, when r_p=1 and b_{pq} is selected, c_j for all $j \neq q$ are not changed in this iteration (Lemma 2). Hence, the non-zero entries in c excluding c_q are not set to zero in this iteration. Since all non-zero entries excluding r_p in r are larger than one (Lemma 3) and these non-zero entries decrement by at most one in this iteration (Lemma 1), they will not become zero in this iteration. Hence, only r_p in r and c_q in c become zero. For the second case where c_q=1 and b_{pq} is selected, it can be proved that only r_p in r and c_q in c become zero in a similar manner.

Chapter 6
Conclusions and Topics for Future Investigation

In this thesis, we tackled the routing and switching issues of multipoint videoconferencing. Both types of issues have the common characteristics that a source sends packet to multiple destinations or multiple sources send packets to one destination.

In chapter 1, we reviewed the broadband video services proposed in CCITT Recommendation I.211, discussed their potential demands and applications. Among the many proposed video services, videoconference service is considered to be one of the most "futuristics" services because the businessmen using this service can save both travel time and expenses and the business community can afford and is willing to pay higher service charges.

In chapter 2, we tackled the multiple destinations routing (MDR) problem. The MDR problem has been shown to be NP-complete. We proposed a method to reduce the number of enumerations required for finding the optimal solution. With this method, it is feasible to determine the optimal solutions when the number of nodes to be connected is small (say ≤ 5). Three heuristics are designed for large MDR problem (i.e. the number of nodes to be connected is large). With the use of the property that a good path to connect the source node and a destination node should not be too much longer than the shortest path connecting them, we designed heuristic $A^{(k)}$. The parameter k allows us to trade-off between optimality and computation time. Heuristic B is modified from the Prim's algorithm for finding minimum spanning tree, taking advantage of the property that two or more paths may share common edges. Heuristic C is for networks with uniform edges weights and it overcomes the deficiency of heuristic B.

In chapter 3, we formulated the connection optimization problem for selectable media conferences and common media conferences, and designed algorithms for finding the optimal connection paths. The blocking probabilities of selectable media conferences and common media conferences in fully connected networks were derived and compared. It was found that, for a given mean conference size, the variance of the conference size distribution has a small but non-negligible effect on the network throughput.

In chapter 4, we designed a TDM-based multibus packet switch. This switch has the advantages of (1) very simple control circuit, (2) 100% potential throughput under uniform heavy traffic, (3) internally unbuffered and (4) adding input and output ports without increasing the bus and I/O adaptor speed. A combined analytical and simulation method is used to obtain the packet delay and packet loss probability. Numerical results show that for satisfactory performance the buses need to run about 30% faster than the input line rate.

We proposed in chapter 5 a switch architecture for distributing broadcast and switched videos using a set of concentration buses (or input buses), a TDM-based bus matrix and a set of distribution buses (or output buses). This switch architecture has three advantages. First, multirate video channels can be accommodated. This can accommodate a variety of video services that have different bit rate requirements. Second, videos can easily be broadcast or multicast to the customers through the shared media. Hence, multipoint communication services (e.g. videoconferencing) can be provided. Third, it can be used as a building block for constructing large video distribution networks.

We now highlight some possible extensions of this research:

A. Routing for Multipoint Videoconferencing

(1) In chapter 2, we assumed that every node in the network can copy an incoming packet to multiple output links (i.e. has multicast capability). If this assumption is relaxed, the multiple destination routing problem becomes quite interesting.

(2) In chapters 2 and 3, we designed algorithms to find the connection paths with minimum cost. Dynamic routing algorithms can be designed to take other factors (e.g. current traffic load in each link) into account.

B. Switching for Multipoint Videoconferencing

(1) For the TDM-based multibus packet switch proposed in chapter 4, the bus bandwidth is allocated to the input adaptors in a fixed cyclic order. Under heavy traffic, the switch has 100% potential throughput. However, if the traffic is not heavy and is highly asymmetric, then dynamic bandwidth allocation policies can be designed to improve the delay performance. In section 4.6, we proposed a dynamic bandwidth allocation policy. Detailed analysis of this policy under non-uniform traffic need to be worked on.

(2) If the number of video ports is not large, the video switch proposed in chapter 5 should be realizable in a university laboratory because the speed requirement for the shared buses is not large. The circuits can first be designed and simulated using CAD packages. Then the circuits can be implemented using off the shelf components.

C. Traffic Engineering for Multipoint Videoconferencing

(1) In chapter 3, we derived the blocking probabilities for selectable media and common media conferences in fully connected networks. Although the analysis is applicable to general network topology, the numerical computation is intractable. Therefore, approximations and traffic engineering tools need to be devised.

(2) In chapter 5, we derived the blocking probability for point-to-point transmission of switched videos in the hierarchical video distribution network. A traffic model for point-to-multipoint transmission of switched videos in the hierarchical video distribution network is needed.

References

[ACAM 90] A.S.Acampora, "A high capacity metropolitan area network using lightwave transmission and time multiplexed switching", *IEEE Trans. Commun.*, vol.38, no.10, pp.1761-1770, October 1990.

[AHAM 90] M.Ahamad (Ed.), *Multicast Communication in Distributed Systems*, IEEE Press, 1990.

[AHMA 89] H.Ahmadi, W.E.Denzel, "A survey of modern high-performance switching techniques", *IEEE Journal on Selected Areas in Communicaions*, vol.7, no.7, p.1091-1103, September 1989.

[AHO 83] A.V.Aho, J.E.Hopcroft, J.D.Ullman, *Data Structures and Algorithms*, Addison-Wesley, 1983.

[ALBE 85] A.F.Alber, *Videotex/Teletext: Principles and Practices*, McGraw Hill, 1985.

[ARMB 84] H.Armbruster, "Systems for information retrieval over wideband networks", *Proceedings of IEEE International Conference on Communications*, pp.1414-1419, May 1984.

[ARMB 86] H.Armbruster, "Applications of future broadband services in the office and hone", *IEEE J. Select. Areas Commun.*, vol.4, no.4, pp.429-437, July 1986.

[ARMB 87] H.Armbruster and G.Arndt, "Broadband communication and its realization with broadband ISDN", *IEEE Commun. Mag.*, vol.25, no.11, pp.8-19, Nov.1987.

[BERT 87] D.Bertsekas, R.Gallager, *Data Networks*, Prentice Hall, 1987.

[BHAR 83] K. Bharath-Kumar, J.M.Jaffe, "Routing to Multiple Destinations in Computer Networks", *IEEE Trans. on Commun.*, vol.31, no.3, March 1983.

[BODS 89] D.Bodson and R.Schaphorst (Ed.), *Teleconferencing*, IEEE Press, 1989.

[BONU 89] M.A.Bonuccelli, "A fast time slot assignment algorithm for TDM hierarchical switching systems", *IEEE Trans. Commun.*, vol.37, no.8, pp.870-874, August 1989.

[BRAN 87] S.Brand, *The Media Lab: Inventing the Future at MIT*, Viking Penguin, 1987.

[BROW 80] E.F.Brown, V.J.Geller, J.Goodnow, D.C.Hoecker, M.Wish, "Some objective and subjective differences between communication over two video conferencing systems", *IEEE Trans. Commun.*, vol.28, no.5, pp.759-764, May 1980.

[CANA 84] F.Canavesio, R.Ceruti and S.Gallo, "Voice switching system for video conference studios", *Proceddings Int. Teleconference System*, Toronto, Canada, pp.52-56, April 1984.

[CCIT 90] CCITT Draft Recommendation I.211, "B-ISDN Service Aspects", Study Group XVIII, Geneva, May 23-25, 1990.

[CHOR 84] D.N.Chorafas, *Interactive message services*, McGraw-Hill, 1984.

[CIDO 88a] I.Cidon, I.Gopal, "PARIS: An approach to integrated high speed private networks", *Int. J. Digit. Analog Cabled Syst.*, vol.1, p.77-85, 1988.

[CIDO 88b] I.Cidon, I.Gopal, G.Grover, M.Sidi, "Real time packet switching: A performance analysis", *IEEE Journal on Selected Areas in Communications*, vol.6, no.9, p.1576-1586, December 1988.

[FERG 84] M.J.Ferguson and L.G.Mason, "Network design for a large class of teleconferencing systems", *IEEE Trans. Commun.*, vol.32, no.7, pp.789-796, July 1984.

[FISC 90] D.Fischer and R.Gupta, "Taking a second look at videoconferencing", *Data Commun. International*, pp.85-92, June 1990.

[GARE 79] M.R.Garey, D.S.Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, San Franciso, CA:Freeman, 1979.

[GILB 68] E.N.Gilbert, H.O.Pollak, "Steiner minimal tree", *SIAM J.Appl.Math*, vol.16, 1968.

[GOPA 84] I.S.Gopal, J.M.Jaffe, "Point-to-Multipoint communication over broadcast links", *IEEE Trans. Commun.*, vol.32, no.9, pp.1034-1044, September 1984.

[GOTZ 87] Gotz Romahn, "System Aspects of Multipoint Videoconferencing", *Proc. of IEEE Globecom* 1987.

[HAYE 84] J.F.Hayes, *Modelling and analysis of computer communication networks*, Plenum Press, 1984.

[HONG 91] D.Hong, T.Suda, "Congestion control and prevention in ATM networks", *IEEE Networks*, pp.10-16, July 1991.

[HOWA 71] R.Howard, *Dynamic Probabilistic Systems*, vol.1: Markov Models, p.286, 1971.

[HUI 89] Joseph Y.Hui, *Switching and Traffic Theory for Integrated Broadband Networks*, 1989.

[IEEE 87] IEEE J. Selected Areas Commun., Special Issue on Switching Systems for Broadband Networks, vol.5, Oct. 1987.

[IEEE 90] IEEE J. Selected Areas Commun., Special Issue on Dense Wavelength Division Multiplexing Techniques for High Capacity and Multiple Access Communication Systems, vol.8, no.6, August 1990.

[INUK 79] T.Inukai, "An efficient SS/TDMA time slot assignment algorithm", *IEEE Trans. Commun.*, vol.COM-27, no.10, pp.1449-1455, Oct. 1979.

[JUDI 96] C.N.Judice, E.J.Addeo, M.I.Eiger, H.L.Lemberg, "Video on demand: a wide-band service or myth?", *Proceedings of IEEE International Conference on Communications*, p.1735-1739, June 1986.

[KAMO 80] F.Kamoun, L.Kleinrock, "Analysis of shared finite storage in a computer network node environment under general traffic conditions", *IEEE Trans. Commun.*, vol.COM-28, no.7, p.992-1003, July 1980.

[KELL 82] C.W.Kelly, "A low bandwidth virtual space video teleconferencing system", *Proceedings of IEEE GLOBECOM*, pp.966-970, Dec. 1982.

[KELL 86] F.P.Kelly, "Blocking probabilities in large circuit-switched networks", *Adv. Appl. Prob.*, vol.18, pp.473-505, 1986.

[KENY 85] N.D.Kenyon, T.A.White and G.M.Reid, "Behavioral and user needs for tele-conferencing", *Proceedings of IEEE*, vol.73, no.4, pp.689-699, April 1985.

[KLEI 74] L.Kleinrock, *Queueing Systems, volumn 1: Theory*, John Wiley & Sons, 1974.

[KLEI 76] L.Kleinrock, *Queueing Systems, volumn 2: Computer Applications*, John Wiley & Sons, 1976.

[KLEI 85] P.Klein, "Video workplace conference", *Proceedings of IEEE GLOBECOM*, pp.109-112, 1985.

[KOBA 78] H.Kobayashi, *Modelling and Analysis: An Introduction to System Performance Evaluation* Methodology, pp.108-110, Addison-Wesley, 1978.

[KOU 81] L.Kou, G.Markowsky, L.Berman, "A faster algorithm on Steiner trees", *Acta Inform.*, vol.15, 1981.

[LEE 90] K.J.Lee, A.Gersht, A.Friedman, "Multipoint connection routing", *International Journal of Digital and Analog Communication Systems*, vol.3, p.177-186, 1990.

[LIAO 87] K.Q.Liao and J.W.Roberts, "Videoconference traffic and network design", *IEEE Trans. Commun.*, vol.35, no.3, pp.275-282, March 1987.

[MAGL 88] B.Maglaris, D.Anastassiou, P.Sen, G.Karlsson, J.D.Robbins, "Performance models of statistical multiplexing in packet video communications", *IEEE Trans. Commun.*, vol.36, no.7, pp.834-844, July 1988.

[MEHL 88] H.Mehlhorn, "A faster approximation for the Steiner problem in graphs", *Inform. Process. Lett.*, vol.27, no.3, pp.125-128, March 1988.

[MOKH 80] N.Mokhoff, "The global video conference", *IEEE Spectrum*, vol.17, pp.45-47, 1980.

[MUKH 91] B.Mukherjee, "Integrated voice-data communication over high-speed fiber optic networks", *IEEE Computer*, vol.24, no.2, pp.49-58, February 1991.

[NOJI 87] S.Nojima, E.Tsutsui, H.Fukuda, M.Hashimoto, "Integrated services packet network using bus matrix switch", *IEEE Journal on Selected Areas in Communications*, vol.5, no.8, p.1284-1292, October 1987.

[PAYN 88] J.A.Payne, *Introduction to Simulation*, McGraw Hill, 1988.

[RAYW 83] V.J.Rayward-Smith, "The computation of nearly minimal Steiner trees in graphs", *Int. Math. Ed. Sci. Tech.*, vol.14, no.1, pp.15-23, 1983.

[RAYW 86] V.J.Rayward-Smith, A.Clare, "On finding Steiner trees", *Networks*, vol.16, pp.283-294, 1986.

[ROBI 91] J.Robinson, E.Rubinov, C.Toulson, B.Prasada, S.Sabri, N.Goldberg, G.Vonderweidt, "A multimedia interactive conferencing application for personal workstations", *IEEE Trans. Commun.*, vol.39, no.11, pp.1698-1707.

[ROMA 87] G.Romahn, "System aspects of multipoint videoconferencing", *Proceedings of IEEE GLOBECOM*, pp.723-725, 1987.

[ROSS 83] S.M.Ross, *Stochastic Processes*, John Wiley and Sons, 1983.

[ROSS 89] S.M.Ross, *A First Course in Probability*, Macmillan, 3rd edition, 1989.

[ROSS 91] S.M.Ross, *A Course in Simulation*, Macmillan, 1991.

[SARI 85] S.Sarin, "Computer-based real-time conferencing systems", *IEEE Computers*, pp.33-45, Oct. 1985.

[SABR 85] S.Sabri and B.Prasada, "Video conferencing systems", *Proceedings of IEEE*, vol.73, no.4, pp.671-688, April 1985.

[SCHW 77] M.Schwartz, *Computer-Communication Network Design and Analysis*, Prentice-Hall, 1977.

[SCHW 87] M.Schwartz, *Telecommunication Networks: Protocols, Modeling and Analysis*, p.519, Addison Wesley, 1987.

[SEET 86] J.Seetzen, "Broadband communications-Prospects of utilization and economic implications", *IEEE J. Select. Areas Commun.*, vol.4, no.4, pp.640-645, July 1986.

[SEN 89] P.Sen, B.Maglaris, N.E.Rikli, D.Anastassiou, "Models for packet switching of variable-bit-rate video sources", *IEEE J. Select. Areas Commun.*, vol.7, no.5, pp.865-869, June 1989.

[SEYL 73] A.J.Seyler, H.Bruggemann, R.W.Kelt and J.Dickson, "The APO TV conferencing facility", *Telecommun. J. Australia*, vol.23, no.3, pp.216, 1973.

[STAL 88] W.Stallings, *Integrated Services Digital Networks*, 2nd edition, IEEE Press, 1988.

[STAL 92] W.Stallings, *ISDN and Broadband ISDN*, 2nd edition, Macmillan Pub., 1992.

[STRA 84] C.E.Strangio, *Digital Electronics*, Prentice Hall, 1984.

[STUC 85] .W.Stuck, E.Arthurs, *A Computer and Communication Network Performance Analysis Primer*, Prentice-Hall, 1985.

[SUZU 89] H.Suzuki, H.Nagano, T.Suzuki, T.Takeuchi, S.Iwasaki, "Output buffer switch architecture for asynchronous transfer mode", *Proceedings of IEEE International Conference on Communications*, p.99-103, 1989.

[SYSL 83] M.M.Syslo, N.Deo, J.S.Kowalik, *Discrete Optimization Algorithms with Pascal Programs*, p.253-268, 1983 Prentice Hall.

[TANE 88] A.S.Tanenbaum, *Computer Networks*, 2nd edition, Prentice Hall, 1988.

[TOBA 90] F.A.Tobagi, "Fast packet switch architectures for broadband integrated services digital networks", *Proceedings IEEE*, vol.78, no.1, p.133-167, January 1990.

[TOBA 91] F.Tobagi, IEEE INFOCOM Tutorial Notes on High-Speed Networking, 1991.

[VOND 91] G.Vonderweidt, J.Robinson, C.Toulson, J.Mastronardi, E.Rubinov, B.Prasadra, "A multipoint communication service for interactive applications", *IEEE Trans. Commun.*, vol.39, no.12, pp.1875-1885, December 1991.

[WAGN 89] S.S.Wagner, R.C.Menendez, "Evolutionary architectures and techniques for video distribution on fiber", *IEEE Commun. Mag.*, vol.27, no.12, pp.17-25, December 1989.

[WALL 80] D.W.Wall, "Mechanisms for broadcast and selective broadcast", Ph.D. dissertation, Department of Electrical Engineering and Computer Science, Standford University, 1980.

[WATA 85] K.Watanabe, S.I.Murakami, H.Ishikawa and T.Kamae, "Audio and visually augumented teleconferencing", *Proceedings of IEEE*, vol.73, no.4, pp.656-669, April 1985.

[WAXM 88] B.M.Waxman, "Routing of multipoint connections", *IEEE J. on Selected Areas in Commun.*, vol.6, no.9, December 1988.

[WEIN 78] C.J.Weinstein, "Fractional speech loss and talker activity model for TASI and for packet-switched speech", *IEEE Trans.Commun.*, vol.26, no.8, August 1978.

[WEIS 90] C.Weiss, "Desk top videoconferencing: an important feature of future visual communications", *Proceedings of IEEE International Conference on Communications*, pp.134-139, 1990.

[WHIT 85] W.Whitt, "Blocking when service is required from several facilities simultaneously", *AT&T Technical Journal*, vol.64, no.8, pp.1807-1856, October 1985.

[WOLF 89] R.W.Wolf, *Stochastic Modelling and the Theory of Queues*, Prentice-Hall, 1989.

[WRIG 83] B.A.Wright, "The design of picturephone meeting service", *IEEE Commun. Mag.*, vol.21, pp.30-36, March 1983.

[YUM 80] T.S.Yum, "Measuring the utilization of a synchronous data link: an application of busy period analysis", *The Bell System Technical Journal*, vol.59, no.5, p.731-744, May 1980.

[YUM 91] T.S.Yum, "Hierarchical distribution of video with dynamic port allocation", *IEEE Trans. Commun.*, vol.39, no.8, pp.1268-1274, August 1991.

[ZIEG 90] C.Ziegler and G.Weiss, "Multimedia conferencing on local area networks", *IEEE Computers*, pp.52-61, Sept. 1990.