DISTRIBUTED CALL SET-UP ALGORITHMS IN BISDN ENVIRONMENT

By

Shum Kam Hong

A Thesis

submitted in partial fulfillment of the requirements

for the Degree of Master of Philosophy

Division of Information Engineering

The Chinese University of Hong Hong

December 1992

Acknowledgement

I would like to express my gratitute to my supervisor Professor S.-Y. Robert Li, for his advice, guidance and encouragement throughout the course of the project. I am benefited from sharing his thirst for elegance and perfection. I also want to give my thanks to Mr. C.S. Leung and W.H. Mow for their help in the preparation of this thesis.

Since a lot of computing resources in the departments of Engineering have been consumed at the time of extensive simulation, I am indebted to my fellow students for bringing them inconvenience. Heartfelt thanks are also due to my family and Denise for their continuing support and patience.

Abstract

The main problem for resource management in broadband ISDN is the difficulties of administrating call admission that can avoid buffer overflow due to traffic burst. Detailed design of distributed algorithms for traffic control and call set-up are proposed and described. The algorithms integrate both connectionless and connection oriented traffic control. The traffic control functions are performed at the packet level. The call set-up and flow control algorithms are for fast execution by simple hardware. Their characteristics include the property of being deadlock-free and starvation-free, preservation of the order among sequential packets, low buffer memory requirement, and high bandwidth utilization.

For evaluating the performance of the algorithms, various traffic mix models have been studied and a simulator for the broadband network of a general topology has been implemented. Simulation results have shown that (i) the algorithms can achieve high bandwidth utilization and low packet loss probability for different patterns of aggregate traffic due to superposition of periodic sources; (ii) If the traffic is mixed with on-off sources which will increase the traffic burstiness, the requirement of packet loss probability can be satisfied by adjusting parameters of the algorithms and trading off certain levels of bandwidth utilization.

Contents

Chapter 1

Introduction

1.1 Background

ISDN (Integrated Services Digital Network) is a means to provide a common integrated infrastructure that can be used by a wide variety of communication traffic. The reason in proposing ISDN is that the economies of scale will allow the services to be affordable to the largest possible set of users. In addition, it is more economical to have single ISDN rather than maintaining several networks, each might be for specific applications. ISDN also allows data sharing among different networks.

Broadband communication refers to bandwidth needed for video signals, that is the signals with bandwidth 100 Mbps range or above. The motivation of evolution is mainly due to the need for future broadband services. It can be predicted that besides the basic POTS (Plain Old Telephony Services) and ISDN services, future business and residential subscribers also required simultaneous delivery of the broadband services like video telephony, TV-quality services, video conference, video/document retrieval services etc. Furthermore, declining price of optical fiber, increasing speed and declining cost of micro-electronics devices make the BISDN technologies become possible.

Excluding the administration factors, such as problems of service pricing and standardization etc, there are two key issues that determine the success of BISDN. First is the implementation of extremely high speed switching. A brief survey of the current art in ATM (Asynchronous Transfer Mode) switch design is described in Section 1.3. The second is the design of fast and efficient management of network resources so as to take the full advantage of the flexibility of ATM. This thesis mainly addresses on this second issue.

Resource management in BISDN includes mechanisms for traffic flow control and call set-up algorithms. In a network of a general topology, a route of transmission may go through multiple inter-office links, and different routes may share a common link. When the combined traffic density on intersecting routes bursts to exceed the capacity of the common link, packets would pile up at the buffer of the link. To ensure that the probability of buffer overflowing in broadband communications is acceptably low, the requirement of buffer memory would have to be exceedingly large unless some constraint is exercised over the traffic influx into the network and over the traffic flow inside the network. For connection oriented communications, such as voice, video and certain realtime data communications, the call flows cannot be interrupted once the call is set up. The call set-up algorithm accepts incoming calls subject to the availability of sufficient free bandwidth to support the new connection. This, as well as the flow control, has a direct impact on the traffic condition in the network.

The classical approach of traffic flow control in many telephone networks, as well as other traffic systems, relies on the use of a control center, which collects and manages realtime traffic data of the whole network. The implementation of the control center incurs substantial overhead in requirements of network facilities, design complexity, and transmission capacity. For packet-level control in a broadband network, the communication and computing power of the control center may present a bottleneck. As stated in [JAI 91], the performance bottleneck of network has now been shifted from the link transmission speed to the processing speed at the network switching nodes and the propagation delay of the link. This consideration leads to the search for efficient distributed algorithms of flow control. A great deal of literature [GIE 81], [LAZ 83], [MAT 81], [PEN 75], [SPR 81], [TYM 81] has been devoted to the subject of flow control in communication networks.

For circuit-switched integrated service networks, [KRA 85], [RO1 89] and [RO2 89] have devised analytic solutions for access control methods to maximize the throughput. However, the scope of the solutions is restricted to single-hop service call with at most two classes of services.

For the ATM network, the traffic characteristics of statistical multiplexing make the multi-hop access control method even more difficult to solve than circuit switching. The simplest solution is to allocate transmission bandwidth at the peak bit rate of the call regardless of the burstiness. When the burstiness is high, this would under-utilize the transmission resource and the performance would be no better than STM (Synchronous Transfer Mode). On the other hand, if the allocated bandwidth is less than the peak bit rate, data may be lost due to buffer overflowing. A usual criterion for call set-up is by the traffic conditions of the transmission links along the route of the intended call. Traffic conditions are described by traffic parameters or traffic descriptors. Lists of proposed traffic parameters for call admission control can be found in [SAI 91] and [JAI 91].

There are generally two approaches of call admission control. The first approach is to make the admission decision based on the traffic parameters specified by a user prior to admission, e.g. the *class related rules* proposed by [GAL 89] and [DEC 90] for the admission control of both homogeneous and inhomogeneous traffic sources. The class related rules accept a call of a homogeneous traffic source only if the sum of the peak rate required to carry the existing connections and the intended call does not exceed the total link capacity. For inhomogeneous traffic sources, the calls are classified into classes according to their traffic parameter values. The intended call is accepted if the resulting mean traffic could be accepted even if it was offered by the class with the largest burstiness. Other examples of this approach are based on the probabilistic distribution of the number of bursts [SCH 88] or the packet loss probability [SUZ 90] and [NAK 89].

The other approach is to make the admission decision from on-line traffic parameter measurement [JOO 89] or real-time scheduling algorithm [HYM 91]. Neural network [HI1 89] and fuzzy logic [KES 91] are also used as on-line traffic monitors to control the call admission decision.

The distributed packet-level flow control algorithm in a broadband network studied in this thesis is concomitant with the call set-up algorithm. These algorithms are based on the preliminary design in [LI 90] and [LI 92]. The contents of this project include some of the detailed designs in Chapters 2 and 3, the traffic modelling in Chapter 4 and the works in Chapters 5 and 6.

The flow control functions make the decisions on whether to momentarily detain a packet at an intermediate node of its route as well as whether to admit a packet into the network. These functions are administered at each individual node and performed at the packet level. The basic criterion in detaining a passing-through packet is when traffic congestion ahead of the packet is anticipated rather than when buffering is forced by congestion at hand. The means of anticipation is through the extraction of back-pressure indicators from background traffic information exchanges between adjacent nodes. When the flow control reacts to back pressure, the burden of packet buffering against traffic burst is spread over all nodes.

The distributed call set-up algorithm is built on top of the previously described flow control algorithm. Under this algorithm, the originating node of an attempted call generates a periodic stream of *scout packets* to test traffic conditions in the network and thereby forms the judgment on whether the new call would cause congestion in the network. The algorithm is solely executed at the call originating node. For a one-way call, the call set-up process requires no inter-node communications.

The distributed call set-up and flow control algorithms are for fast execution by simple hardware. Their characteristics include the property of being deadlock-free and starvation-free, preservation of the order among sequential packets, low buffer memory requirement, and high bandwidth utilization.

1.2 Outline of the thesis

Consider a communication network of a general topology. Every node on the network represents an exchange center. For the example of a metropolitan-area telephone network, a node is either a local/toll office or simply a connection point. A link is a one-directional channel from one node to another. Channels in opposite directions are treated as two separate links. Outgoing links from a node will be referred to as links *of* that node, and all other links in the network will be *remote* links with respect to that node. A route of transmission may go through multiple links.

A link is logically divided into a data channel and a small control channel. The algorithms of flow control and call set-up are built on top of a scheme of traffic information exchanges over control channels. This scheme is described in Chapter 2 and can be outlined as follows. Every node in the network constantly assesses the congestion status of its own (outgoing) links and broadcasts this information to neighboring nodes through control channels. In fact, a node broadcasts not only the current status of its own links but also its *best* assessment of the status of other links of the network. Through such constant local information exchanges, every node acquires the congestion status of all links. There may possibly be contradictions among information from different sources due to different time lags. With the aid of static data characterizing the topography of the network, each node extracts from the ensemble of received information the most updated status of each remote link. The extent of freshness of this updated knowledge depends upon the distance between the node and the remote link. In any case it is the most updated knowledge that the node can possibly acquire through the transmission medium at the speed of light.

The flow control algorithm, described in Chapter 3, performs packet-level decisions at each individual node. The algorithm decides on the detention of passing-through packets and on the admission of new packets into the network. The basic criterion in detaining a passing-through packet is when traffic congestion ahead of the packet is anticipated rather than when buffering is forced by congestion at hand. The means of anticipation is through the extraction of back-pressure indicators from routine background information exchanges between adjacent nodes. The back-pressure scheme is designed to avoid the problem of excessive packet accumulation at any single node. Thus control decisions at a node react on the most updated link status that can possibly be acquired by that node. This helps striving for high utilization of link bandwidth. In regulating the traffic influx to the network, some capability of customer interface is assumed: When a new packet is denied entry to the network, it is either buffered or to be regenerated and, meanwhile, the packet source is notified of such an nodal action.

Each decision process in the packet-level control consists of only a few primitive operations. First, the address of the packet is translated into a binary word via a static routing table stored in a high-speed memory. Then, this word is logically ANDed with certain dynamic data word representing the traffic information. The control decision is purely based upon whether the AND result is zero. A sub-microsecond execution time is expected for these primitive operations. Fast execution of packet-level control is crucial for broadband communications. For example, if packets are 53-byte ATM cells transmitted at the STS-12 rate

(662 Mb/s), a packet-level control has to be performed in 0.64 microseconds.

For connection oriented communications, it is up to the call set-up process to control the traffic volume of periodic packet streams generated by calls. A call set-up algorithm is designed in conjunction with the aforementioned flow control algorithm. Under the call set-up algorithm, the originating node of an attempted call generates a periodic stream of *scout packets* to test traffic conditions in the network and thereby forms the judgment on whether the set-up of the new call would cause or worsen congestion in the network. The stream of scout packets has the same route as the attempted call. In the case of a successful call set-up, the flow of scout packets reserves a virtual circuit till the call starts. On the other hand, if the flow of scout packets is interrupted by the anticipation of traffic congestion, the call set-up process is aborted. For a one-way call, the call set-up process involves no inter-node communication and is solely by the call originating node. Thus an emulated circuit is created without bandwidth allocation. The call tear-down also involves no bandwidth accounting; a call ends when the contents of its final packet so indicates.

This call set-up algorithm is depicted in Chapter 4. Parameters in the algorithm are fine tuned with respect to some traffic models of integrated services. The first traffic model in Chapter 4 is the integration of periodic sources of different bit rates. The second is the integration of on-off sources of different bit rates. The third is the mix of these two. Chapter 5 is devoted to parameter fine tuning and analysis.

A Simulator for broadband network, described in Chapter 6, is designed to evaluate the performance of the algorithms. It is found that low loss rate and high bandwidth utilization can both be achieved by the algorithms under different network configurations, traffic loading and burstiness.

The organization of the remainder of this thesis may be summarized as follows. Section 1.3 gives a brief survey of the current art in packet switching. Chapter 2 describes the mechanism of traffic information exchanges over the control channels. The traffic flow control algorithm is introduced in Chapter 3. Chapter 4 proposes the basic call set-up algorithm and the various traffic models. Parameter fine tuning and analysis of the call set-up algorithm, described in Chapter 5, is based on the traffic models discussed in Chapter 4. Finally, Chapter 6 describes the simulator for broadband network and the performance evaluation of the flow control and call set-up algorithms.

1.3 Current Art in Packet Switching

It was confirmed at the Seoul meeting of CCITT SGXVIII in February 1988 [INT 88] that the target network solution for broadband communications would be based on ATM (Asynchronous Transfer Mode, formerly called ATD). In ATM, packets are in the form of fixed-size slots called *Cells* (Figure 1.1), which are identified and switched by means of a label in the header. ATM cell consists of a 5-Octet header and 48-Octet information payload. The header includes a label called Virtual Channel Identifier (VCI) and an error detecting field but the remaining contents of the header have not been fully defined.

The major advantage of applying ATM is the flexibility in handling unknown traffic characteristics in broadband network because bandwidth are dynamically shared among the active calls. The size of the cell is small, that implies smaller buffer and switch size. However, someone may argue that the efficiency of transmitting computer data traffic may be upgraded if the size of the cell is greater. In contrast to the situation in ATM, STM (Synchronous Transfer Mode) multiplexes packets synchronously from fixed number of channels (Figure 1.2). Although STM is simpler and more cost effective than ATM, it requires hierarchy to multiplex and may lead to inflexibility and inefficiency.

ATM switch design is one of the major issue in broadband communications. In this section, a brief survey on the current design of electronic broadband switches is presented. The transmission time of data in the traditional medium was the significant delay due to the limitation of bandwidth but the delay becomes negligible in optical fiber link. Therefore, switching delay becomes the bottleneck for performance because switches have to handle 100 Thousands - 1 Million packets per second per line. Device that can operate in such a high performance is difficult to design and implement. Some basic and desirable features of the fast packet switch are listed as below.

Table 1.1 : Basic Features

Features	Requirements
Delay	< 1 ms/node
Cell loss probability	$< 10^9$ per node for CBR services
	$< 10^7$ per node for VBR services
Line speed	> 150 Mb/s
Capacity range (ports)	several tens to several thousands or more

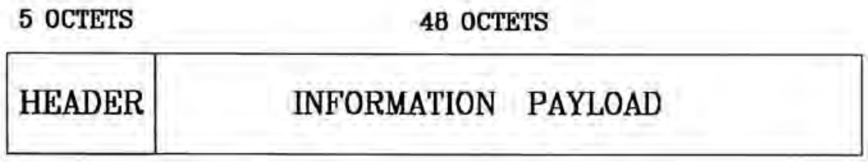


Figure 1.1 : ATM CELL

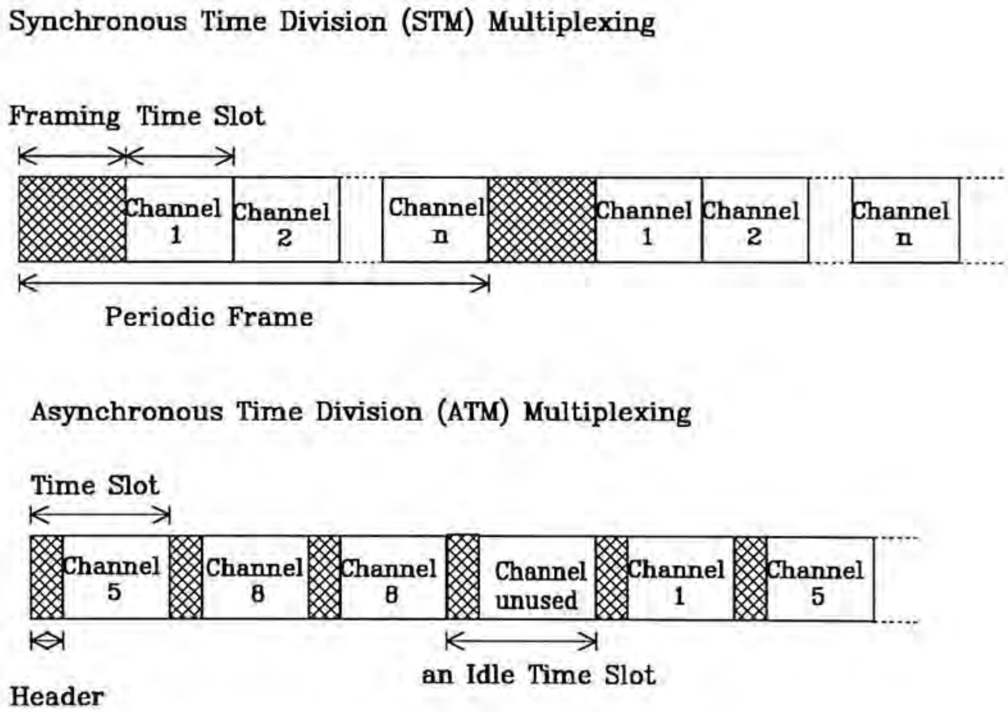


Figure 1.2 : STM and ATM

Desirable Features

1. Preserving the FIFO order of packets
2. Free from internal path conflicts and output conflicts
3. Broadcast / multicast capabilities
4. Supporting priority function
5. Supporting variable length record
6. Simple and small in size
7. high fault tolerance capability

Besides switching fabric, control algorithm, IC technologies etc, the basic ways to classify current electronic broadband switches are the buffering and switching techniques. For the switching techniques, space-division switching and address filtering are the most common methods. In the former case, transmitting switch element chooses between two or more physical links to route packets, such as the Crossbar and Banyan based switches (Figure 1.3). Complexity of Banyan switch is lower than that of the Crossbar switch because the number of crosspoints is reduced from $O(n^2)$ to $O(n \log n)$. However, extra control, say packet sorting by Sorter Network [BAT 68], is needed to solve the problem of internal conflict in Banyan switch. For address filtering, receiving switch element selects labelled data packets from a link carrying multi-destination traffic, such as the Broadcast Bus (Figure 1.4). Besides, some current designs apply centralized control technique to build switches. However, if the size of switch is large, distributed control is preferable.

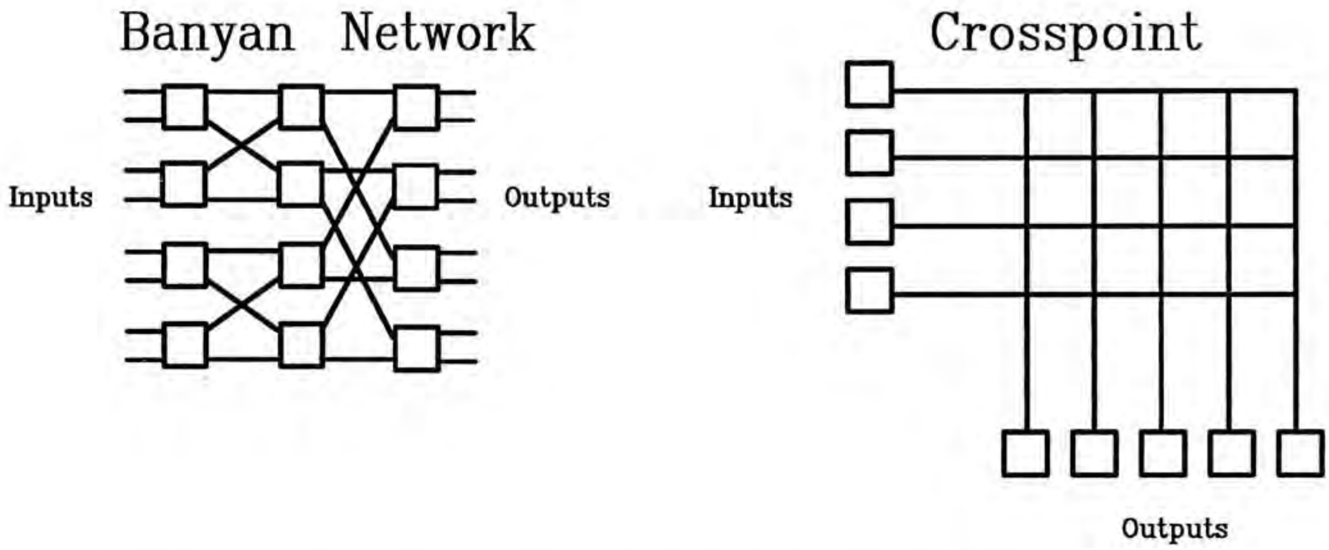


Figure 1.3 Space–Division Switching

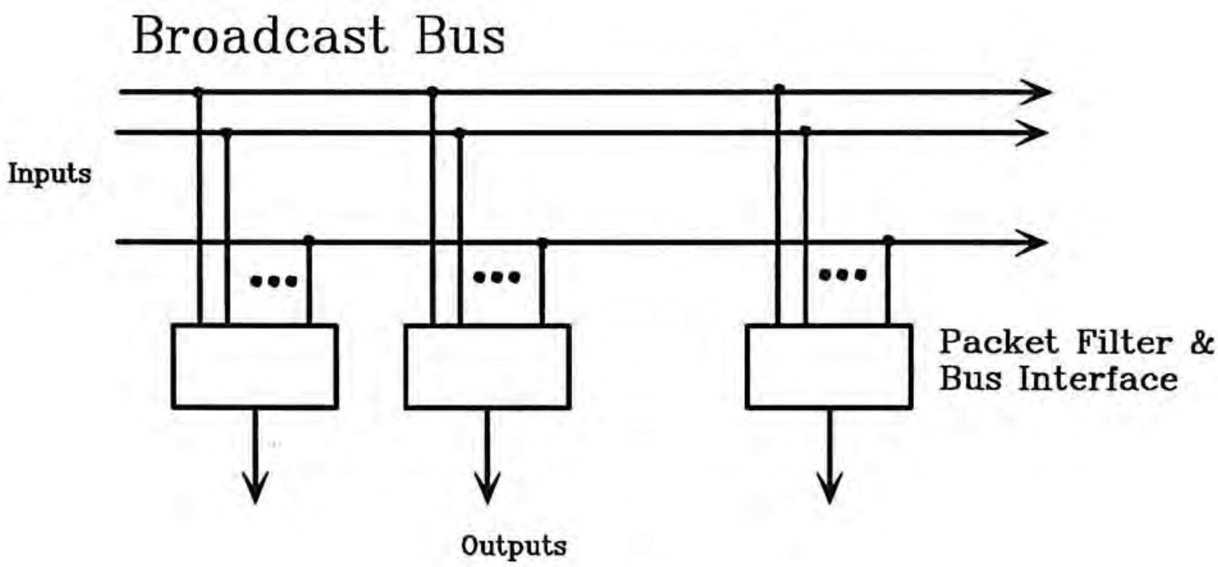


Figure 1.4 Address Filtering

Buffering is an essential means to deal with the problems of output conflict and internal blocking. Four classes of buffering techniques are shown in Figure 1.5, namely *input, output, crosspoint buffering* and *shared memory.* Undoubtedly, Crosspoint buffering consumes the largest buffer size as the number of I/O ports increases. On the contrary, buffering by shared memory has the smallest memory consumption. [KUW 89] found that the switches of shared buffer type require only 14% of the buffer size of the separated buffer type under the condition that cell loss probability is less than or equal to 10^{-9}, link utilization is equal to 0.8 and switch size is 32. However, shared buffer memory usually needs centralized switching control which will increase the hardware complexity if the size of switch grows. [KAR 87] found that the throughput by input buffering is limited to 58.6% with FIFO buffers. But switches with output buffering can achieves optimal throughput and delay performance. Tremendous amount of research work has been done recently on the design of electronic broadband switches and they are classified by the buffering and switching techniques as shown in Table 1.2.

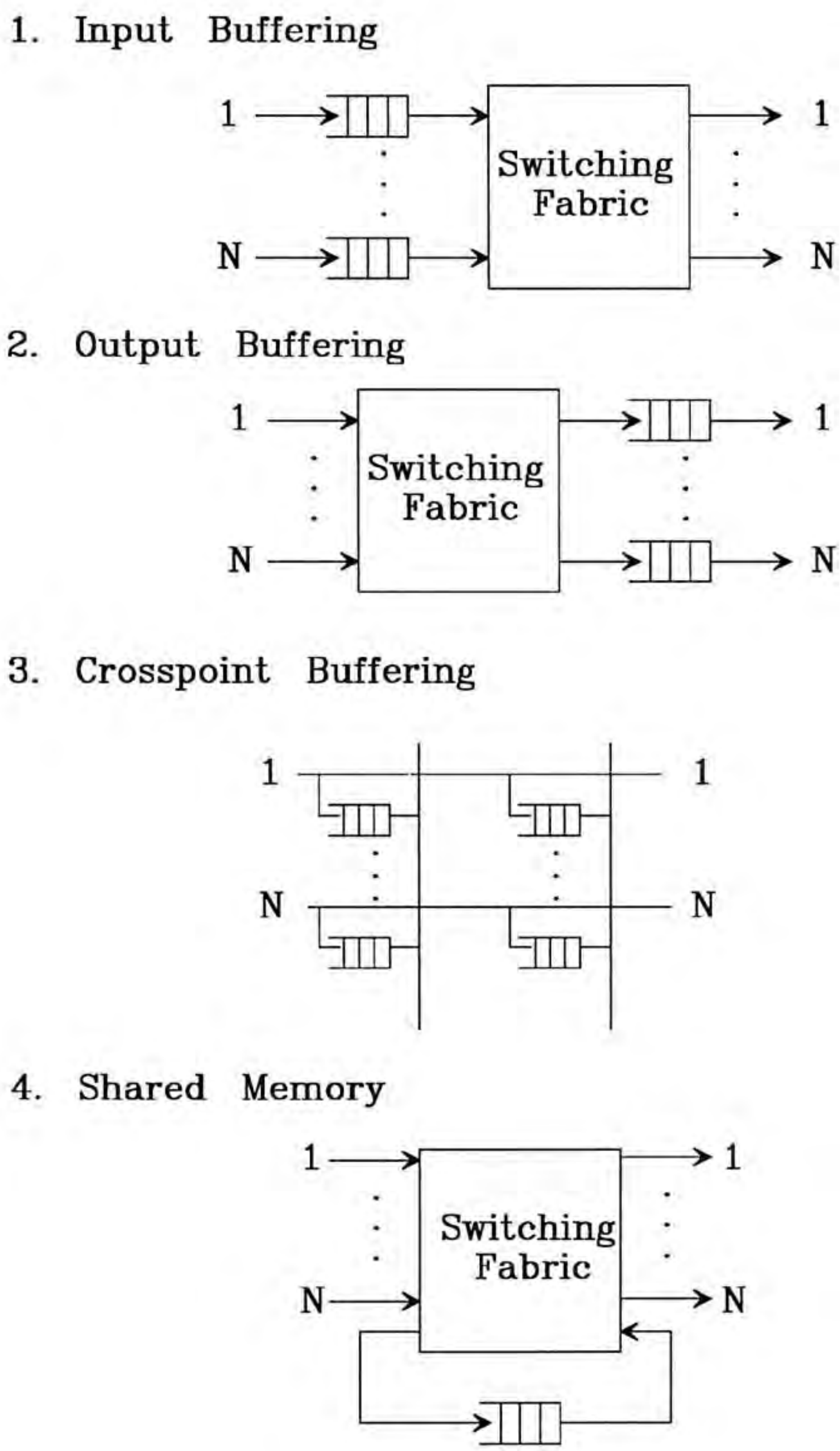


Figure 1.5 : Buffering Techniques

Table 1.2 Classification of current typical Broadband Switch Designs [1] :

Buffering Techniques	Space-Division Switching	Address Filtering	Centralized Control
Input Buffering	Nemawashi Switch [ARA 90] Startite [HUA 84] Batcher Banyan [MAR 90] Cambridge FP Switch [NEW 88]	Paris Switch [CID 88]	
Output Buffering	Integrated Switch [AHM 88] Tandem Banyan [TOB 91] Batcher Banyan [MAR 90] Cambridge FP Switch [NEW 88]	ATOM Switch [SUZ 89] Knockout Switch [YEH 87] ALCATEL Switch [CLO 91] Tandem Banyan [TOB 91] Paris Switch [CID 88]	
Crosspoint Buffering	MSSR [HAJ 88] CMOS Crosspoint [AKA 90] Buffered Banyan [LAU 90]	Bus Matrix Switch [NOJ 87]	
Shared Memory			Hitachi's Switch [KUW 89] Toshiba's Switch [TAN 91] Prelude Switch [DEV 88]

[1]Switch designs may appear more than once if more than one techniques are applied.

Chapter 2

Management of Control Information

The topography of the packet communication network involves the following parameters. There are N_{lk} one-directional links connecting N_{nd} nodes. Some of the nodes are office nodes, where packets originate from and terminate at; others are simply connection points. Each node owns up to a number of links. From each node, there are a number of different routes leading to other nodes. A link is logically divided into a data channel and a small control channel. The latter is for the exchange of control information, which is normally the link congestion status.

The scheme of traffic information exchanges over control channels and its application on traffic flow control are depicted in the following Figure 2.1. Periodically in every δ (= 12, say) microseconds, every node assess the congestion status of its own links based upon traffic conditions. The assessment mechanism will be detailed in Section 3.4. The congestion status is stored in the *link*

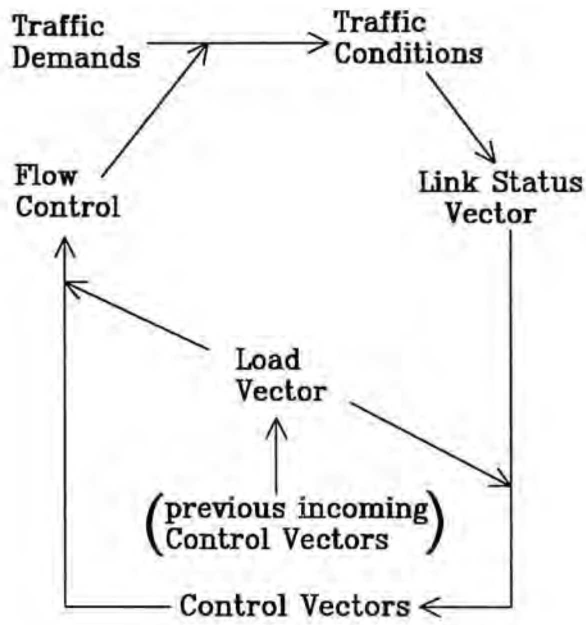


Figure 2.1 Relationships among elements of traffic control and the applications of traffic control

status vector, in which each bit represents a link. The node then broadcasts the *control vector* to neighbouring nodes. The control vector is the combination of the link status vector with the *load vector*. Recursively, the load vector contains information extracted from control vectors previously received from neighbouring nodes. This information extraction relies on static *masks* and *route vectors* that characterize the topography of the network. The load vector represents the node's best knowledge about the congestion status of remote links, and the control vector represents the node's best knowledge about the congestion status of the whole network. These two vectors are used in the control of traffic flow described in Chapter 3. Through the control of traffic flow, the two vectors feed back to the traffic conditions.

Congestion status established is stored in an N_{lk}-bit link status vector which represents the set of all congested links of this node. A bit 1 stands for *congested* and a 0 for *non-congested.* Assume that there are n outgoing links in node s and the links are labelled as Link $k, k+1, \ldots, k+n-1$. Let $L_{s,t}$ be the link status vector of the outgoing links of node s at time t,

$$L_{s,t} = [0 \ldots 0 \; l_k l_{k+1} \ldots l_{k+n-1} \; 0 \ldots 0]$$

$$\text{where } l_k = \begin{cases} 1 & \text{if link status of link } k \text{ is congested} \\ 0 & \text{otherwise} \end{cases}$$

The normal function of a control packet is to communicate information about link congestion. In every δ microseconds, every node broadcasts a control packet to all adjacent nodes through outgoing control channels. The control packet contains a vector representing the set of all links that, according to the current assessment of the broadcasting node, are congested. Thus every node also receives a vector from every incoming control channel. The most updated congestion status of a remote link received by a node only comes through the incoming channel which lies on the shortest path from the owner of the remote link to the node.

The format of a control packet includes a control vector and a category field. Each bit position in the control vector corresponds to a link in the network. The category field specifies the meaning of the vector, e.g., category 0 interprets the vector as the link congestion status, category 1 interprets it as the link abnormality status, etc. Load vector has the same format as the control vector except that it does not consist of link information from the link status vector.

Let $C_{s,t}$ be the control vector of any node s at time t,

$$C_{s,t} = [S_1 S_2 S_3 \ldots S_{N_{lk}}]$$

$$\text{where } S_k = \begin{cases} 1 & \text{if link } k \text{ is congested} \\ 0 & \text{otherwise} \end{cases}$$

and S_k represents link status of link k for $k = 1, \cdots, N_{lk}$.

When a packet requests entry into the network at an office node, the addressing information stored at the packet header is translated into the set of route. This translation is by a static routing table stored in a high-speed memory at the node, and its result is in the form of an N_{lk}-bit Route Vector.

Let $R_{s,d}$ be the Route Vector from node s to d

$$R_{s,d} = [r_{sd,1} r_{sd,2} \dots r_{sd,N_{lk}}]$$

$$\text{where } r_{sd,k} = \begin{cases} 1 & \text{if the route from node s to d includes link k} \\ 0 & \text{otherwise} \end{cases}$$

Static data at a node relates each remote link to the incoming channel that lies on the shortest path from the owner of that remote link to the local node. This data is stored in the form of an N_{lk}-bit vector mask for each incoming control channel, which will be referred to as the mask of *who knows best.* Moreover, the node stores the latest link status of its own outgoing links, thus its masks should mask off these link status from the control vectors.

Let $M_{i,j}$ be the vector mask of the incoming link from node i to j.

$$M_{i,j} = [m_1 \dots m_{N_{lk}}]$$

$$= \bigvee_{\substack{\forall\ k\ s.t.\ r_{jk,l}=1 \\ \text{where } l \text{ is the outgoing link from node } j \text{ to } i}} R_{j,k}$$

$$\text{where } m_k = \begin{cases} 1 & \text{mask on} \\ 0 & \text{mask off} \end{cases}$$

2.1 Inter-node Exchange of Link Congestion Status

The control vector received from each channel will be logically ANDed with the mask of *who knows best* for that channel. The AND results for all incoming channels are then ORed together to form the Load-vector, which represents the set of congested remote links and will be used in the control of passing-through packets. This vector will further be ORed with the vector representing the set of congested links of the local node to form the control vector, which represents the set of all congested links and will be used in the control of influx to the network. The control vector is then loaded into the next control packet for broadcasting to neighbouring nodes.

Eqn.(2.1) depicts the operation in generating outgoing control vectors from incoming control vectors and link status vector of a node s with neighouring nodes labelled as $1, 2, 3, \ldots, n$. Figure 2.2 shows the hardware realization of the above operation. Assume that nodes are synchronized to send control vectors at δ interval and δ is greater than the propagation delay of any link in the network.

$$C_{s,t} = L_{s,t} \vee (C_{1,t-\delta} \wedge M_{1,s}) \vee (C_{2,t-\delta} \wedge M_{2,s}) \vee \ldots \vee (C_{n,t-\delta} \wedge M_{n,s})$$

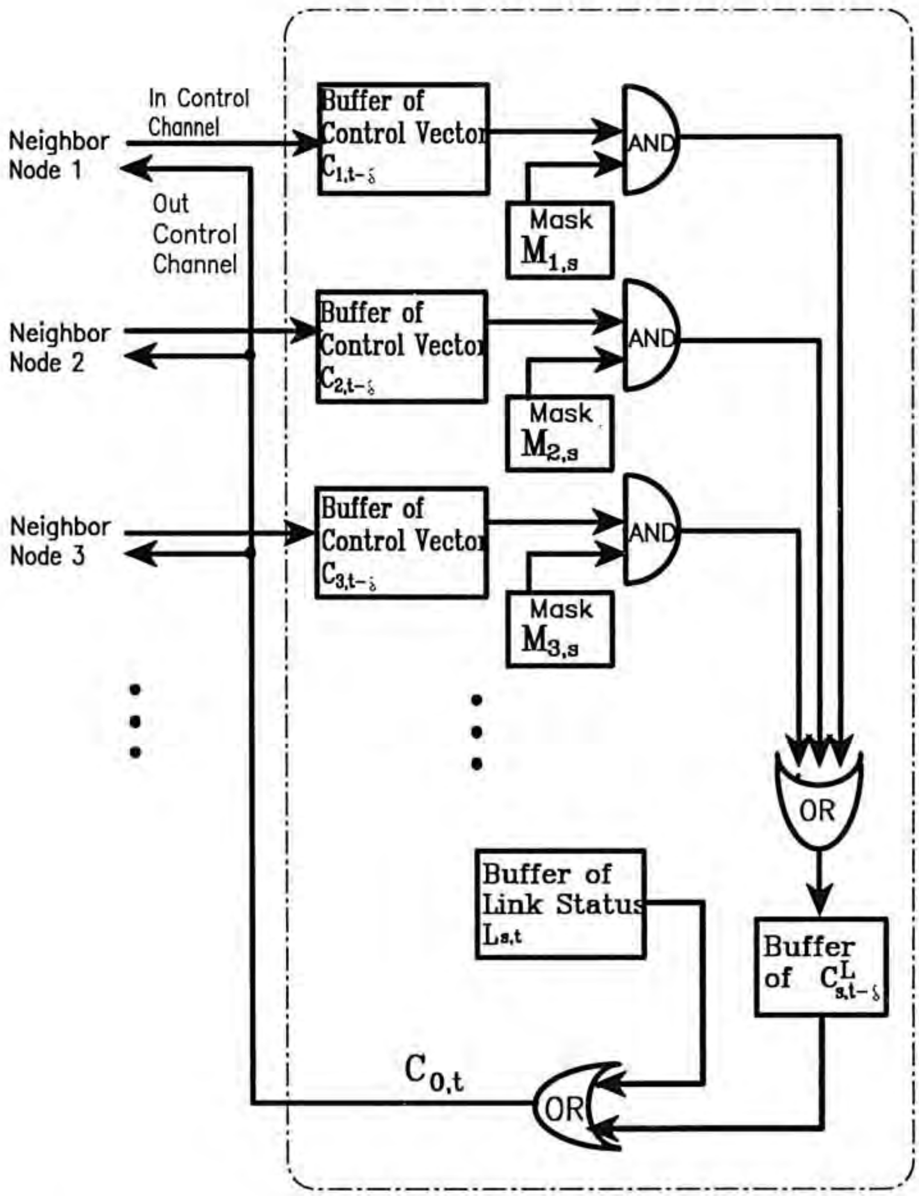


Figure 2.2 Operation at a node in generating Control Vector

$$= L_{s,t} \vee C^L_{s,t} \tag{2.1}$$

where $C^L_{s,t}$ is the *load vector* of node s at time t.

Similarly, neighouring nodes of the node n are $n\_1, n\_2, \ldots, m_n$.

$$\begin{aligned}
C_{1,t-\delta} &= L_{1,t-\delta} \vee (C_{s,t-2\delta} \wedge M_{s,1}) \vee \ldots \vee (C_{m_1,t-2\delta} \wedge M_{m_1,1}) \\
C_{2,t-\delta} &= L_{2,t-\delta} \vee (C_{s,t-2\delta} \wedge M_{s,2}) \vee \ldots \vee (C_{m_2,t-2\delta} \wedge M_{m_2,2}) \\
&\vdots \\
C_{n,t-\delta} &= L_{n,t-\delta} \vee (C_{s,t-2\delta} \wedge M_{s,n}) \vee \ldots \vee (C_{m_n,t-2\delta} \wedge M_{m_n,n})
\end{aligned} \tag{2.2}$$

substitute Eqn.(2.2) into (2.1),

$$\begin{aligned}
C_{s,t} = L_{s,t} \; &\vee[(L_{1,t-\delta} \wedge M_{1,s}) \vee (C_{s,t-2\delta} \wedge M_{s,1} \wedge M_{1,s}) \vee \ldots \vee (C_{m_1,t-2\delta} \wedge M_{m_1,1} \wedge M_{1,s})] \\
&\vee[(L_{2,t-\delta} \wedge M_{2,s}) \vee (C_{s,t-2\delta} \wedge M_{s,2} \wedge M_{2,s}) \vee \ldots \vee (C_{m_2,t-2\delta} \wedge M_{m_2,1} \wedge M_{2,s})] \\
&\vdots \\
&\vee[(L_{n,t-\delta} \wedge M_{n,s}) \vee (C_{s,t-2\delta} \wedge M_{s,n} \wedge M_{n,s}) \vee \ldots \vee (C_{m_n,t-2\delta} \wedge M_{m_n,1} \wedge M_{n,s})]
\end{aligned} \tag{2.3}$$

now define :

$$\begin{aligned} L_{i,t}^{-1} &= L_{i,t} \wedge M_{i,s} \\ L_{i,t}^{-2} &= L_{i,t} \wedge M_{i,j} \wedge M_{j,s} \end{aligned} \tag{2.4}$$

Then Eqn.(2.3) can be simplified by Eqn.(2.1) and (2.4),

$$\begin{aligned} C_{s,t} &= L_{s,t} \vee \\ & L_{1,t-\delta}^{-1} \vee L_{2,t-\delta}^{-1} \vee \ldots \vee L_{n,t-\delta}^{-1} \vee \\ & L_{11,t-2\delta}^{-2} \vee L_{12,t-2\delta}^{-2} \vee \ldots \vee L_{m_1,t-2\delta}^{-2} \vee \\ & L_{21,t-2\delta}^{-2} \vee L_{22,t-2\delta}^{-2} \vee \ldots \vee L_{m_2,t-2\delta}^{-2} \vee \\ & \vdots \\ & L_{n1,t-2\delta}^{-2} \vee L_{n2,t-2\delta}^{-2} \vee \ldots \vee L_{m_n,t-2\delta}^{-2} \vee \\ & \vdots \end{aligned} \tag{2.5}$$

Remark. From Eqn.(2.5) it can be found that the control vector of node s at time t is made up of the current local link status and the remote link status in the past moments $t-\delta, t-2\delta, t-3\delta, \ldots$ according to the distance away from the remote nodes.

2.2 Consistency of Control Information

Through the constant control information exchanges, every node acquires the congestion status of all links. There may possibly be contradictions among information from different sources due to different time lags. With the aid of the vector masks characterizing the topography of the network, each node extracts from the ensemble of received information the most updated status of each remote link. To obtain consistent control information from the control vector and load vector, the vector masks of any node in the network have to satisfy the following conditions.

1. Vector masks of the incoming links of the node are mutually exclusive.
2. Link status of the node's outgoing links are masked off by the vector masks because the latest link status stores at its link status vector.

Theorem 2.1 : The consistency of control information from control vector and load vector of any node in the network is guaranteed.

Proof : For the node s with neighbouring nodes labelled as $1, 2, \ldots, n$, condition 1 implies that,

$$\begin{aligned} M_{i,s} \wedge M_{j,s} &= [00\ldots0] \\ &= \bar{O} \qquad \text{for } i,j = 1,2,\ldots,n \text{ and } i \neq j \end{aligned} \tag{2.6}$$

then,

$$(C_{i,t} \wedge M_{i,s}) \wedge (C_{j,t} \wedge M_{j,s}) = \bar{O} \qquad \text{for } i,j = 1,2,\ldots,n \text{ and } i \neq j \tag{2.7}$$

therefore, information consistency of load vector is guaranteed. In addition, condition 2 implies that,

$$L_{s,t} \wedge M_{i,s} = \bar{O} \qquad \text{for all t and } i = 1,2,\ldots,n \tag{2.8}$$

then,

$$L_{s,t} \wedge (C_{1,t-\delta} \wedge M_{1,s}) \wedge (C_{2,t-\delta} \wedge M_{2,s}) \wedge \ldots \wedge (C_{n,t-\delta} \wedge M_{n,s}) = \bar{O} \qquad \text{for all t}$$

thus, the control vector $C_{s,t}$ contains no inconsistent information.

2.3 Alternate Format of Control Information

Instead of sending and receiving control vectors in fixed length and mask off the irrelevant bits by the vector masks at the receiving side, only the essential bits are sent and received, for example the control vector consists of link status of the links within N_{hp}-hops. An alternative format of the control vector is for a node to send to each neighbour the congestion status of only those links that the particular neighbour wants to learn from this node. The substantial reduction on the bandwidth of the control channels is at the cost of computational complexity. For the alternative format, the size of the link status vector is different from that of the incoming control vectors. The generation of outgoing control vectors by the incoming control vectors and the link status vector is not as straightforward as before.

The control vector of a node is generated by concatenation of its link status vector and the incoming control vectors as shown in Eqn. (2.9). The outgoing control vectors are extracted and sent to the neighbouring nodes, it will then become the incoming control vectors of the neighbouring nodes. Figure 2.3 suggests a method for hardware implementation of this process. The local link status and the incoming control vectors are the inputs of a Programmable Array Logic (PAL) which has been preprogrammed. The outgoing control vectors are

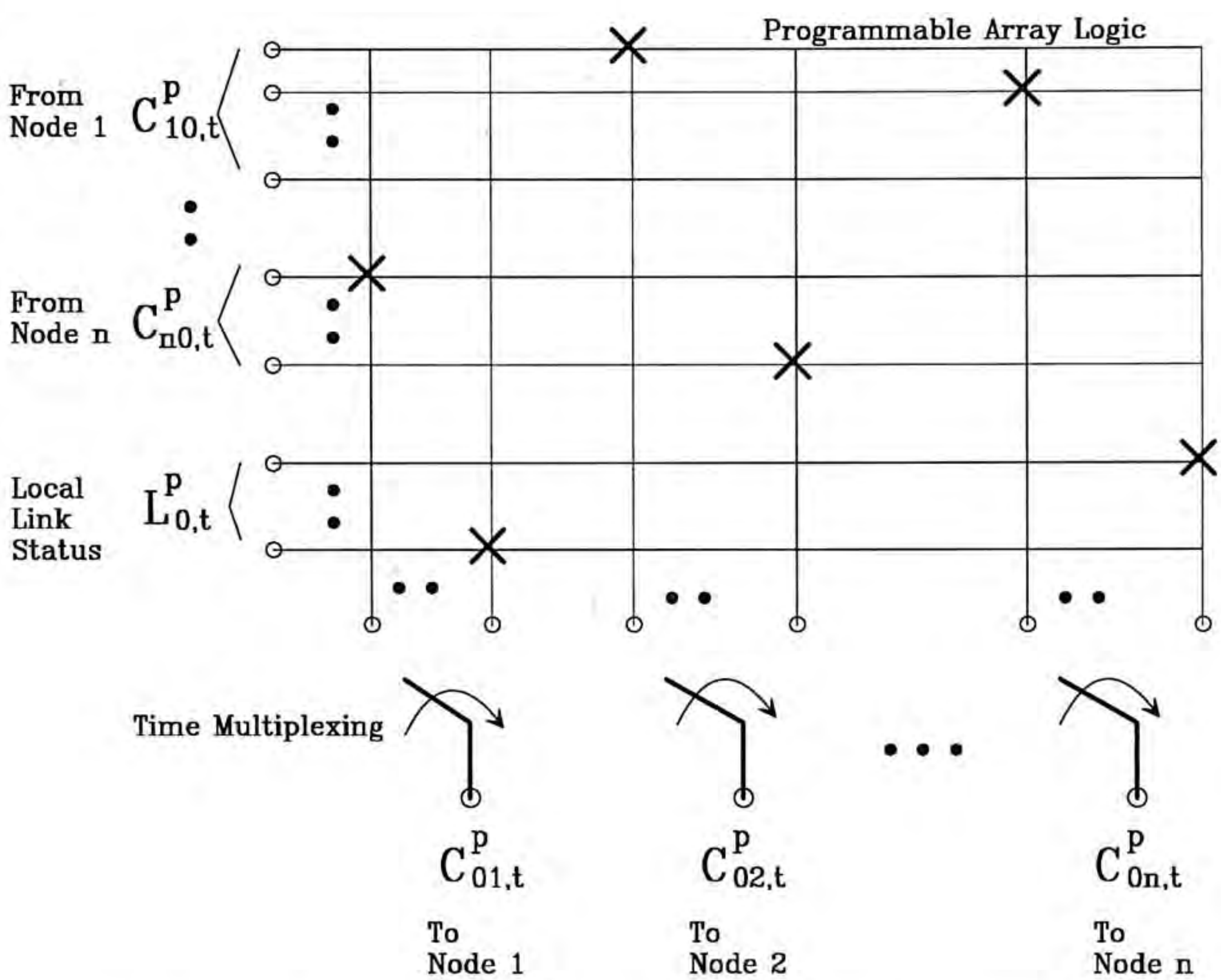


Figure 2.3 Receving and Despatching Variable–Length Control Vectors

then extracted from the output lines by time multiplexing. The computation of these vectors is done in every δ microseconds.

The control vector of node s at time t is defined as :

$$C_{s,t} = L^{p}_{s,t} \oplus C^{p}_{1s,t-\delta} \oplus C^{p}_{2s,t-\delta} \oplus \ldots \oplus C^{p}_{ns,t-\delta} \tag{2.9}$$

where $L^{p}_{s,t} = [l_k l_{k+1} \ldots l_{k+n-1}]$,

$\oplus$ is the operator which means concatenation of the control vectors.

$C^p_{sk,t}$ is the incoming variable-length control vector from node s to k at time t and it only consists of link status of the related links. The conditions for control information consistency is similar to that of the fixed-length control vector.

c.f. Eqn. (2.6) : $C^p_{jk,t}$ and $C^p_{kj,t}$ (for $j \neq k$) consist of completely distinctive link status information.

c.f. Eqn. (2.8) : $L^p_{s,t}$ and $C^p_{jk,t}$ consist of completely distinctive link status information.

Chapter 3

Traffic Flow Control

In this chapter, a framework for traffic flow control in broadband network is described. The objective of the packet-level control is to achieve high speed performance, say sub-microsecond execution time, by simple hardware design which can cope with the high transmission and switching speed requirement in BISDN. Firstly, the traffic control for the packets of connectionless communications is introduced in Sections 3.1 and 3.2. Then the traffic control of connection oriented calls, which relates closely with the call set up algorithm, is proposed in Section 3.3. The properties of being starvation-free and deadlock-free of the traffic control are also presented.

3.1 Control of Traffic Influx into the Network

When a packet of connectionless communications requests entry into the network at an office node, if it passes the *influx vector screening*, then the packet is admitted into the network. Otherwise, the entry request is denied. Some capability of customer interface is assumed here. When a new packet is denied entry to the network, it is either buffered or to be regenerated and, meanwhile, the packet source is notified of such traffic situation. It is then up to the node function of route selection to decide whether to shift to an alternative route upon reattempt.

Let $O_{sd,t}$ be the result of influx vector screening to the route from node s to d at time t,

$$O_{sd,t} = \begin{cases} \bar{O} & \text{Pass the influx vector screening} \\ \text{otherwise} & \end{cases}$$

For the traffic control of the connectionless packets, the influx vector screening from node s to d at time t is shown as below.

$$O_{sd,t} = R_{s,d} \wedge C_{s,t'} \tag{3.1}$$

$$\text{where } t' \leq t < t' + \delta \quad \text{and} \quad t' \text{ is a multiple of } \delta$$

3.2 Control of Traffic Loading from the Node

Packets that have arrived at a node via incoming links and those that are locally generated (and admitted into the network) are all routed through an ATM packet switch. The switch has at least N_{og} inputs and N_{rt} outputs. Outputs of the switch are in one-to-one correspondence with routes. It is also assumed that the switching capacity has been engineered to sufficiently compensate for the delay due to output contention in switching [KAR 87], [LI2 90]. If the route from node s to d corresponding to a queue (Q_{sd}) passes the *load vector screening*, then the head-of-line packet of the queue is loaded onto the outgoing link.

The result of load vector screening determines whether the packets in the queue can load onto the outgoing link for transmission. Let $I_{sd,t}$ be the result of load vector screening to the route from node s to d at time t,

$$I_{sd,t} = \begin{cases} \bar{O} & \text{Pass the load vector screening} \\ \text{otherwise} & \end{cases}$$

Similar to influx vector screening, the load vector screening of packets for packet-level control is,

$$I_{sd,t} = R_{s,d} \wedge C^L_{s,t'} \tag{3.2}$$

$$\text{where } t' \leq t < t' + \delta \quad \text{and} \quad t' \text{ is a multiple of } \delta$$

and

$$C^L_{s,t'} = (C_{1,t'-\delta} \wedge M_{1,s}) \vee (C_{2,t'-\delta} \wedge M_{2,s}) \vee \ldots \vee (C_{n,t'-\delta} \wedge M_{n,s}) \tag{3.3}$$

The neighbouring nodes of node s are node $1, 2, \ldots, n$. $C^L_{s,t'}$ is the load vector of node s at time t', it is the same as $C_{s,t'}$ except that the former do not consist of the link status information of the local links. The queue Q_{sd} is said to be *flowing* at time t if $I_{sd,t} = \bar{O}$.

3.3 Flow Control for Connection Oriented Traffic

The flow control of packet streams of the connection oriented call is by the call set-up process rather than at the packet level. If the *priority class* of a data packet indicates that the packet belongs to the connection oriented call, it is unconditionally admitted. Packets are classified into three categories by their priority classes : High Priority (HP) packets are the packets of connection oriented call, scout packets have intermediate priority, and Low Priority (LP) packets are the packets of connectionless communications. Priority here means the loading priority of packets from buffers or queues onto the outgoing links. HP packets carry real-time information from connection oriented calls which cannot suffer from long delay but small packet loss is acceptable, like video and voice packets. A scout packet has the same format as a data packet. Its payload field has null contents, but the header includes an indication of the scout identity plus the addressing information. Thus, scout counters can simply stores the number of scout packets that are sending to the same destination and scout packets can be reconstructed when they are loading onto the outgoing link. Scout packets are generated by the sender of a call at the time of call set-up. Lastly, LP packets store information of connectionless communications which is insensitive to delay time but intolerable to packet loss, like the electronic mail.

For the sake of call set up, a stream of N_s scout packets with inter-arrival time p_j (for call type j) are injected into the network. Every scout packet has to pass the influx vector screening to set up the call. When a call of type j is testing the availability of a route from node s to d at time t by a stream of scout packets, if $\delta \geq p_j$ (see Figure 3.1), the result of influx vector screening will be :

$$O_{sd,t+n\cdot p_j} = R_{s,d} \wedge C_{s,t'} \tag{3.4}$$

$$\text{where } t' \leq t + n \cdot p_j < t' + \delta,\ n = 0, 1, 2, \ldots, N_S, \ \text{ and } \ t' \text{ is a multiple of } \delta$$

Since the control vector updates periodically with time interval δ, the value of $C_{s,t'}$ only keeps constant at time interval $[t', t'+\delta)$. If the congested link status in $C_{s,t'}$ is reset in $C_{s,t'+\delta}$, see Figure 3.2, then the influx vector screening cannot work properly when $\delta < p_j$. It is because the control vector is updated too soon for the sender of HP call to stop its scout packet by influx vector screening. To remedy this drawback, the influx vector screening have to consider not only the current control vector but also the control vectors at previous multiple δ time periods (Eqn.(3.5)).

$$\begin{aligned} O_{sd,t+n\cdot p_j} &= R_{s,d} \wedge (C_{s,t'} \vee C_{s,t'-\delta} \vee \ldots \vee C_{s,t'-(k-1)\delta}) \\ &= R_{s,d} \wedge (\bigvee_{i=0}^{k-1} C_{s,t'-i\delta}) \\ &\quad \text{where } k = \left\lceil \tfrac{p_j}{\delta} \right\rceil \\ &\quad \text{and } t' \leq t + n \cdot p_j < t' + \delta,\ n = 0, 1, 2, \ldots, N_S \end{aligned} \tag{3.5}$$

Remarks.

1. In practice, it is difficult to determine the value of k that suit various types of call in real-time. To simplify the hardware complexity,

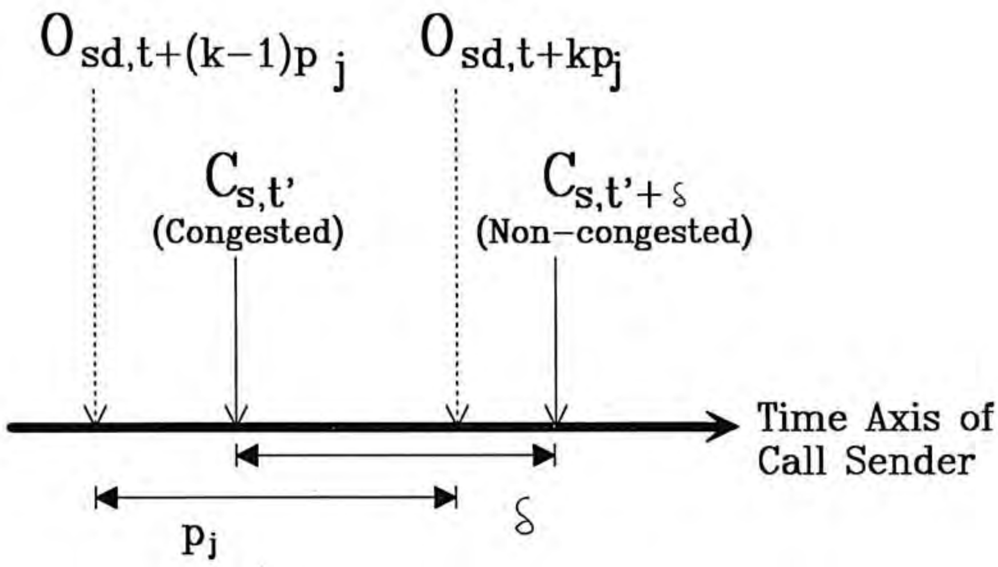


Figure 3.1 Case when $\delta \geq p_j$

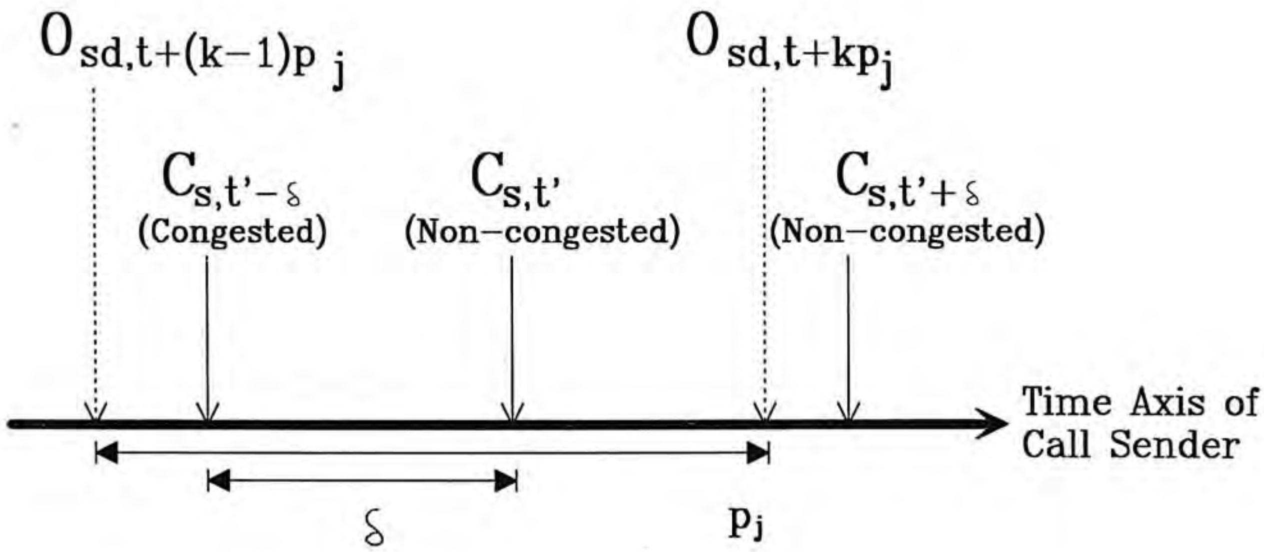


Figure 3.2 Case when $\delta < p_j$

$$k = \left\lceil \frac{Max(p_1, p_2, \ldots)}{\delta} \right\rceil \tag{3.6}$$

where $Max(x_1, x_2, \ldots)$ denotes the maximum value among $x_1, x_2, \ldots$,
$\lceil x \rceil$ denotes the smallest integer not less than x.

The hardware realization of generating $O_{sd,t+k\cdot p_j}$ is shown in Figure 3.3. The shift register, buffer of route vector and the control vectors are in the same length. However it should be noted that the above value of k makes the calls with the larger bandwidth (ie. smaller p_j) more vulnerable to call rejection.

2. Adaptive routing at call set-up level is possible by using alternate route vectors $R_{s,d}$ for influx vector screening. Simple adaptive routing algorithms can be adopted. For example, choose the routes with links that are least frequently marked as congested within a recent period of time.

For an office node (Figure 3.4), the output of the switch corresponding to the null route is connected to facilities that handle arriving packets. Every other output leads to a multiplexer for HP packets and a queue for LP packets. Each multiplexer corresponds to an outgoing link and outputs to an buffer that leads to the load controller of that outgoing link. The switch outputs LP packet through a queue to the load controller of the proper outgoing link. The load controller will load packets from the buffer of HP packets and from those queues whose corresponding routes pass the *load vector screening.*

A buffer for HP packets plus a number of queues for LP packets are connected

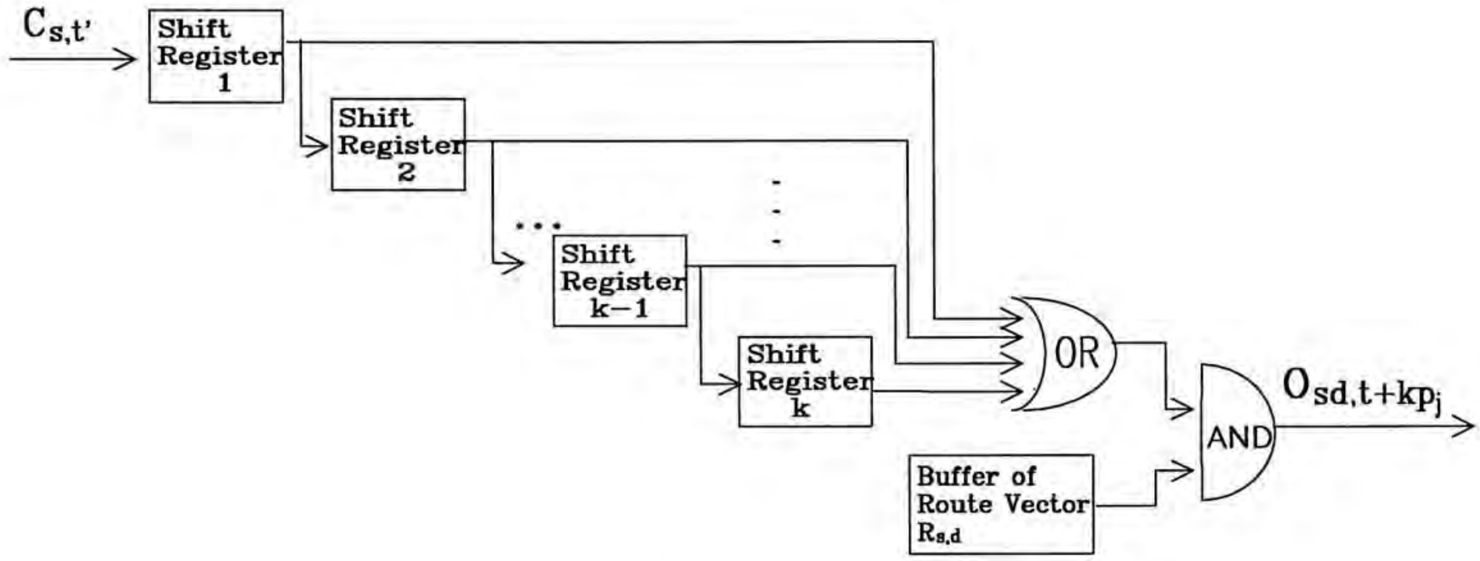
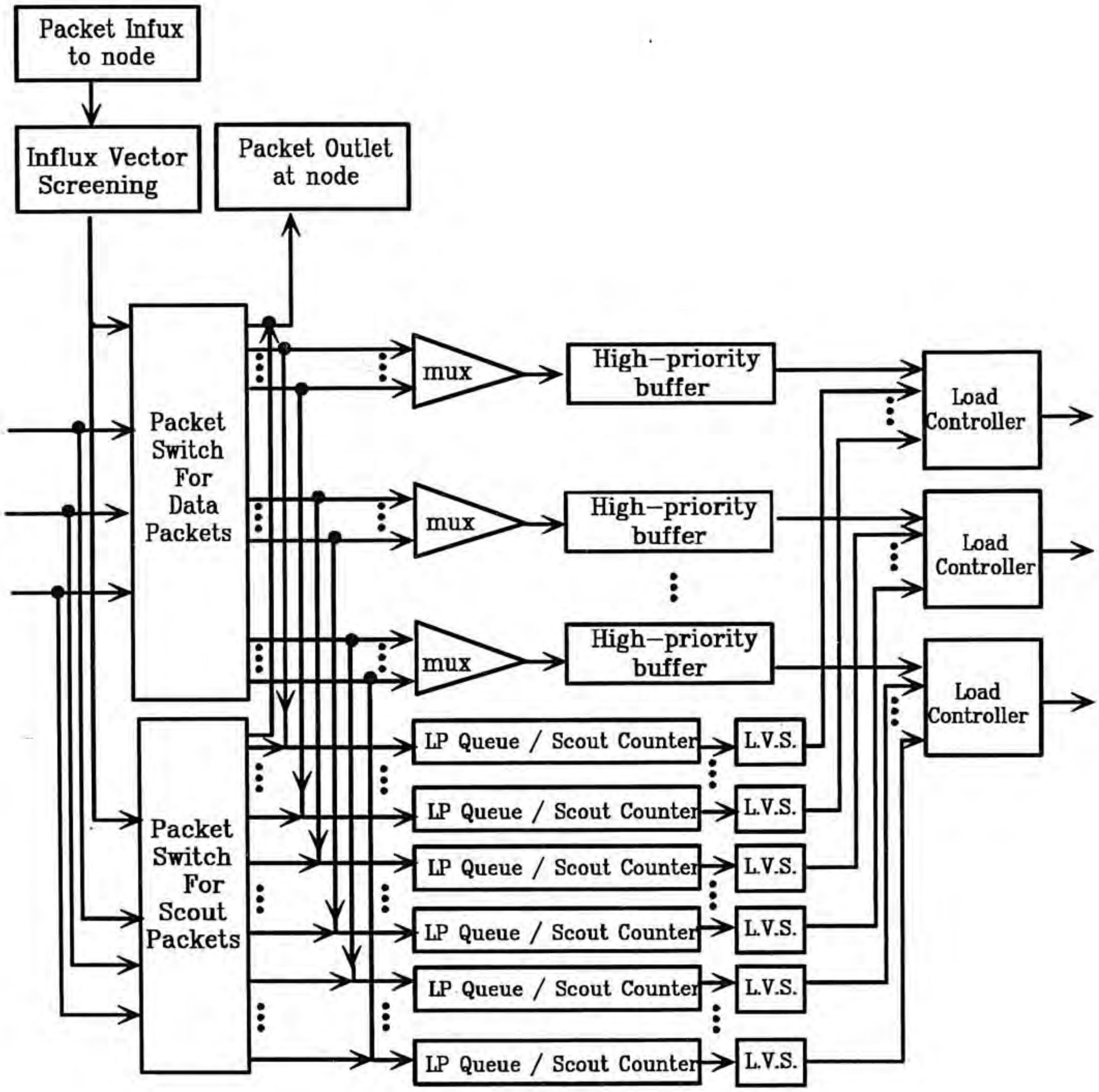


Figure 3.3 Hardware Realization of Influx Vector Screening

to an outgoing link through the load controller. In each cycle of its operation, the load controller completely unloads the HP buffer and then scans through the queues. If the HP buffer is empty, the HP packets can cut through the buffer directly [KER 79]. If the route corresponding to a queue passes the load vector screening, then the head -of-line packet of the queue is loaded onto the outgoing link. The definition of the head-of-line packet incorporates the consideration of the priority class in the case when the queue contains packets of multiple (low) priority classes. There are two packet switches for each node, one is dedicated to data packet switching and the other is used for scout packet switching.

For the load vector screening of scout packets,

$$\begin{aligned} I_{sd,t+n\cdot p_j} &= R_{s,d} \wedge (C^L_{s,t'} \vee C^L_{s,t'-\delta} \vee \ldots \vee C^L_{s,t'-(k-1)\delta}) \\ &= R_{s,d} \wedge (\bigvee_{i=0}^{k-1} C^L_{s,t'-i\delta}) \end{aligned} \tag{3.7}$$



L.V.S. : Load Vector Screening

Figure 3.4 Internal Structure of a Switching Node where $k = \left\lceil \frac{p_j}{\delta} \right\rceil$

and $t' \leq t + n \cdot p_j < t' + \delta$, $n = 0, 1, 2, \ldots, N_S$

The method of hardware realization of Eqn.(3.7) is same as Figure 3.3 except the input is $C^L_{s,t'}$.

For each loading cycle of the load controller, HP packet has the highest priority for loading. If there is no HP packet in the HP buffer, load vector screening will be executed. Scout and LP packets have to pass the load vector screening before loading onto the link. If both a scout and a LP packet of the same route are available, they will be superimposed for transmission because scout packet contains no information payload. If the values of scout counters of all the routes shows zeros, then LP queue s are examined. The order of scanning LP queues and scout counters are cyclical. If any scout packet do not pass the load vector screening, the scout count will be reset. Figure 3.5 depicts the above loading discipline of load controller at every loading cycle by a flowchart.

3.4 Judgement of Link Status

The load controller of an outgoing link scans through those queues and the high-priority buffer that are connected to it. In each loading cycle, load controller loads as many packets as possible from the high priority buffer and also loads the head-of-line packet from every queue that passes the load vector screening.

The status of a link is generated from the output signals of *low priority queue count scanners* (LN_{sd}, LC_{sd}) and *scout count scanners* (SC_{sd}) of the route from

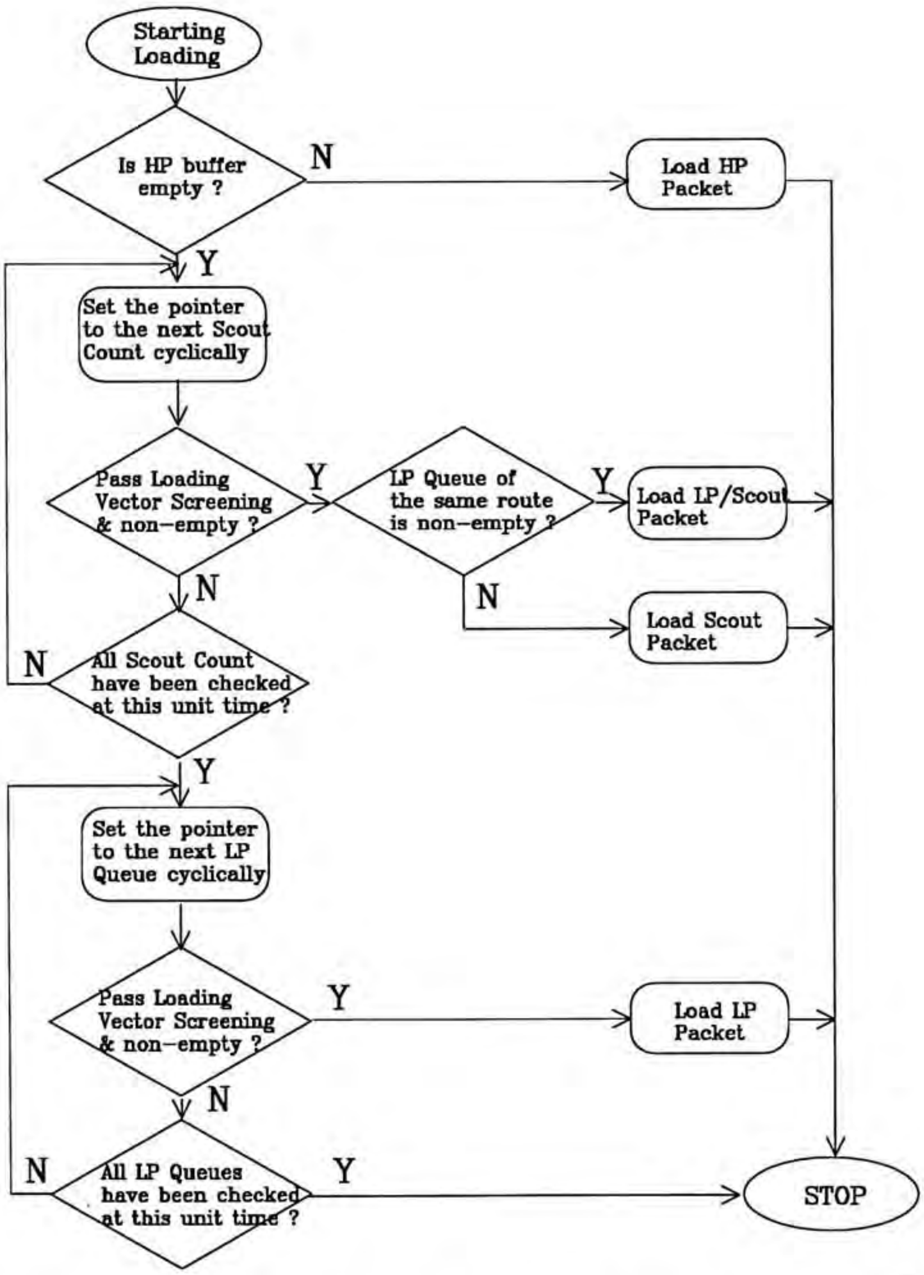


Figure 3.5 : Loading Discipline of Load Controller at every loading cycle

node s to d (Figure 3.6). LP queue count scanner checks the queue size of the LP queue of the corresponding route periodically and scout count scanner checks the number of *detecting scout packets* of the route periodically. The function of the detecting scout packet is discussed in Section 4.1.

In addition, it is important to synchronize link status with the control vector interval (Figure 3.7) because control vectors are updated and sent out every δ period. The link status sets to high level for a multiple of δ period and keeps constant at any time interval $[t, t+\delta)$.

The link status is congested if any LP packet count is greater than or equal to the upper threshold level for LP packets or any detecting scout count is greater than or equal to the threshold level for detecting scout packets. On the other hand, if no flowing queue such that its LP packet count is greater than or equal to the lower threshold level for LP packets, then the link status is updated as non-congested.

Let N_{UT} be the upper threshold level for LP queues,
N_{LT} be the lower threshold level for LP queues,
N_T be the threshold level for detecting scout packet count.

The output signals of the scanners react at the following conditions :

1. LN_{sd} is set if the queue is flowing and the LP packet count $< N_{LT}$ <u>or</u> the queue is not flowing

2. LC_{sd} is set if the LP packet count $> N_{UT}$

3. SC_{sd} is set if the DS packet count $> N_T$

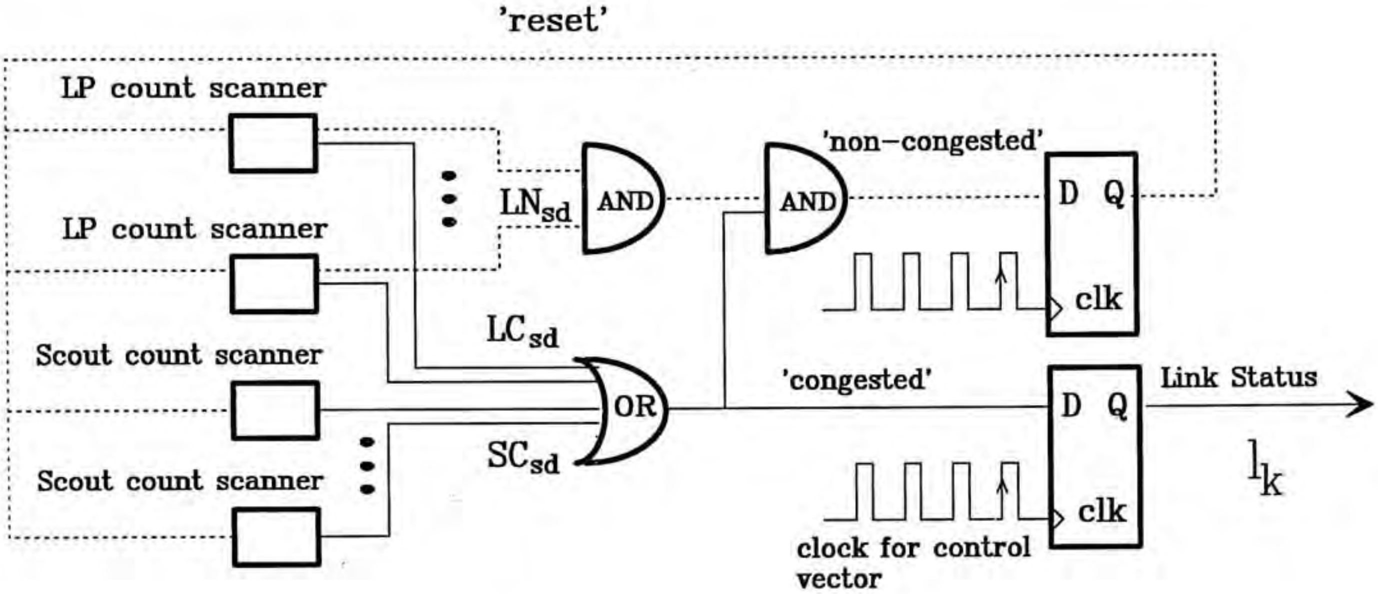


Figure 3.6 Generation of Link Status

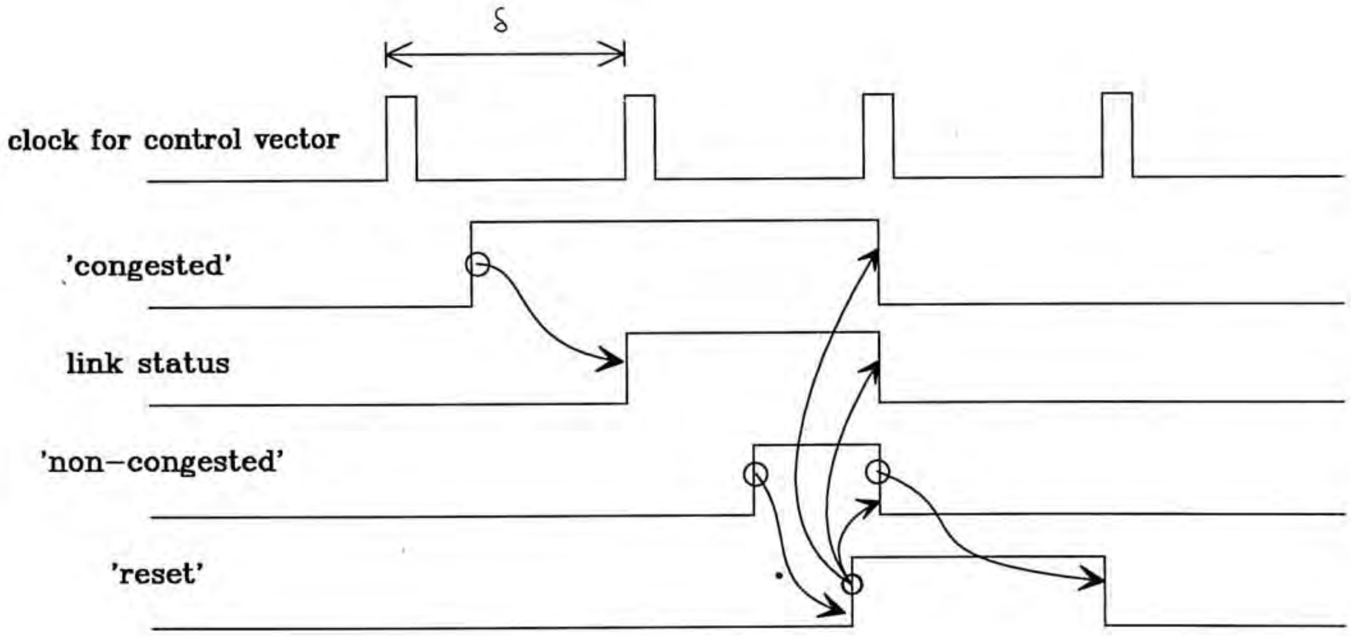


Figure 3.7 Timing Diagram

4. LN_{sd}, LC_{sd} *and* SC_{sd} are reset when the 'reset' signal is set.

5. The link status is set if either one of the signals LC_{sd} or SC_{sd} is set but it can only be reset if all LN_{sd} are reset.

3.5 Starvation-free and Deadlock-free

Without carefully planned scheme of congestion control, some unexpected and undesirable situations may be arose, such as *starvation* and *deadlock*. If these problems arise, there will be great impact on network performance. Instead of exercising extra detecting and recovery algorithms at the time of Starvation and Deadlock, it is more desirable to have a congestion control mechanism that inherits the Starvation-free and Deadlock-free properties.

Definition 3.1 : Starvation
Starvation is a situation that packets in the any queue of the network are not dequeued for a finite period of time.

Definition 3.2 : Deadlock
Deadlock is a situation that there is a circular blocking in a group of more than or equal to two nodes so that no packets can be loaded onto outgoing links from one node to the other nodes among this group.

Once a call is set up, HP packets are transmitted without screening. The call will be terminated only when all the packets are sent out. For the scout packets, although they have to pass the screening controls, they are only stored as values in scout counter. The scout count will be reset if congestion occurs at the outgoing link. Therefore, deadlock-free for LP packet switching [1] has to be guaranteed.

For the problem of Deadlock, [GUN 81] states that partitioning of queues of a node into at least two parts implies that circular blocking of only two nodes will not occur. Since the structure of queues in the network is partitioned by the routes of packets, this kind of *direct deadlock* or *head-on collision* can be avoided. Deadlock here means *indirect deadlock* due to LP packets switching, i.e. more than three nodes are involved in the circular blocking by LP packets.

Theorem 3.1 : The transition time of changing from congested to non-congested link status is bounded by a finite value $\triangle t$.

Proof : If link congestion occurs, incoming packets from both the environment and remote links are stopped forwarding by control vector screening and load vector screening respectively. Moreover, the difference of influx vector screening and load vector screening shows that queues are flowing or not is independent of the status of its local links. Therefore, at least one queue of the congested link (i.e. queue for single-hop route) is flowing and packets detained in the flowing queues can forward to other nodes. Since the number of detained packets in the flowing queues is finite, within a finite transition time $\triangle t$, no flowing queues will eventually have size greater than N_{LT}. Then the congested link status is changed to non-congested [2].

Remark. Calculation of the transition time $\triangle t$,

[1]LP packet switching can be regarded as Store-and-Forward [TAN 88].

[2]Refer to conditions 1 and 5 in Section 3.4

Let $q_{ij}(t)$ be the no. of packets queued in Q_{ij} of the congested link at time t,

b_r be the mean residue bandwidth of the congested link,

b_j be the mean influx rate of Q_{ij},

N_q be the no. of flowing LP queues related to the congested link,

$A_j(t, t+\delta)$ be the no. of arrival to queue Q_{ij} during the time

t to $t+\delta$.

$$\triangle t = \frac{\sum_j^{N_q} Min(q_{ij}(t) + A_j(t, t+\delta) - N_{LT},\ 0)}{b_r} \tag{3.8}$$

$$\text{where} \quad A_j(t, t+\delta) \ \le \ \sum_j^{N_q} \delta \cdot b_j$$

for all j, $j \neq i$ and queue Q_{ij} of the congested link is flowing.

Theorem 3.2 : The network is Starvation-free.

Proof : Since $\triangle t$ is bounded by a finite value, any queue of a switching node will not be blocked for an infinite period of time. Therefore, together with the fact that the load controller scans all the queues cyclically, then queues are not suffer from Starvation.

Theorem 3.3 : The network is Deadlock-free.

Proof : Except during the finite $\triangle t$ time, it cannot happen that any time slot on the outgoing link is left unoccupied while some packets are being queued.

Example One : Figure 3.8 is a common situation that may lead to indirect deadlock. There is a cycle of four nodes A, B, C, and D, they are sending LP packets to nodes B', C', D' and A' respectively so that the LP queues $Q_{AB'}$ (of route A $\rightarrow$ B $\rightarrow$ B') in node A, $Q_{BC'}$ (of route B $\rightarrow$ C $\rightarrow$ C') in node B, $Q_{CD'}$ (of route C $\rightarrow$ D $\rightarrow$ D') in node C and $Q_{DA'}$ (of route D $\rightarrow$ A $\rightarrow$ A') in node D exceed the upper threshold level N_{UT}. Then, links a, b, c and d are congested. However, whether these LP queues are flowing or not are independent of the link status of links a, b, c and d (see the difference of influx vector screening and load vector screening). Therefore, Deadlock will not occur.

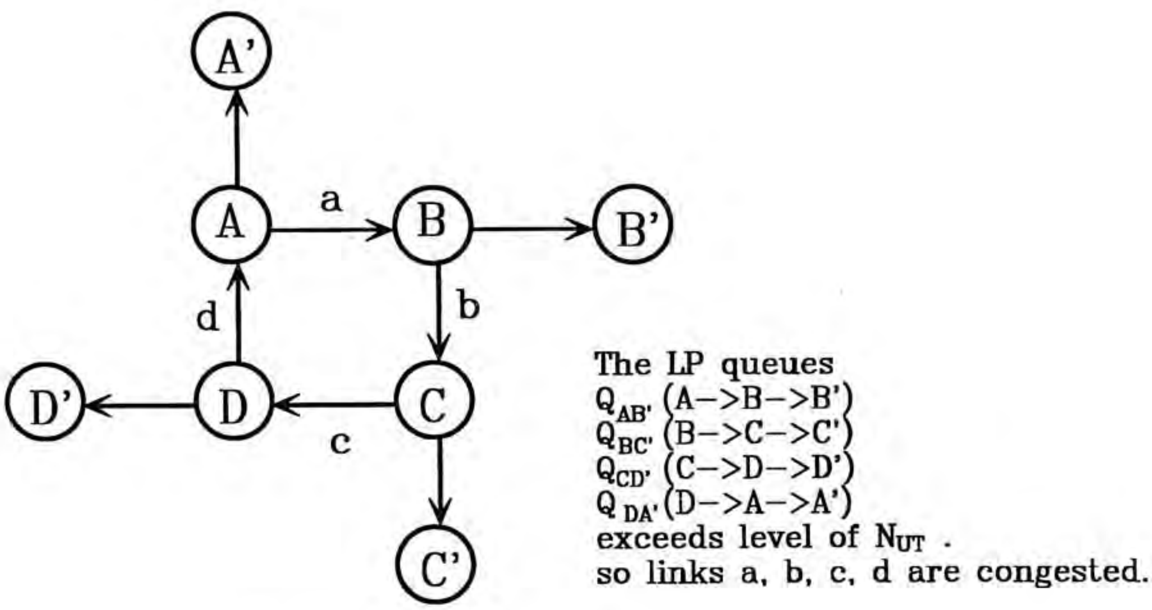


Figure 3.8 Example One

Example Two : Figure 3.9 shows another situation of potential deadlock. Suppose a cycle of four nodes A, B, C and D are sending LP packets to nodes C, D, A and B respectively. The links a, b, c and d are congested due to the LP queues Q_{AC} (of route A $\rightarrow$ B $\rightarrow$ C) in node A, Q_{BD} (of route B $\rightarrow$ C $\rightarrow$ D) in node B, Q_{CA} (of route C $\rightarrow$ D $\rightarrow$ A) in node C and Q_{DB} (of route D $\rightarrow$ A $\rightarrow$ B) in node D exceed N_{UT}. Packets in these LP queues will be blocked instantaneously but after a Δt period, one of the links, say link a, will changed to non-congested by Theorem 3.1. Thus, the queue Q_{DB} in node D becomes flowing which will make link d change to non-congested. If link d is non-congested, the queue Q_{CA} in node C can be flowing. Eventually, other congested links will become non-congested.

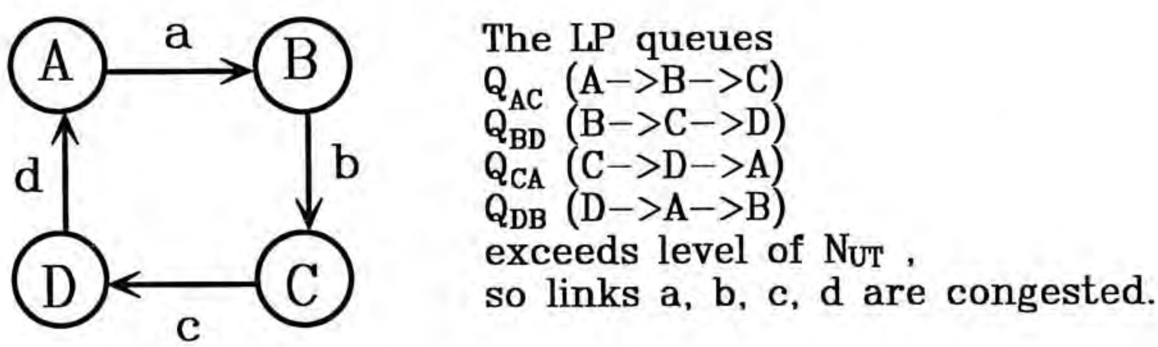


Figure 3.9 Example Two

Chapter 4

Call Set-up Algorithm & Traffic Modelling

4.1 Basic Algorithm

For connection oriented communications, it is up to the call set-up process to control the traffic volume of periodic packet streams generated by calls. A call set-up algorithm is designed in conjunction with the traffic flow control algorithm. Upon the request of call set up, the call originating node transmits the stream of scout packets. If the periodic stream of scout packets enters the network smoothly for a specified length of time, then it is followed by the stream of HP packets. In that case, a forward emulated circuit for data transmission is set up. The scout packet stream consists of two parts (Figure 4.1), first part is a stream of N_{DS} *detecting scout* (DS) packets which is used to test the bandwidth availability of all the links along the route of the intended call. Two schemes are proposed in Sections 5.1 and 5.2 to manage how DS packets are loaded into the transmission link. The second part is a stream of N_{RS} *reserving scout* (RS) packets. It is to reserve the bandwidth of the tested links. RS packets are generated with the same bit rate as the HP packets.

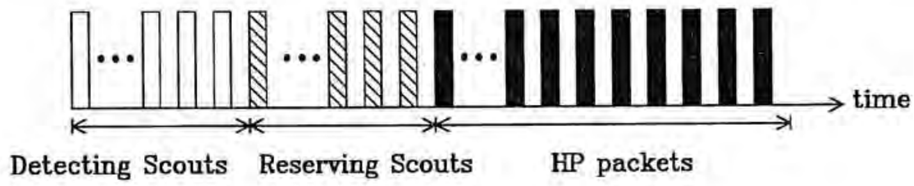


Figure 4.1 Scout streams of a HP call

If a scout packet is denied entry to the network, the set-up attempt is aborted. It is then up to the nodal function of route selection to decide whether to shift to an alternative route upon reattempt. The pause time before a reattempt may be designed according to any post-collision strategy in medium access protocols for multi-user communication channels. Some of such protocols can be found in [KLE 76]. Thus the call set up decision is purely made by the call originating node. The whole process is transparent to other nodes on the route of the intended call. Figure 4.2 describes the basic call set-up algorithm by a flowchart. Two simple and efficient call set up schemes are proposed to investigate how scout packets are managed to set up a call. Estimation and fine tuning of parameters of these two schemes are studied in Chapter 5.

4.2 Minimization of Bandwidth Overhead

In a queue fed by an output of the switch mentioned in Section 3.3, scout packets take precedence over LP packets by virtue of their slightly higher priority class. When the outgoing link that the queue leads to is congested, all scout packets are annihilated. The reasoning behind is that, when scout packets have achieved

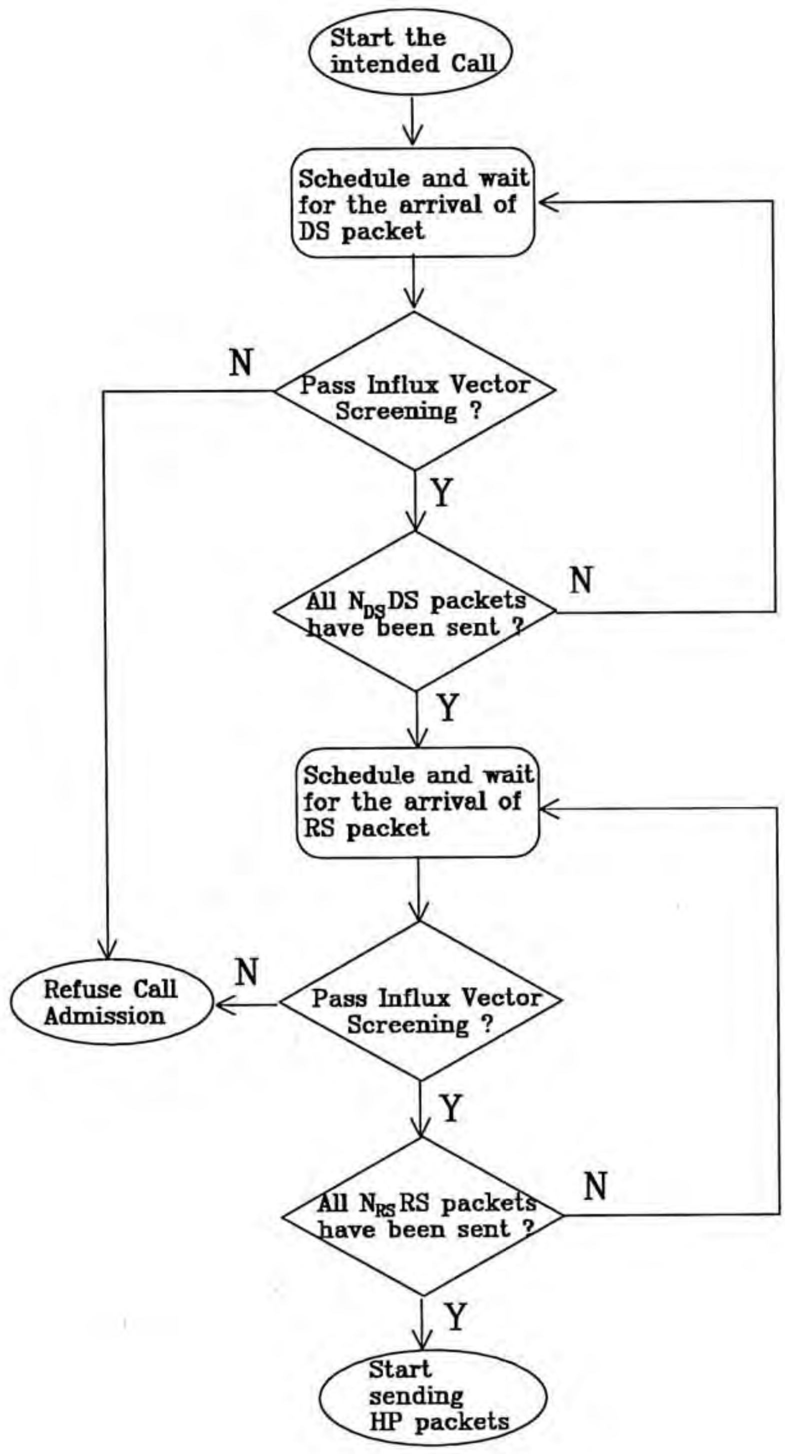


Figure 4.2 Basic Algorithm for Call Set−up

the mission of detecting link congestion, they serve no further purpose and hence there is no need of further investing network resources on them. Thus the number of scout packets in any queue is at most N_T. The annihilation of scout packets as soon as possible maximizes the part of link transmission capacity that is left for LP packets to fill in.

Moreover, it is possible to superimpose a scout packet over an LP packet of the same route during transmission in the link. Each superimposed packet has a label to identify whether the scout packet is a DS or RS packet. On the receiving side of the packet, a superimposed packet will be split into an LP packet and a scout packet according to its label (Figure 4.3). This mechanism can also minimize the bandwidth overhead of scout packets.

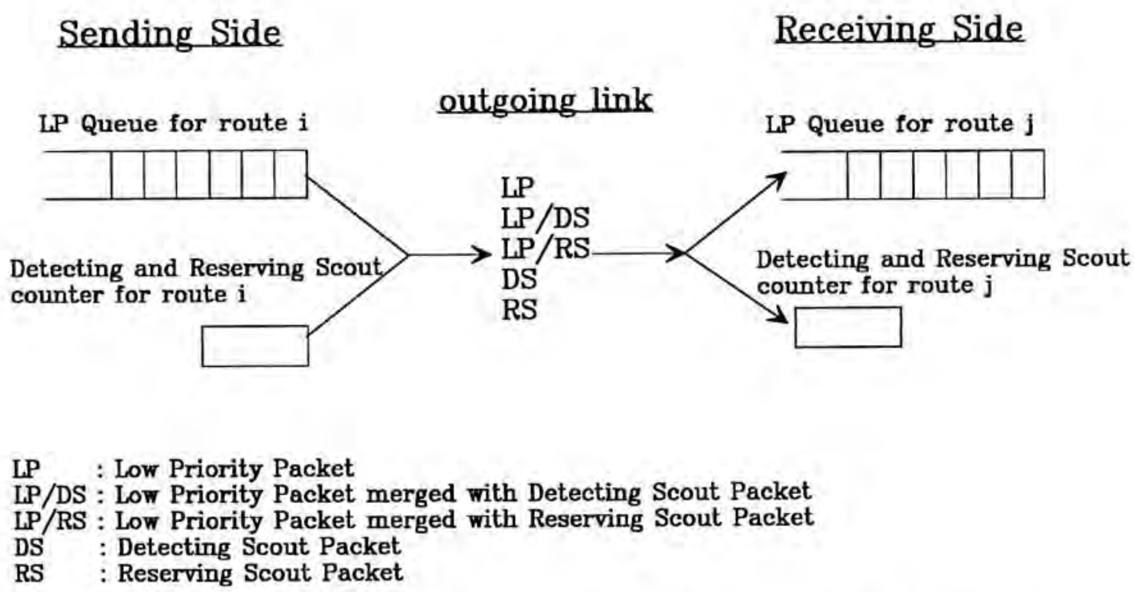


Figure 4.3 Superimposing LP packet over scout

4.3 Two-way Transmission

So far, the algorithm has only dealt with the set-up of one-way transmission with the assumption that the receiving end is always ready to receive. A two-way call would require a hand-shake process between calling and called ends before the call can flow in both directions. A straightforward extension of the algorithm to two-way calls is as follows.

First, the call originating node clears a forward emulated circuits with scout packets as before. A request for response follows. The transmission of data packets can not start until the call receiving end responds. Meanwhile the request for response is followed by a stream of time-filling packets. The function of time-fillers is to preserve the forward emulated circuit until the transmission of data packets can start. This function is equivalent to time-slot reservation in the call set-up process in the circuit-switching environment. The priority class and periodicity of time-fillers are the same as of data packets of the intended call. Like the scout packets, time-fillers have null content in their payload field. They can be treated as pseudo packets in the same way as scout packets in order to save switching resources except that their priority class is higher.

Upon receiving the request for response, the call terminating node tries to reserve a backward emulated circuit and rings the called party. When the backward emulated circuit is set up by scout packets and the called party has answered, a response signal is sent to the call originating node. The stream of data packet can then follow.

4.4 Traffic Modelling

Before the estimation and tuning of parameters of the aforementioned algorithm of call set up, knowledge of traffic models of broadband integrated services have to be acquired. This section studies the properties of three traffic models. The first traffic model is the integration of periodic sources of different bit rates. The second is the integration of on-off sources of different bit rates. The third is the mix of these two.

In Broadband ISDN network, sources like data, voice calls and video calls etc. are transmitted in a single channel in which packets are statistically multiplexed from different calls. It is still uncertain about the actual traffic mix in real life because development of broadband ISDN network remains in infancy, though experimental work on broadband network is progressing such as the RACE project (BLNT), MAGNET II [LAZ 90] and the Fairisle [LES 91].

A definition of Quality of Services (QOS) is given in CCITT Recommendation I.350 according to which 'Quality of Service is defined as the collective effect of service performances which determined the degree of satisfaction of a user of the specific service'. QOS is usually measured as the averaged packet delay time and packet loss probability. To satisfy different QOS of packet transmission in integrated services environment, packets are grouped into two priority classes. The use of priorities also allow us to achieve a greater efficiency.

Data traffic for computer network is insensitive to delay time but have a straight requirement on packet loss. Therefore, packets generated from sources like data file transfer and retrieval are regarded as LP packets. The arrival process of LP packets can be modelled by Poisson Process [FUC 70]. On the other hand, the HP packets, like voice and video packets, can withstand packet loss but very sensitive to transfer delay time. The above descriptions of HP and LP packets can be summarized by Table 4.1.

Table 4.1

Packet Type	QOS	Arrival Models	Traffic Control	Examples
Low Priority	Low cell loss probability	Poisson	Packet-level Control	Electronic Mail, Non-critical File Transfer
High Priority	Low transfer delay	Periodic / On-off Sources	Call Set up Algorithm	Video, Voice and Critical File Transfer

4.4.1 Aggregate Traffic Models

At the time of loading packet onto the outgoing link, other packets have to pass over in favour of HP packets. As a result, LP packets have no effect on the aggregate traffic models of HP packets. The aggregate traffic pattern is modelled solely by superposition of HP packets. Individual HP packet call is modelled by two types of traffic sources, namely periodic source (Figure 4.4) and on-off source (Figure 4.5). The aggregate traffic patterns generated by periodic sources and on-off sources are periodic traffic mix and on-off traffic mix respectively. The process of superposition is done by an ideal statistical multiplexer which consists of an adder and a single buffer queue as shown in Figure 4.6. The buffer queue is to detain the multiple packets at coincident packet arrival epochs. These traffic

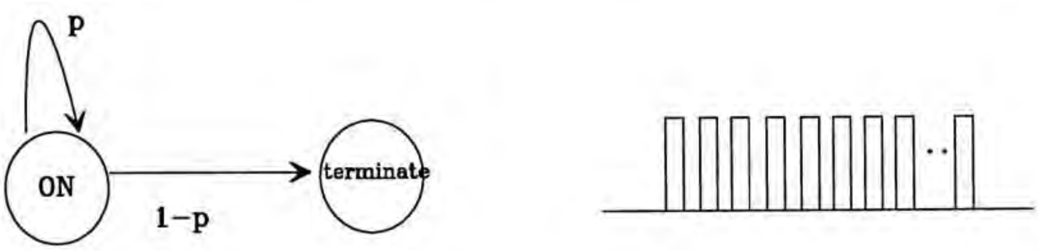


Figure 4.4 A Periodic Source

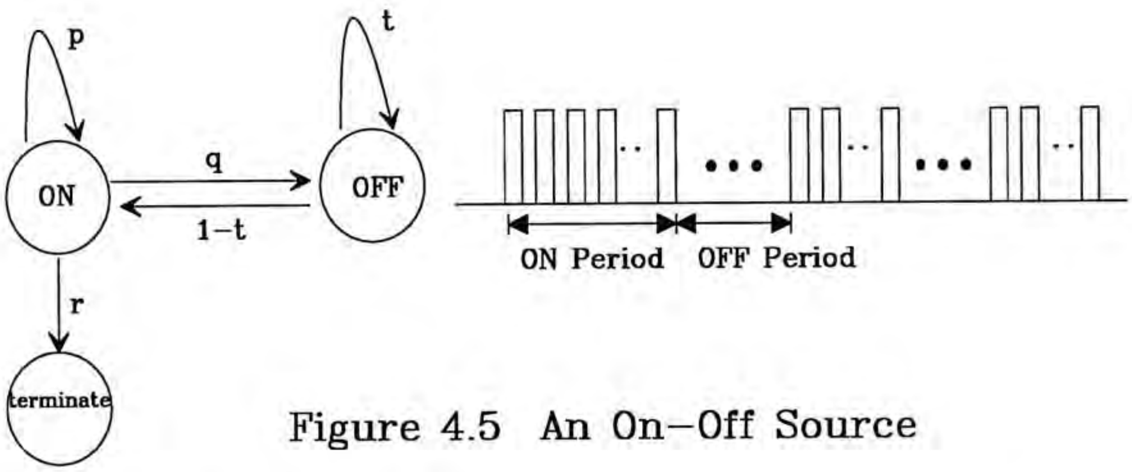


Figure 4.5 An On–Off Source

sources are in Constant Bit Rate (CBR).

The aggregate traffic pattern of a link in a network with multiple hop calls is the result of multiplexing traffic from both the influx and the other nodes. Packet multiplexing takes place at either channel multiplexers or switches. The traffic sources are superimposed by cascade multiplexing, see Figure 4.7. If statistical multiplexing is applied and total loading of traffic sources is less than the link bandwidth, the effect of cascade multiplexing can be approximated by single stage multiplexing.

Periodic traffic mix (PTM) is modelled by superposition of a number of calls

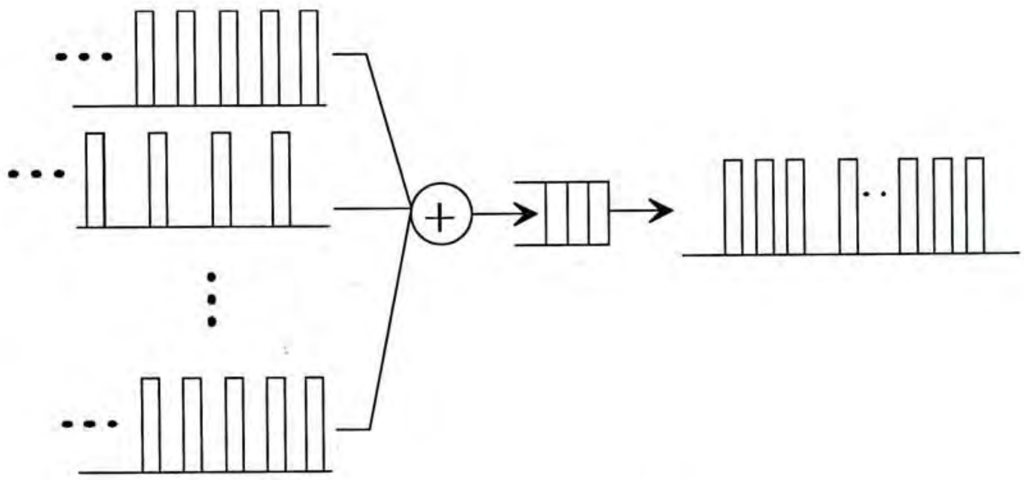


Figure 4.6 Superposition of traffic Sources

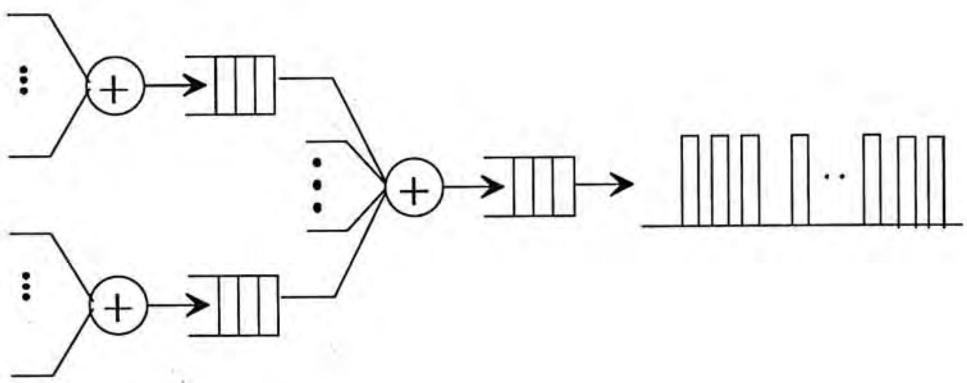


Figure 4.7 Cascade Multiplexing

which generate packets continuously in different bit rate. Each call terminates after generating a geometrically distributed number of HP packets. Moreover, the interarrival time of calls are also geometrically distributed. The mean interarrival time of a call is usually very long compared with the packet loading time in broadband network. The mean call duration is in the order of minutes but the packet loading time is only in the order of sub-microseconds.

The traffic mix is described by the following notations,

Vector of packet interarrival time T_i of call type i,$(i = 1, \ldots, t)$: $[T_1, T_2, \ldots, T_t]$

Vector of mean number of sources N_i for the corresponding type : $[N_1, N_2, \ldots, N_t]$

Mean bit rate of the traffic mix :

$$B_{mean} = \sum_{i=1}^{t}(N_i \cdot \frac{1}{T_i}) \tag{4.1}$$

On-off sources traffic mix (OFTM) is modelled by superposition of a number of on-off sources. Each on-off source consists of alternating sequence of on-periods and off-periods. During on-period, packets are generated in CBR. Both the length of on- and off-period is geometrically distributed. Like PTM, the interarrival time of the calls are geometrically distributed and it can be represented by a similar notations, except that an extra vector is used to represent the mean on-period $P_{on,i}$ and off-period $P_{off,i}$ of type i call.

$$[P_{on,1}, P_{on,2}, \ldots, P_{on,t}], [P_{off,1}, P_{off,2}, \ldots, P_{off,t}]$$

Mean bit rate of the traffic mix :

$$B_{mean} = \sum_{i=1}^{t} \left[\frac{P_{on,i}}{P_{on,i} + P_{off,i}} (N_i \cdot \frac{1}{T_i}) \right] \tag{4.2}$$

Periodic and on-off traffic mix (POM) is the mix of the PTM and OFTM. The mean bit rate of POM is simply the sum of the mean bit rate of these two components.

4.4.2 Traffic Burstiness

Besides the peak bit rate and average bit rate, burstiness of packets interarrival time is the most important traffic parameter in determining performance of ATM network. Basically, burstiness formulation can be generalized into three methods. The first method [DIT 88] [KUL 84] [GAL 89] is to get the ratio of peak bit rate to mean transmission bit rate of a source averaged over the duration of a call. The second method [HI2 89] is to calculate the average burst length measurement. Average burst length is the mean duration of time interval during which the traffic source transmits at the peak rate. The final method [HUI 87] [SRI 86] is to determine the *index of dispersion* for counts $I(t)$ and this method is adopted as the measurement of Burstiness throughout this thesis.

$$I(t) = \frac{Var[N(t)]}{E[N(t)]}, \quad t > 0. \tag{4.3}$$

where $N(t)$ is counting process associated with the arrival process of packets.

$Var[N(t)]$ and $E[N(t)]$ are variance and expected value of cumulative $N(t)$ of packets arriving in $[0, t]$ respectively.

For a Poisson Process, $I(t) = 1$, and for a constant bit rate process $I(t) = 0$, for all $t > 0$.

One unit time is the time for loading one packet onto the link. A time slot is known as active, if a HP packet is loaded onto a link at that time slot , otherwise it is called the *idle time slot.* Since HP packets have priority over other kinds of packets and scout packets have priority over (or equal priority as) LP packets, scout packets can only be loaded at the idle time slots. Therefore, the idle time slot interarrival time of a link under a traffic mix of HP packets relates closely to the loading rate of scout packets which in fact affects the performance of the call set up algorithm. The distribution of idle time slot interarrival time at short time interval is examined in order to understand know how the scout packets can get through the links along its path of the connection at the time of call set-up.

Graphs 4.1-8 show the index of dispersion of idle time slot interarrival time under different loading of PTM, OFTM and POM as shown in Tables 4.2-6. It is found that the burstiness of idle time slot interarrival time of PTM and OFTM are close to that of Poisson Process at very short time interval (Graphs 4.1-2). Due to the periodic nature of different periodic sources of PTM, it is obvious that as the time interval increases, the Index of Dispersion of PTM tends to be zero value. The curves also dip at the periodic moments of each type of traffic sources.

For the OFTM, Graphs 4.3-4 show that the burstiness of idle time slot increased slowly as the time interval is longer. The burstiness also increases as the traffic loading increases. Graphs 4.5-6 portray the effect of the call to link bandwidth ratio on the burstiness of the idle time slot interarrival time. For the same traffic loading, if the ratio is small, more traffic sources can be accommodated. In other words, if the call bandwidth of sources does not change, the traffic burstiness decreases as the link bandwidth increases. Graphs 4.7-8 depict the burstiness under periodic and on-off traffic mix (POM).

It is found that the traffic burstiness of models which consist of on-off traffic sources relates closely to the following parameters :

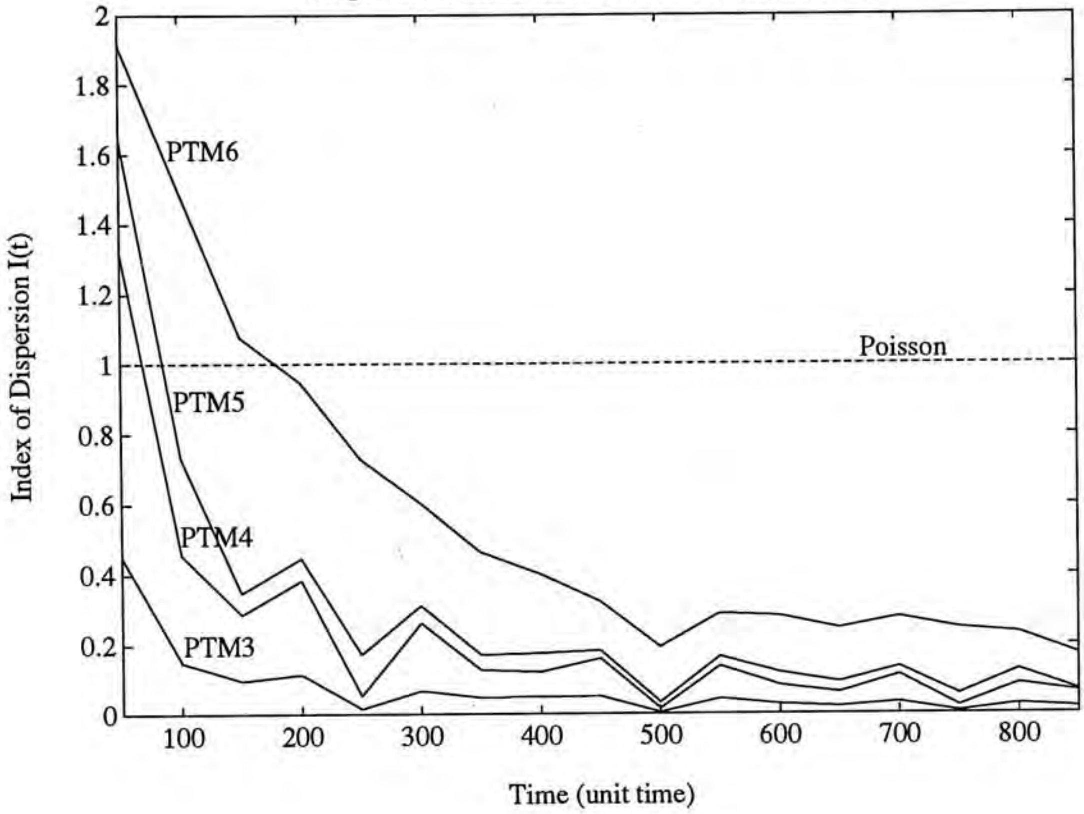
1. Mean values of on and off period of various types of calls.

2. Ratio of mean on and off period of various types of calls.

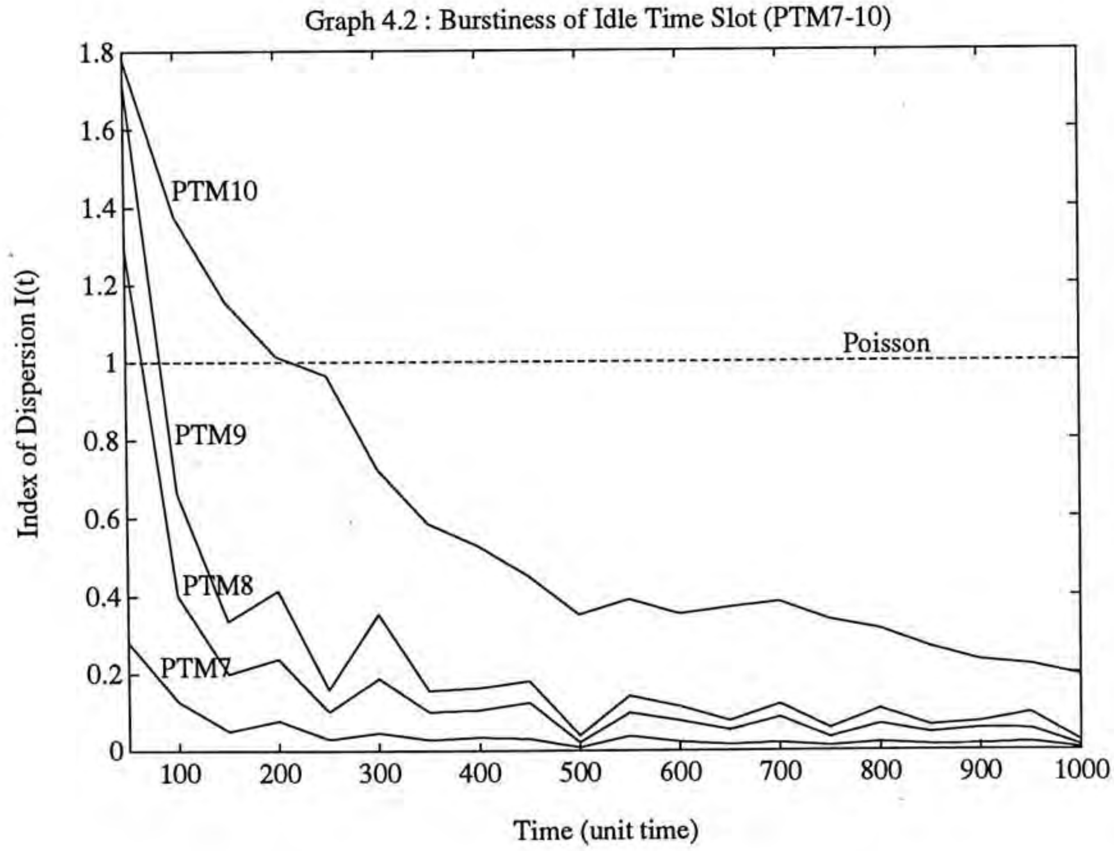
3. Total traffic loading, (i.e. the total number of calls).

4. Call to Link bandwidth ratio.

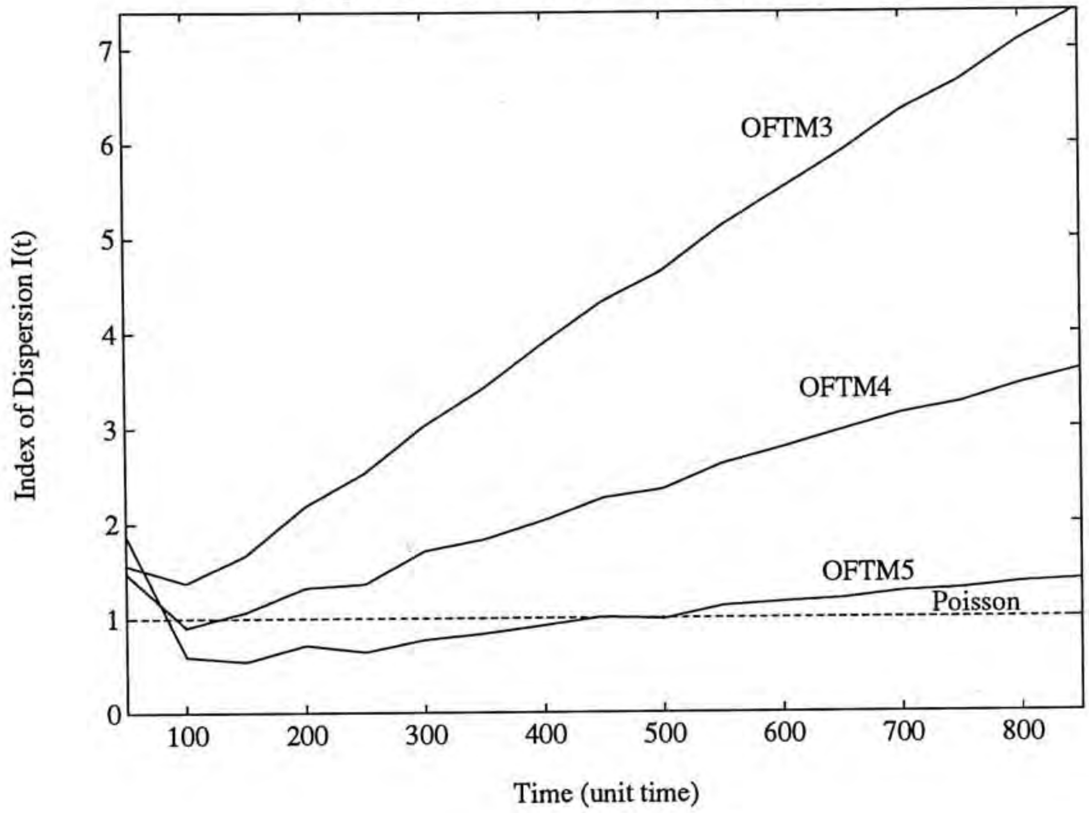
Graph 4.1 : Burstiness of Idle Time Slot (PTM3-6)



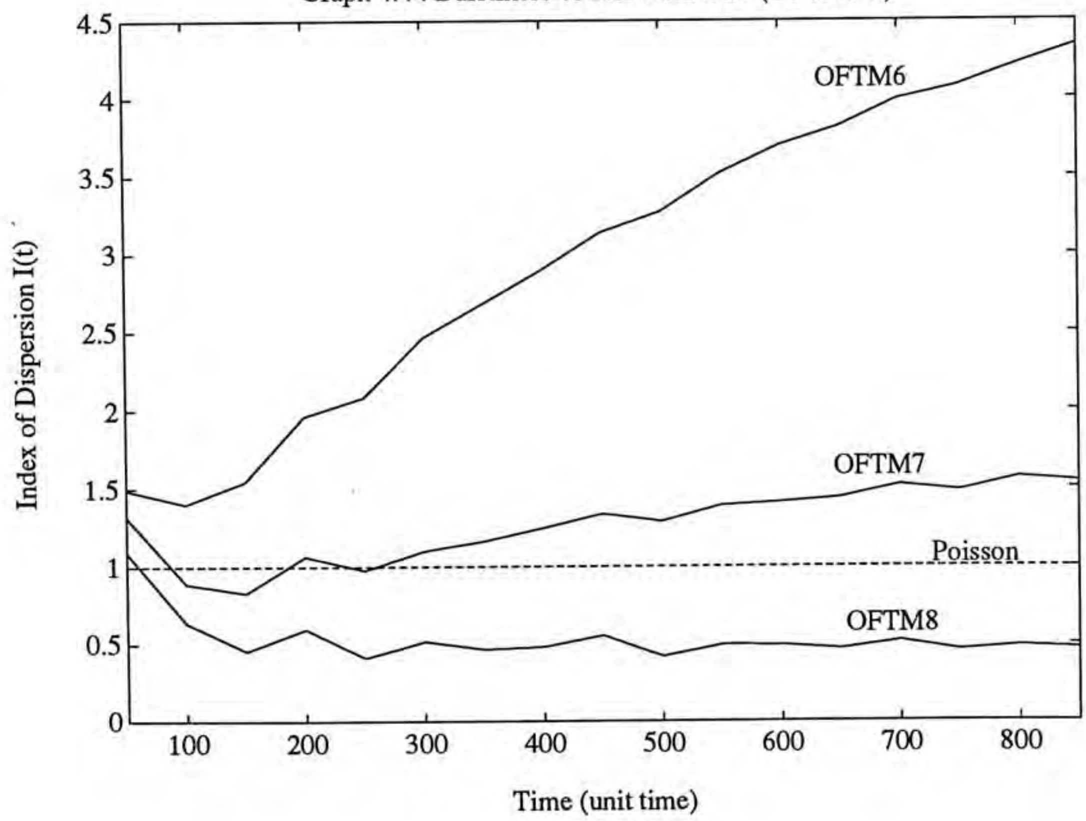
Graph 4.2 : Burstiness of Idle Time Slot (PTM7-10)

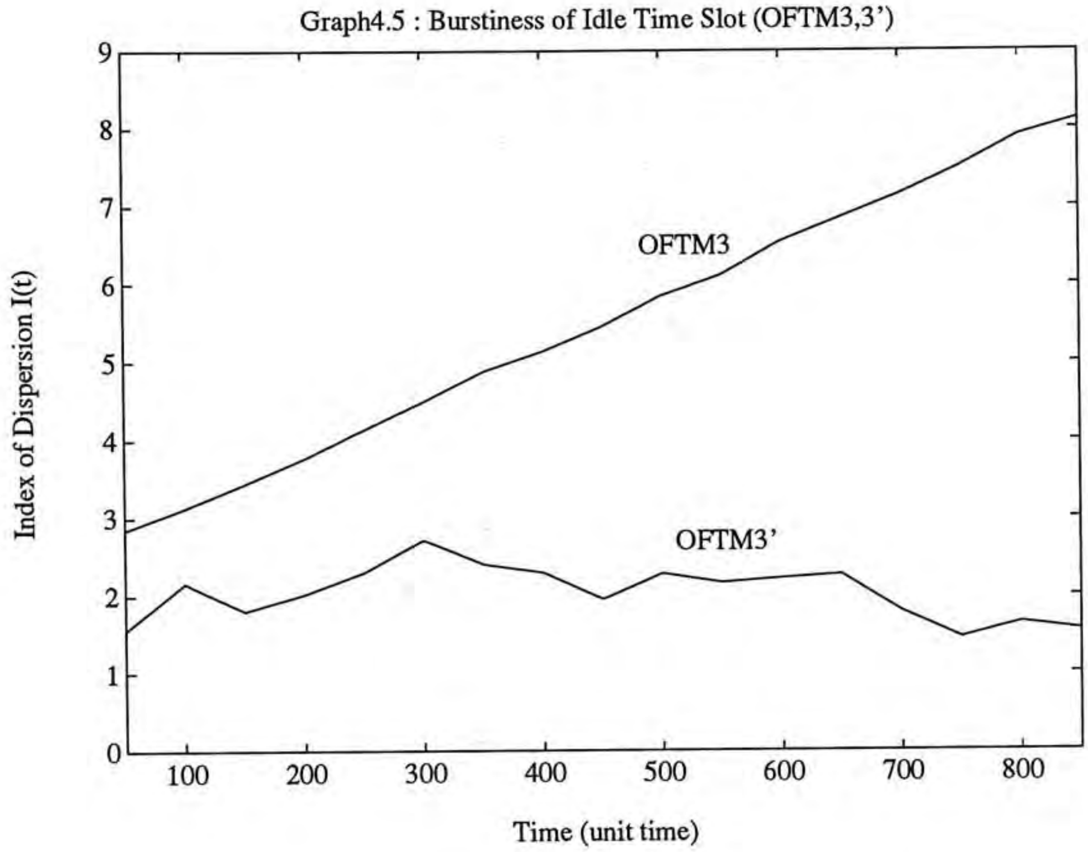


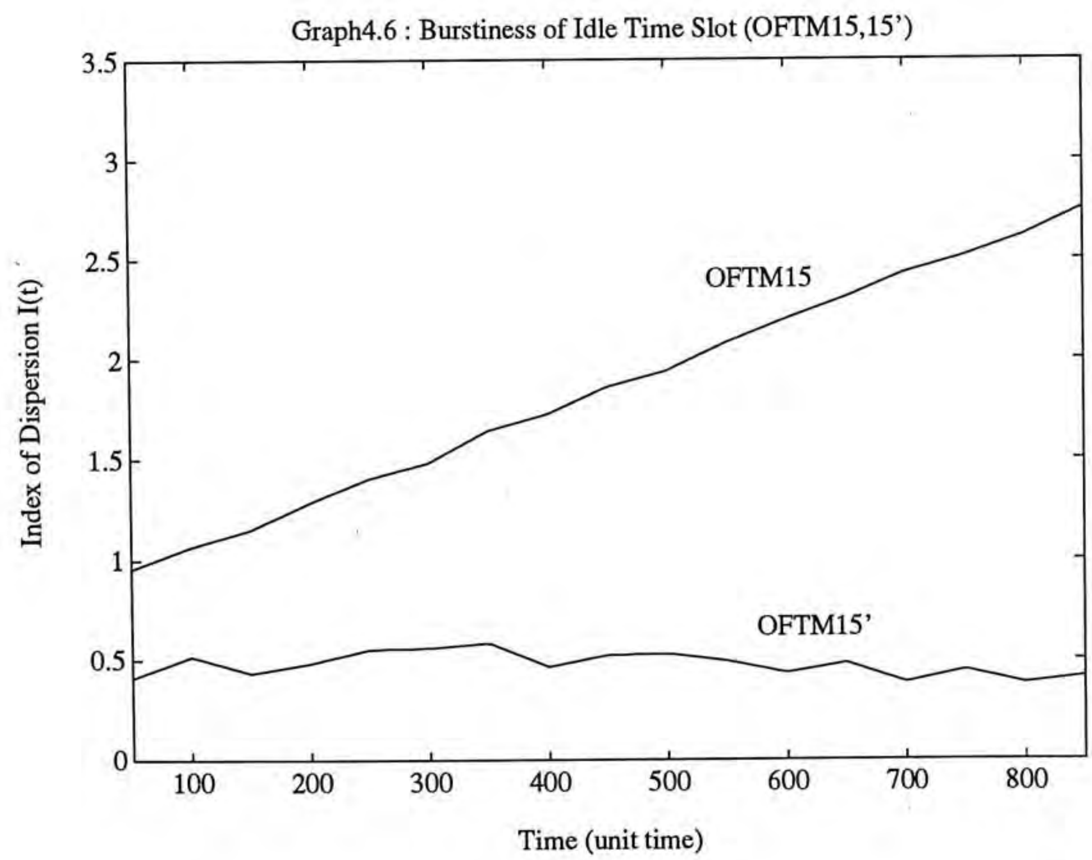
Graph 4.3 : Burstiness of Idle Time Slot (OFTM3-5)



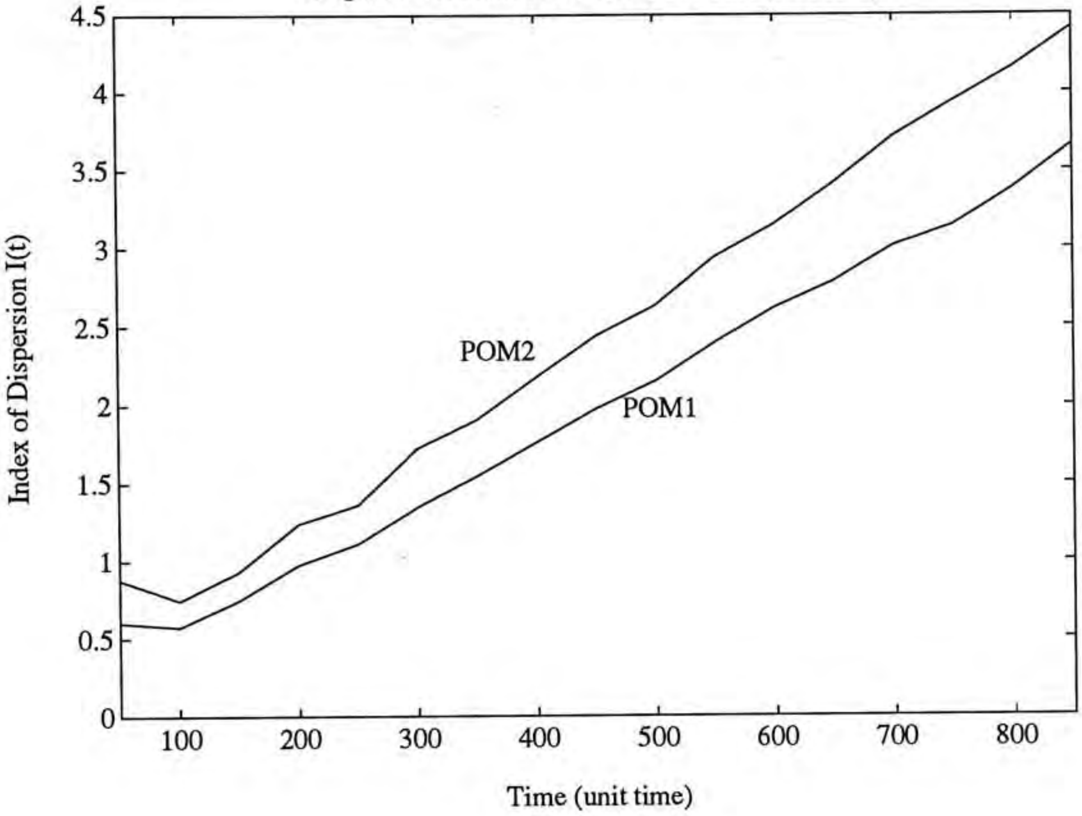
Graph 4.4 : Burstiness of Idle Time Slot (OFTM6-8)



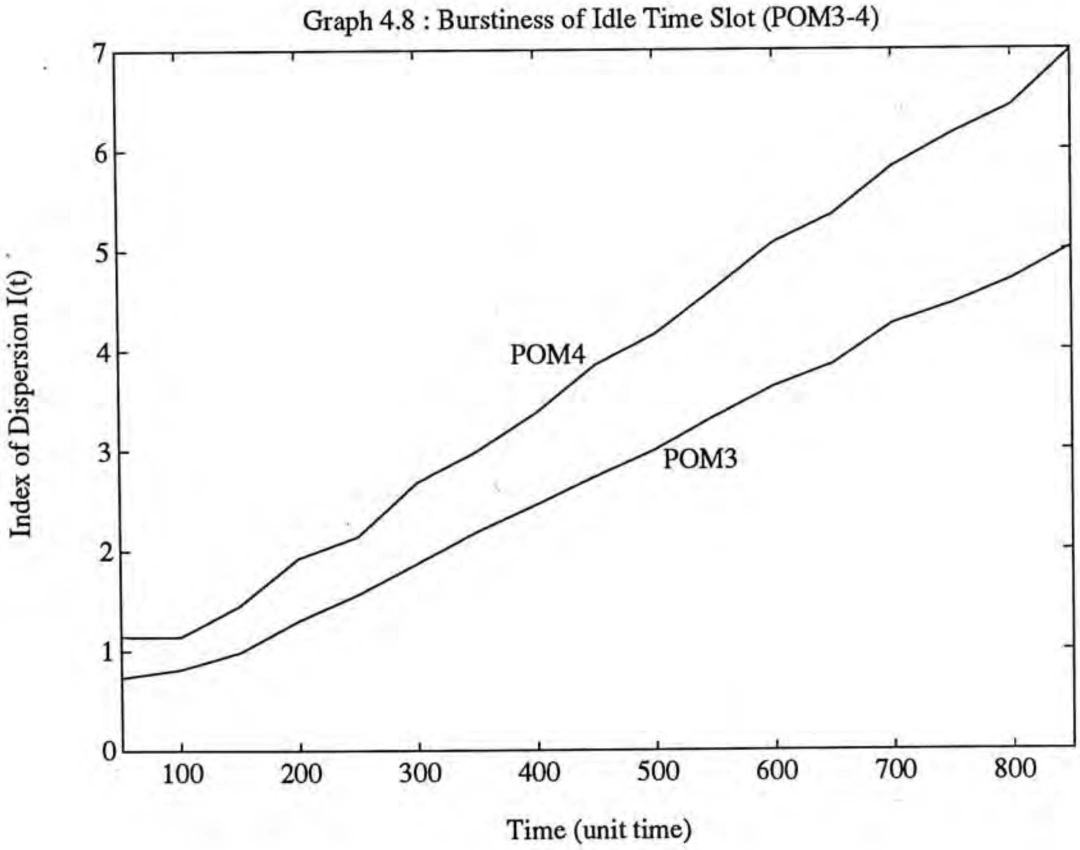

Graph4.5 : Burstiness of Idle Time Slot (OFTM3,3')
Index of Dispersion I(t)
Time (unit time)
OFTM3
OFTM3'
0
1
2
3
4
5
6
7
8
9
100
200
300
400
500
600
700
800


Graph4.6 : Burstiness of Idle Time Slot (OFTM15,15')
Index of Dispersion I(t)
Time (unit time)
OFTM15
OFTM15'
0
0.5
1
1.5
2
2.5
3
3.5
100
200
300
400
500
600
700
800

Graph 4.7 : Burstiness of Idle Time Slot (POM1-2)



Graph 4.8 : Burstiness of Idle Time Slot (POM3-4)



Throughout the thesis, all the On-off traffic sources are chosen in a way that $P_{on,1} = P_{on,2} = P_{on,3} = P_{on,4} = P_{on}$ and $P_{off,1} = P_{off,2} = P_{off,3} = P_{off,4} = P_{off}$.

Table 4.2

$[T_1, T_2, T_3, T_4] = [125, 72, 51, 37]$

$[N_1, N_2, N_3, N_4]$ of PTM :

B_{mean}	Type 1 Call Dominating	Even Traffic
0.50	$PTM3 = [40, 5, 3, 2]$	$PTM7 = [20, 10, 6, 3]$
0.80	$PTM4 = [60, 10, 5, 3]$	$PTM8 = [40, 15, 7, 5]$
0.90	$PTM5 = [51, 21, 6, 3]$	$PTM9 = [44, 17, 9, 5]$
0.99	$PTM6 = [62, 21, 6, 3]$	$PTM10 = [42, 20, 11, 6]$

Table 4.3

$[T_1, T_2, T_3, T_4] = [125, 72, 51, 37]$

$B_{mean} = 0.8$, $[N_1, N_2, N_3, N_4]$ of OFTM :

OFTM Traffic	P_{on}	P_{off}
$OFTM3 = [50, 18, 8, 6]$	10000	2000
$OFTM4 = [44, 17, 7, 6]$	10000	1000
$OFTM5 = [43, 16, 7, 5]$	10000	500
$OFTM6 = [50, 18, 8, 6]$	2000	400
$OFTM7 = [44, 17, 7, 6]$	2000	200
$OFTM8 = [43, 16, 7, 5]$	2000	100

Table 4.4

$[T_1, T_2, T_3, T_4] = [125, 72, 51, 37]$

B_{mean}	$P_{on} = 10000, P_{off} = 500$	$P_{on} = 2000, P_{off} = 100$
0.50	$OFTM9 = [32, 10, 4, 2]$	$OFTM12 = [32, 10, 4, 2]$
0.80	$OFTM5 = [43, 16, 7, 5]$	$OFTM7 = [43, 16, 7, 5]$
0.90	$OFTM10 = [51, 15, 10, 5]$	$OFTM13 = [51, 15, 10, 5]$
0.99	$OFTM11 = [55, 16, 12, 5]$	$OFTM14 = [55, 16, 12, 5]$

Table 4.5

B_{mean}	$[T_1, T_2, T_3, T_4]$	$P_{on} = 40000, P_{off} = 40000$
0.80	[125, 72, 51, 37]	$OFTM3 = [50, 18, 8, 6]$
0.80	[1250, 720, 510, 370]	$OFTM3' = [500, 180, 80, 60]$
0.40	[125, 72, 51, 37]	$OFTM15 = [25, 9, 4, 3]$
0.40	[1250, 720, 510, 370]	$OFTM15' = [250, 90, 40, 30]$

Table 4.6

$[T_1, T_2, T_3, T_4] = [125, 72, 51, 37]$

B_{mean}	Traffic Mix	Prop. of on-off sources		$P_{on} = 40000, P_{off} = 40000$
0.80	POM1	9%	$PTM1 = [59, 9, 4, 2]$	$OFTM1 = [2, 2, 2, 2]$
0.80	POM2	20%	$PTM1' = [50, 8, 4, 2]$	$OFTM1' = [20, 4, 2, 2]$
0.90	POM3	9%	$PTM2 = [49, 20, 5, 2]$	$OFTM2 = [4, 2, 2, 2]$
0.90	POM4	20%	$PTM2' = [40, 18, 5, 2]$	$OFTM2' = [22, 6, 2, 2]$

Then the arrival process of idle time slot at short time interval under PTM and less bursty OFTM is compared with Poisson Process.

Definition of Poisson Process :

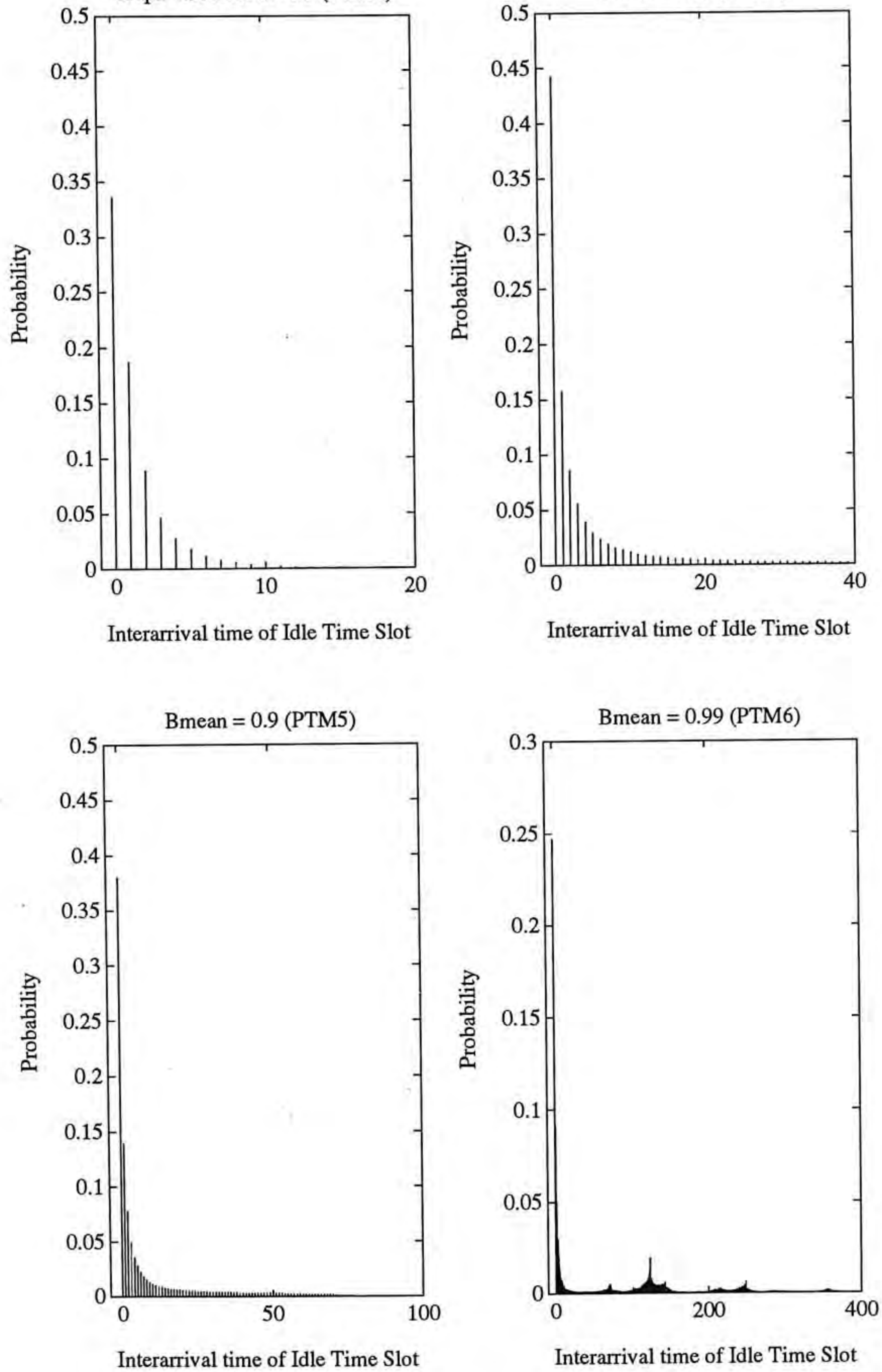
1. $N(0) = 0$.
2. Increments over disjoint time intervals are independent.
   That is, increment over $(t, t+s) = N(t+s) - N(t)$
3. The number of events in any interval of length t is Poisson distributed with mean λt. That is, for all $s, t \geq 0$

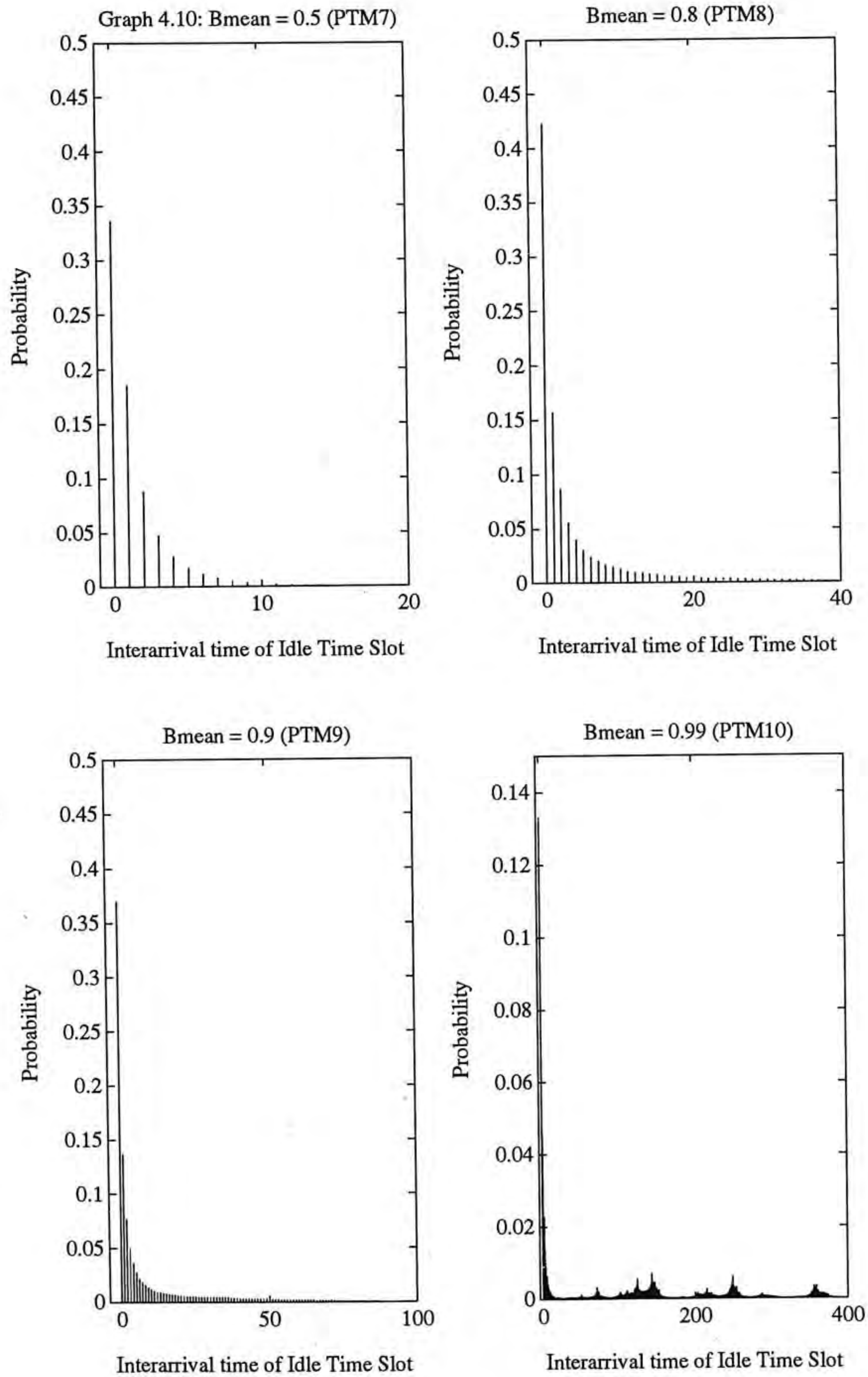
$$P\{N(t+s) - N(s) = n\} = e^{\lambda t}\frac{(\lambda t)^n}{n!}, \quad n = 0, 1, \ldots \tag{4.4}$$

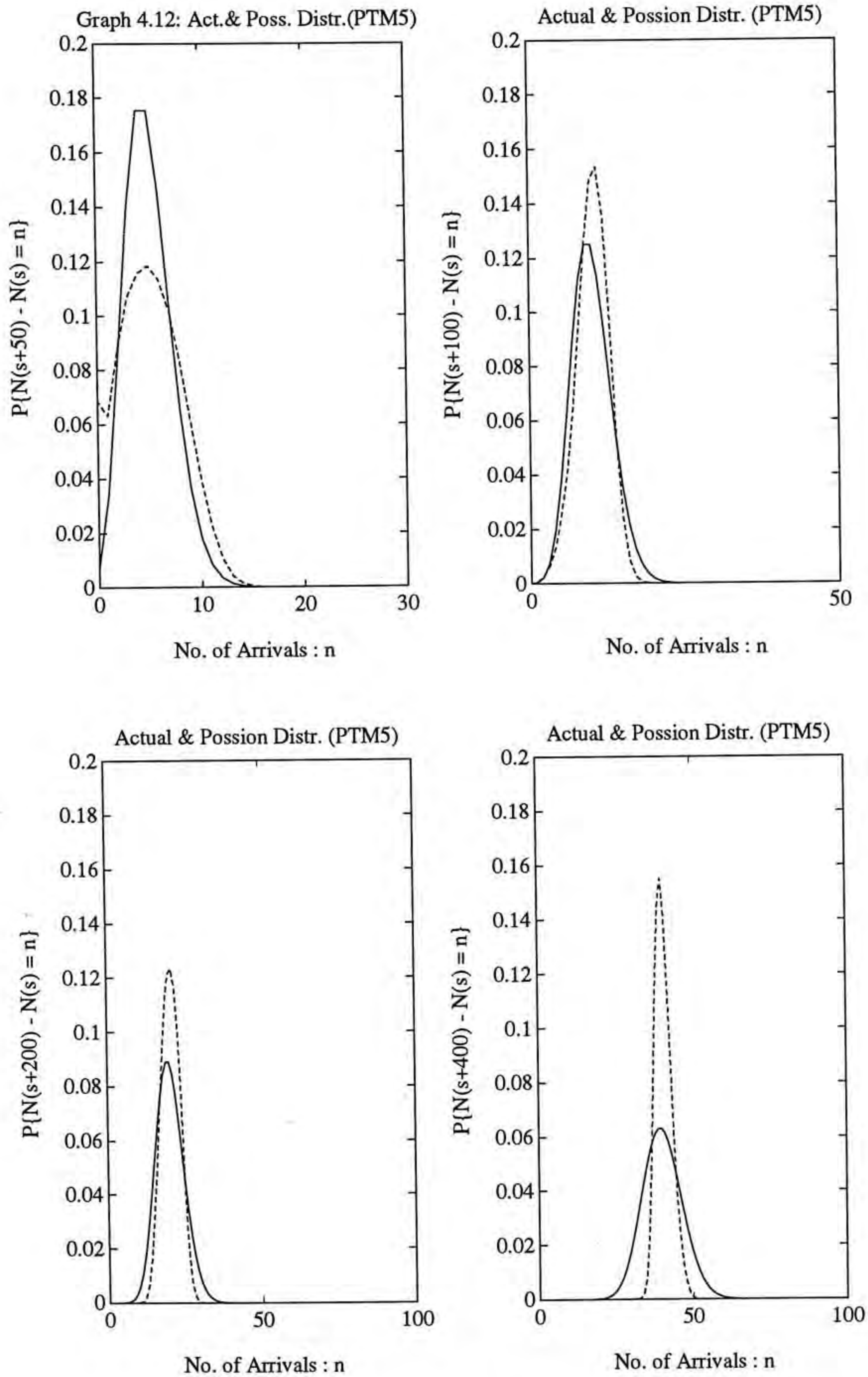
It is obvious that $N(0) = 0$ for no arrival of idle time slot at $t = 0$. The distribution of idle time slot interarrival time can be studied carefully by Graphs 4.9-10 and Graphs 4.13-14 . The periodicity of interarrival time of idle time slot become observable when the traffic loading is extremely high. However, if the time interval is small, the distribution is approximately exponential which implies Def.2. The distribution of the increments (the dotted lines) under different time intervals is also compared with the actual Poisson Distribution (the solid lines) in Graphs 4.11-12 and Graphs 4.15-16 to validate Def.3 at short time interval.

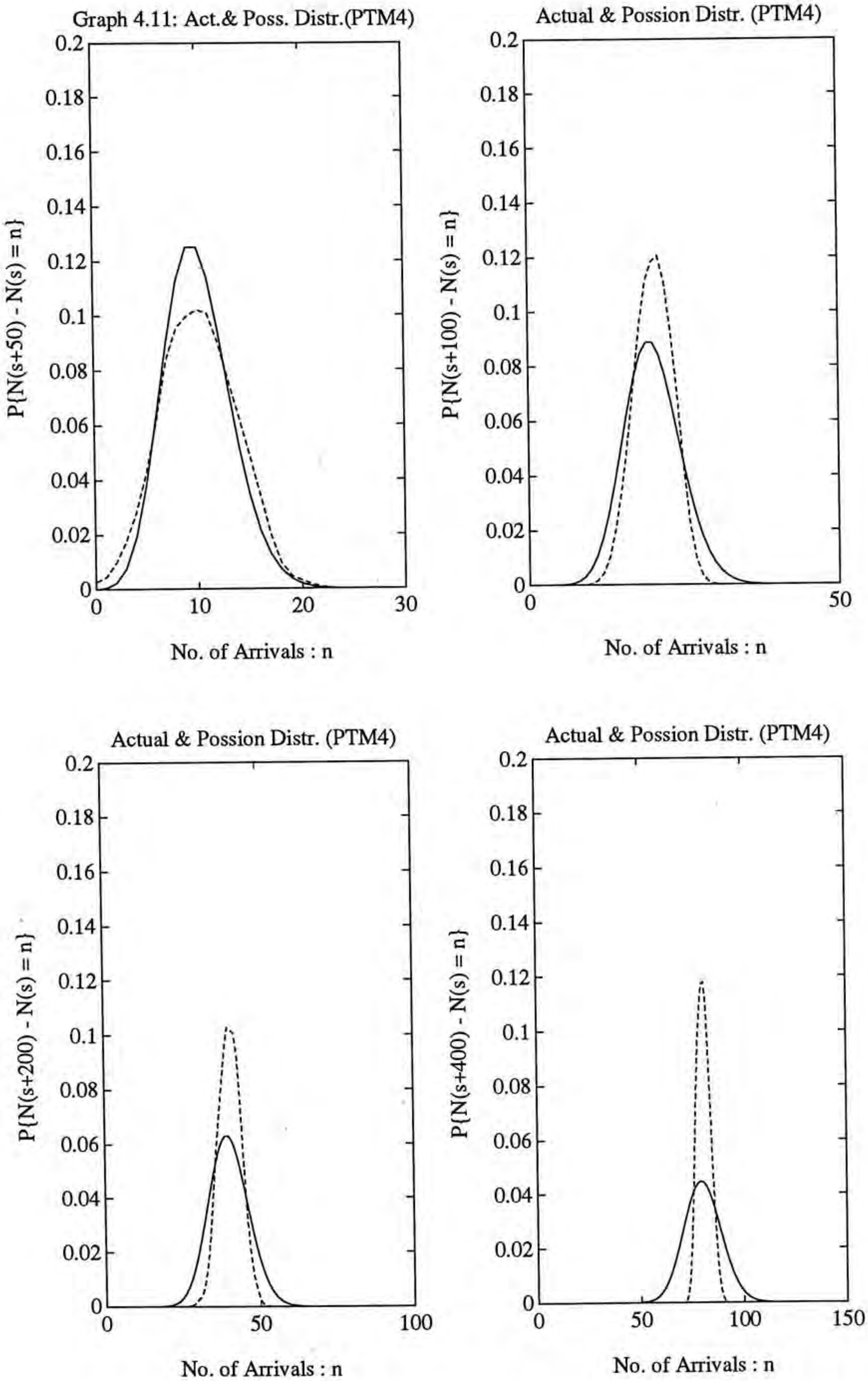
According to the above observations, it is found that the idle time slot interarrival process of PTM and less bursty OFTM at short time interval can be approximated by Poisson Process. Poisson Process may also assume to be the upper bound of burstiness of idle time slot interarrival time of PTM. The same assumption can be applied to OFTM only if the burstiness is low.

Graph 4.9: Bmean=0.5 (PTM3)

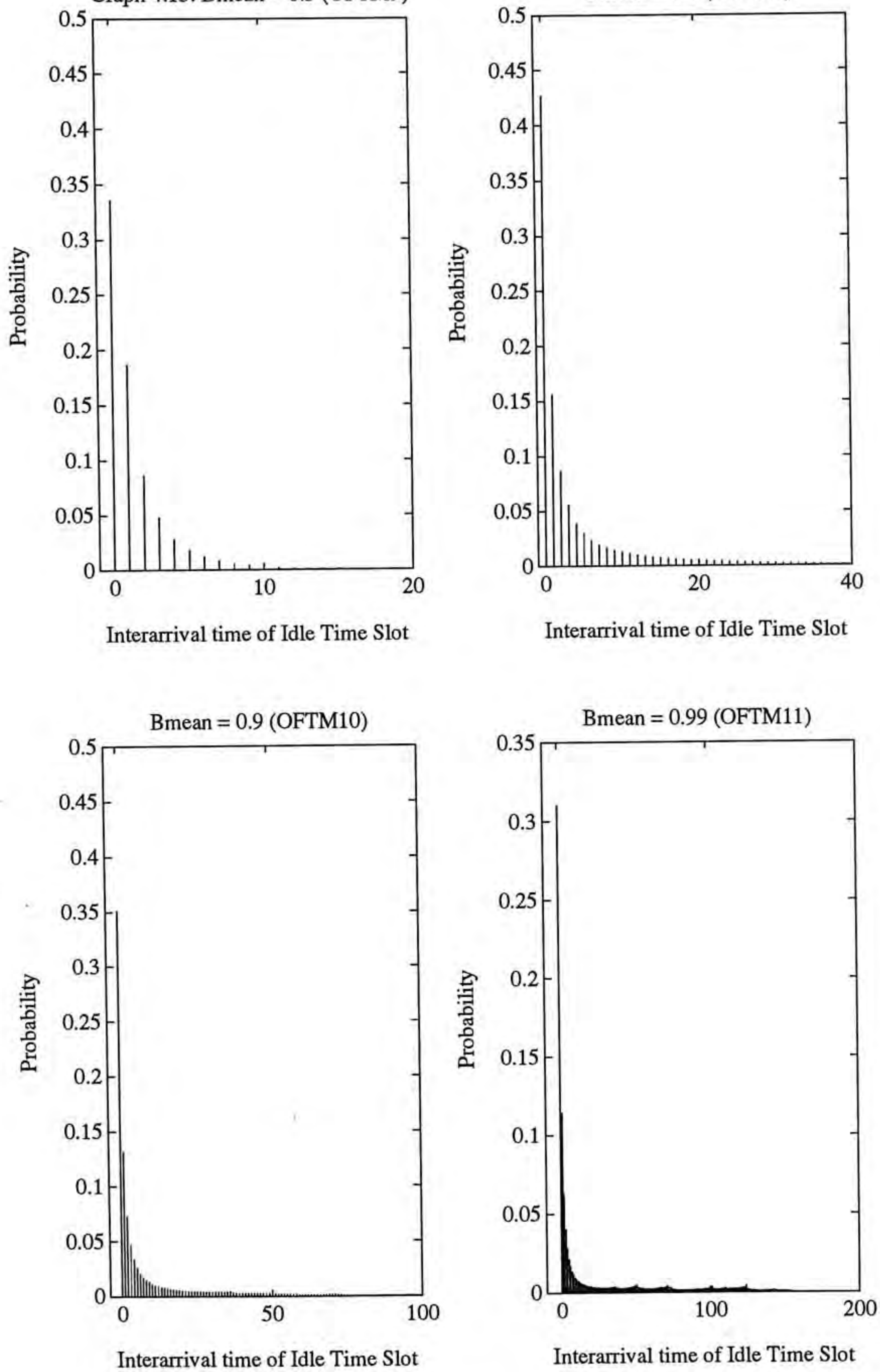


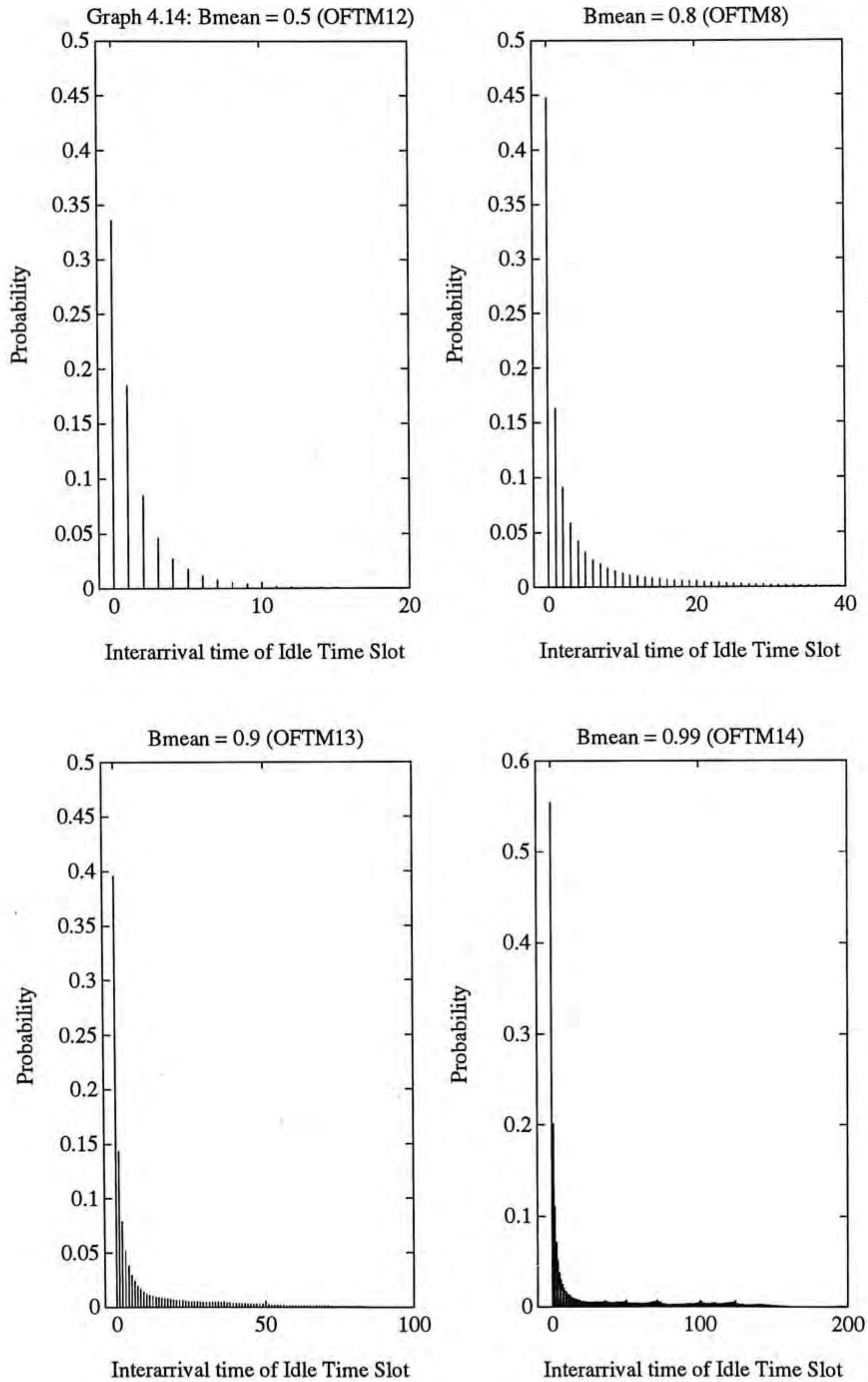

Graph 4.10: Bmean = 0.5 (PTM7)
Probability
Interarrival time of Idle Time Slot
Bmean = 0.8 (PTM8)
Probability
Interarrival time of Idle Time Slot
Bmean = 0.9 (PTM9)
Probability
Interarrival time of Idle Time Slot
Bmean = 0.99 (PTM10)
Probability
Interarrival time of Idle Time Slot

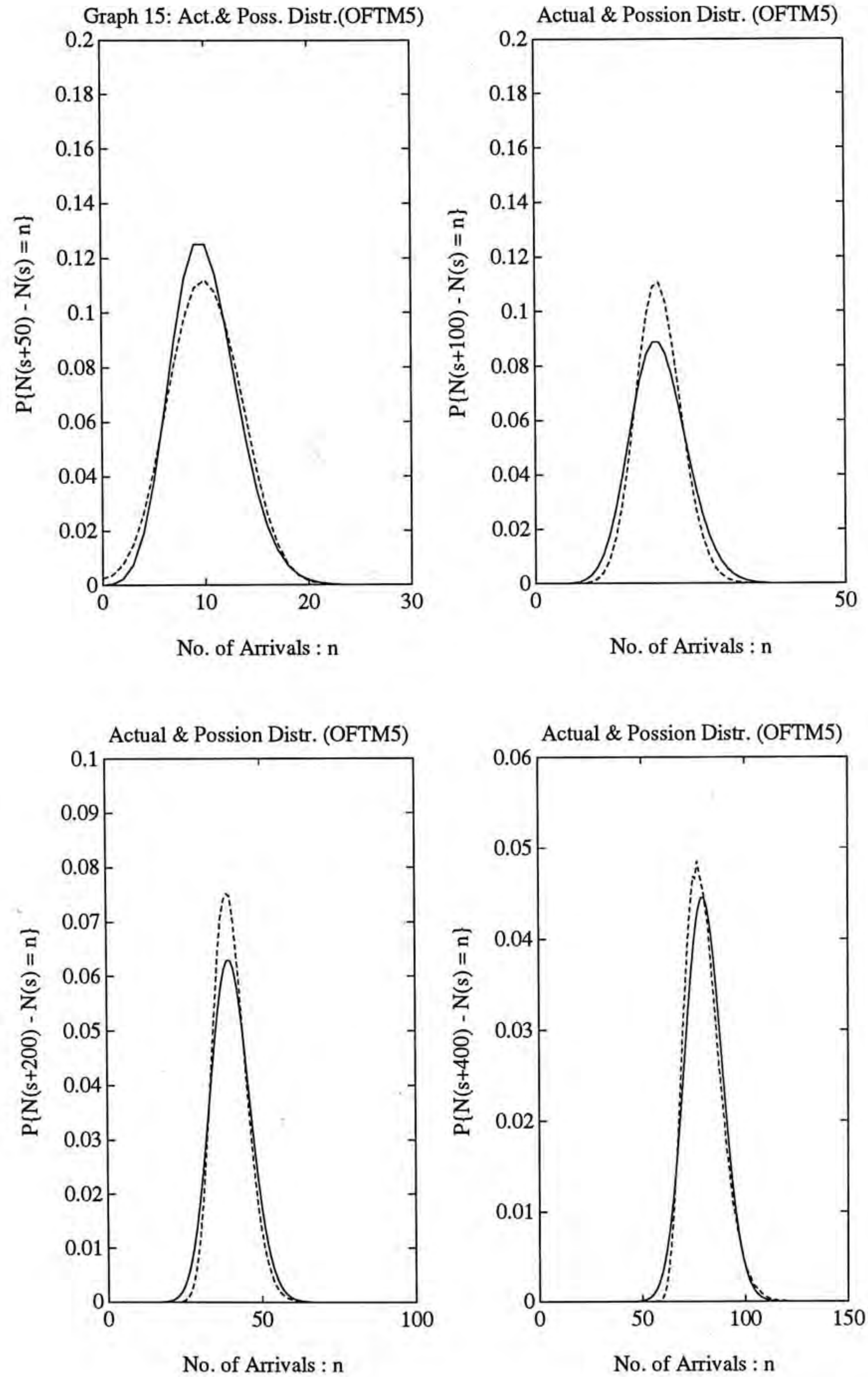

Graph 4.12: Act.& Poss. Distr.(PTM5)
P{N(s+50) - N(s) = n}
No. of Arrivals : n
0
10
20
30
0.2
0.18
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0.02
Actual & Possion Distr. (PTM5)
P{N(s+100) - N(s) = n}
No. of Arrivals : n
50
Actual & Possion Distr. (PTM5)
P{N(s+200) - N(s) = n}
No. of Arrivals : n
100
Actual & Possion Distr. (PTM5)
P{N(s+400) - N(s) = n}
No. of Arrivals : n

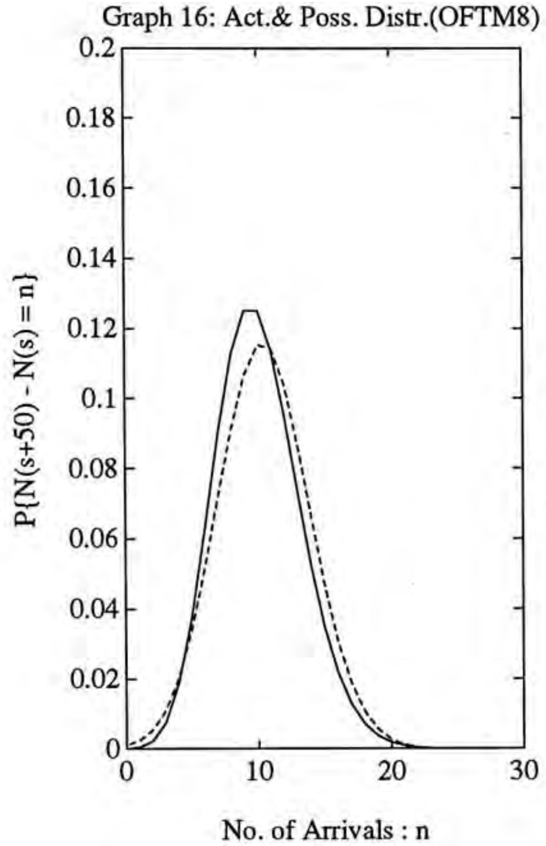

Graph 4.11: Act.& Poss. Distr.(PTM4)
P{N(s+50) - N(s) = n}
No. of Arrivals : n
Actual & Possion Distr. (PTM4)
P{N(s+100) - N(s) = n}
No. of Arrivals : n
Actual & Possion Distr. (PTM4)
P{N(s+200) - N(s) = n}
No. of Arrivals : n
Actual & Possion Distr. (PTM4)
P{N(s+400) - N(s) = n}
No. of Arrivals : n

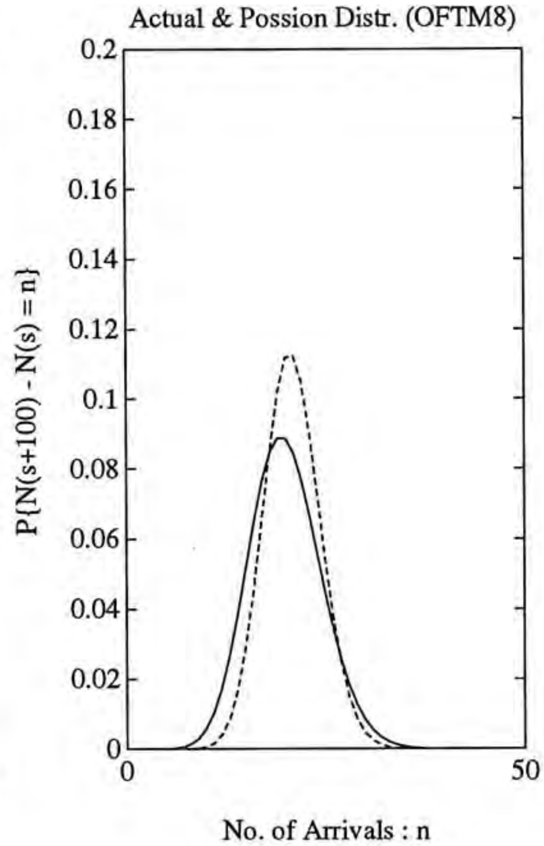
Graph 4.13: Bmean = 0.5 (OFTM9)

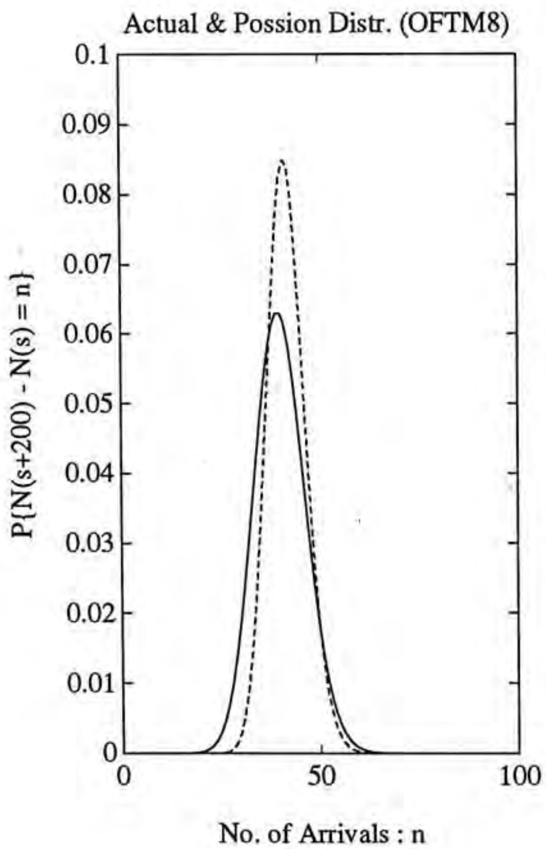


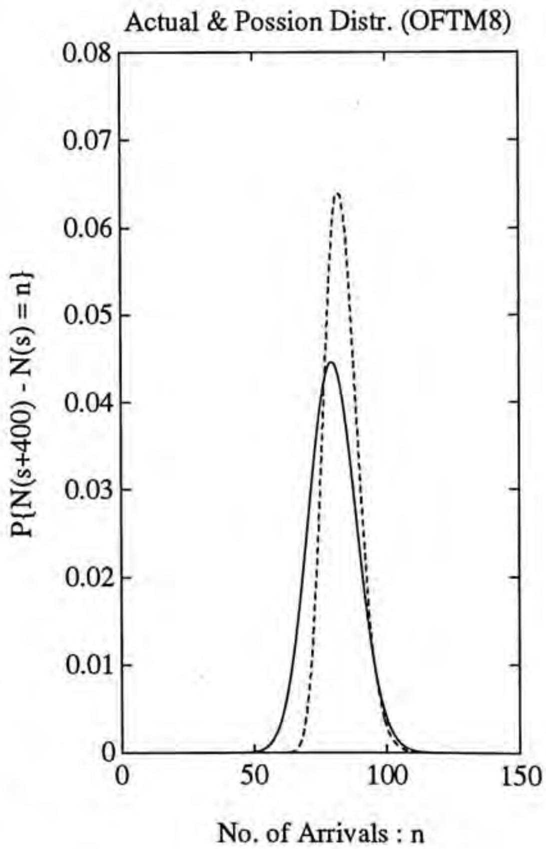

Graph 4.14: Bmean = 0.5 (OFTM12)
Probability
Interarrival time of Idle Time Slot
Bmean = 0.8 (OFTM8)
Probability
Interarrival time of Idle Time Slot
Bmean = 0.9 (OFTM13)
Probability
Interarrival time of Idle Time Slot
Bmean = 0.99 (OFTM14)
Probability
Interarrival time of Idle Time Slot


Graph 15: Act.& Poss. Distr.(OFTM5)
P{N(s+50) - N(s) = n}
No. of Arrivals : n
Actual & Possion Distr. (OFTM5)
P{N(s+100) - N(s) = n}
No. of Arrivals : n
Actual & Possion Distr. (OFTM5)
P{N(s+200) - N(s) = n}
No. of Arrivals : n
Actual & Possion Distr. (OFTM5)
P{N(s+400) - N(s) = n}
No. of Arrivals : n


Graph 16: Act.& Poss. Distr.(OFTM8)
P{N(s+50) - N(s) = n}
0.2
0.18
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
0
10
20
30
No. of Arrivals : n


Actual & Possion Distr. (OFTM8)
P{N(s+100) - N(s) = n}
0.2
0.18
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
0
50
No. of Arrivals : n


Actual & Possion Distr. (OFTM8)
P{N(s+200) - N(s) = n}
0.1
0.09
0.08
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
0
50
100
No. of Arrivals : n


Actual & Possion Distr. (OFTM8)
P{N(s+400) - N(s) = n}
0.08
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
0
50
100
150
No. of Arrivals : n

Chapter 5

Parameters Tuning and Analysis

5.1 Scheme I : Scout Pumping

In this scheme, the threshold level of the scout counter value N_T is two, that is the link will be congested if two DS packets are recorded by the scout counter of a route. Both DS and RS packets are loaded as the same rate as HP packets. The numbers of DS and RS packets (N_{DS} and N_{RS}) in the scout stream are estimated in this section. Figure 5.1 depicts the time instants that the scout packets arrive at different locations along the path of the call, in this case, congestion is detected at the 3^{rd} hop of the call.

It can be shown that,

$$T_{scout} = T_{detect} + T_{reserve} \tag{5.1}$$

$$T_{detect} = (N_{DS} - 1) \cdot c_j \tag{5.2}$$

$$T_{reserve} = N_{RS} \cdot c_j \tag{5.3}$$

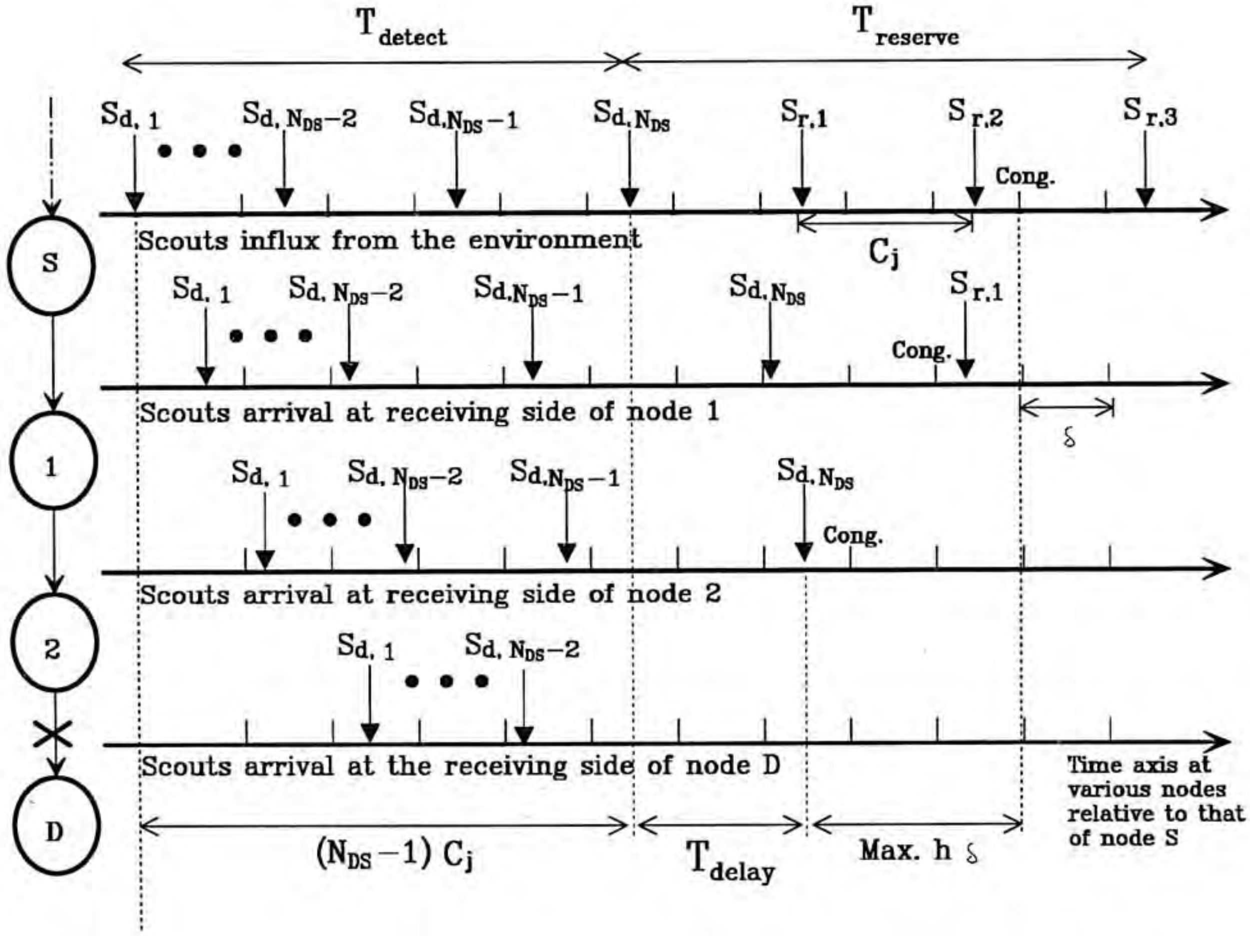


Figure 5.1 Congestion detected at the 3rd hop of type j call by Scheme I. (h=3)

where N_{DS} is the number of DS packets,

N_{RS} is the number of RS packets,

c_j is the periodicity of call type j,

$T_{reserve}$ is the transmission time of RS packets,

T_{detect} is the transmission time of DS packets.

From Figure 5.1, $T_{reserve}$ consists of the delay time of the RS packet from the first link to the link preceding the congested link along the testing path (T_{delay}) and the delay due to the response time of updating the control vector ($h \cdot \delta$).

then,

$$T_{reserve} = \begin{cases} c_j & \text{if } h = 1 \\ \left\lceil \frac{T_{delay} + h \cdot \delta}{c_j} \right\rceil \cdot c_j & \text{if } h > 1 \end{cases} \tag{5.4}$$

Since N_T is only two, the maximum delay time of scout packet per link is limited by the periodicity of the call c_j. Thus the transmission delay time T_{delay} of any scout packet at h hops away can be approximated as,

$$T_{delay} \leq (h-1) \cdot c_j + D, \quad h > 1 \tag{5.5}$$

where h is the number of hops of the call,

$D = \sum_{i=1}^{h-1} d_i$,

d_i is the propagation delay of link at the i^{th} hop.

so,

$$T_{reserve} \leq [(h-1) + \left\lceil \frac{h \cdot \delta + D}{c_j} \right\rceil] \cdot c_j, \quad h > 1 \tag{5.6}$$

and,

$$T_{scout} \leq \begin{cases} N_{DS} \cdot c_j & \text{if } h = 1 \\ ((N_{DS} + h - 2) + \left\lceil \frac{h \cdot \delta + D}{c_j} \right\rceil) \cdot c_j & \text{if } h > 1 \end{cases} \tag{5.7}$$

Remark. N_{RS} can be calculated from $T_{reserve}$ and it is dependent on the values of variables c_j and h. In practical implementation, all the possible values of N_{RS} should be preprocessed and stored in fast data memory for retrieval at the time of call set up.

To avoid serious packet loss due to buffer overflow, the *probability of overrun* should be bounded by a minimal value. Overrun means that a call is accepted even if the call bandwidth is greater than the bandwidth available in the link. The value of N_{DS} has to be analyzed to get a minimal overrun probability.

Assumptions for analysis :

1. Throughout the call set up period, no more than one stream of scout packets is loading onto the same links, i.e. No scout stream collision.
2. The interarrival time of scout packet is periodic with time c_j.
3. The upper bound of burstiness of idle time slot interarrival time of PTM and OFTM with low burstiness can be approximated by Poisson Process with mean rate λ equal to the mean available bandwidth in the link [1].

[1]Refer to the assumptions of the Traffic models in Chapter 4.

In testing the availability of each link of the testing path by a type j call, between every interarrival time of DS packets, there are at least one arrival of idle time slot. Thus,

$$P\{\text{Overrun of type } j \text{ call}\} \leq (1 - e^{-\lambda c_j})^{N_{DS}} \tag{5.8}$$

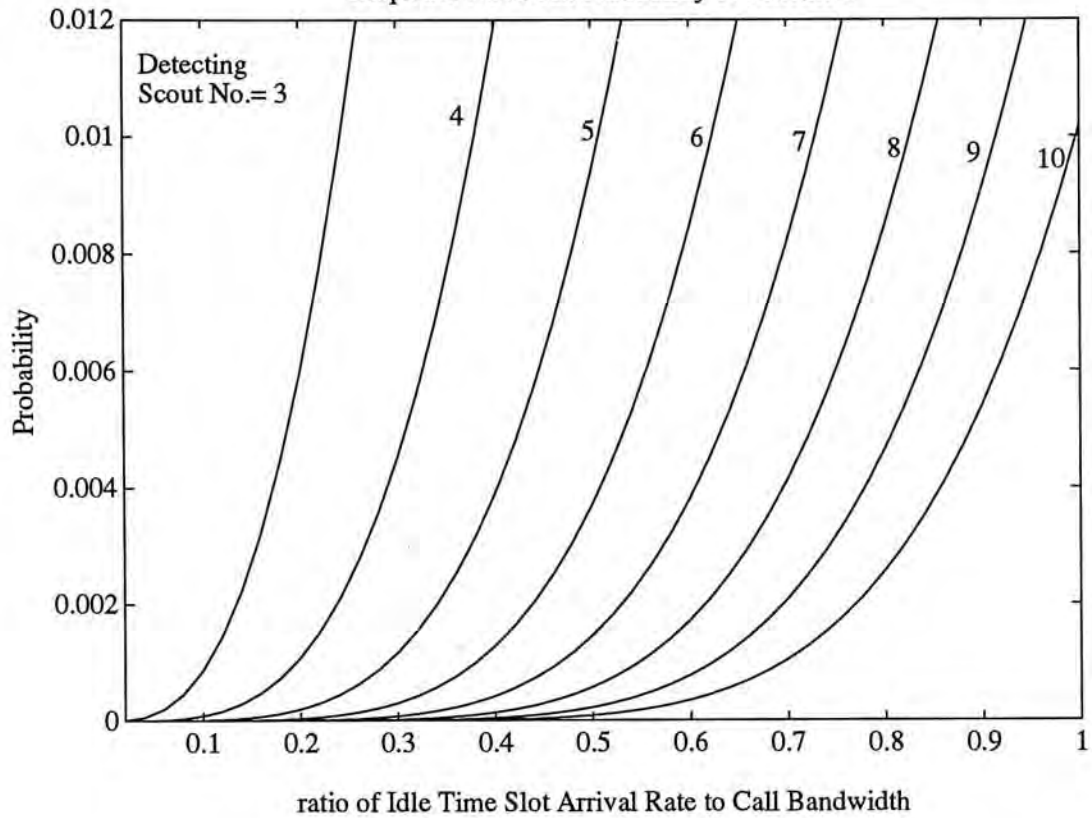
where $\lambda < \frac{1}{c_j}$.

Graphs 5.1 and 5.2 report the relation among the maximum overrun probability, ratio of the available link bandwidth to call bandwidth (λc_j) and number of DS packets (N_{DS}). Therefore, the values of N_{DS} for different value of λc_j can be estimated. If the values of N_{DS} are chosen to be 5 and 15, the probabilities of accepting a call in different values of call bandwidth and available link bandwidth can be obtained from 3D-plots (Graphs 5.1.1-2) and their contour-plots (Graphs 5.2.1-2). To achieve small overrun probability, N_{DS} has to be large enough but large N_{DS} may leads to higher blocking probability.

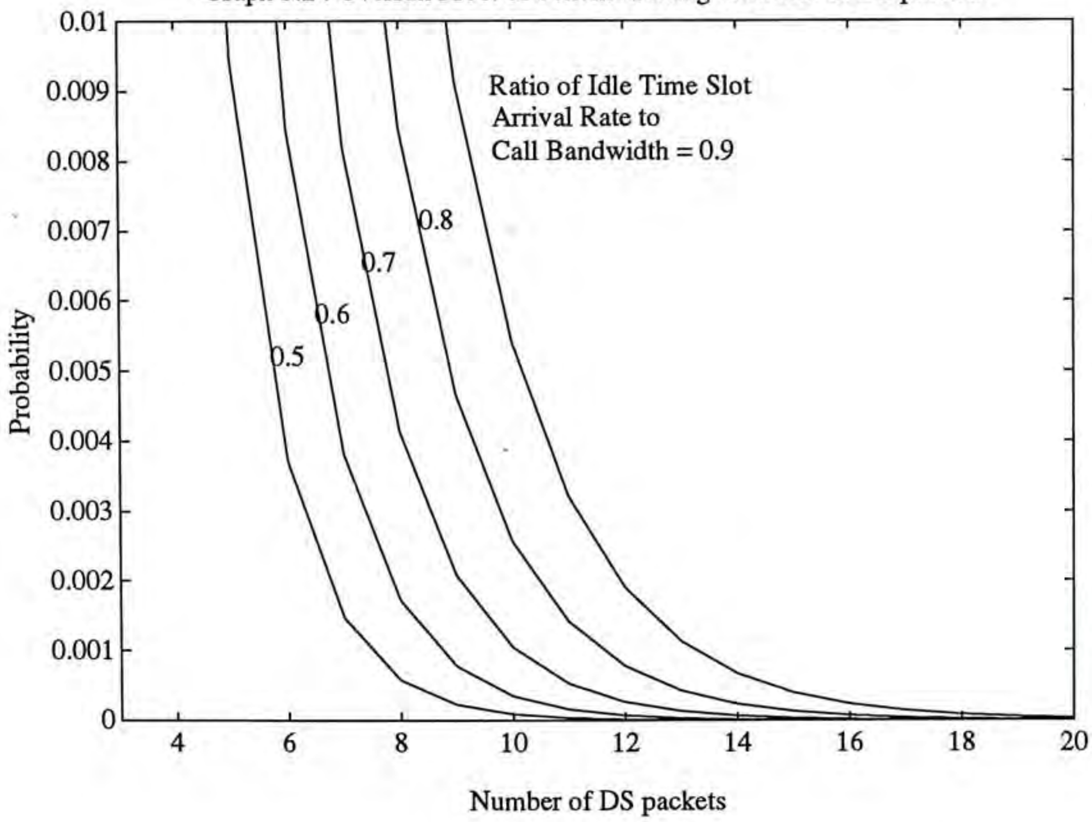
On the other hand, suitable values of N_{DS} can be estimated by simulation. The simulation model describes that a number of DS packets are loaded onto a link under PTM. By setting the mean bit rate of PTM marginally lower than a level that a selected call type cannot be set-up, the probabilities of wrongly accepting a call by different number of DS packets can then be determined. The input traffic is recorded as interarrival time of idle time slot which is stored in a file so that the same input traffic can be reused by different simulation trials.

The values of N_{DS} can be estimated using two sets of PTM models in the simulation as shown in Table 5.1. It is observed from Graphs 5.3-4 that if the

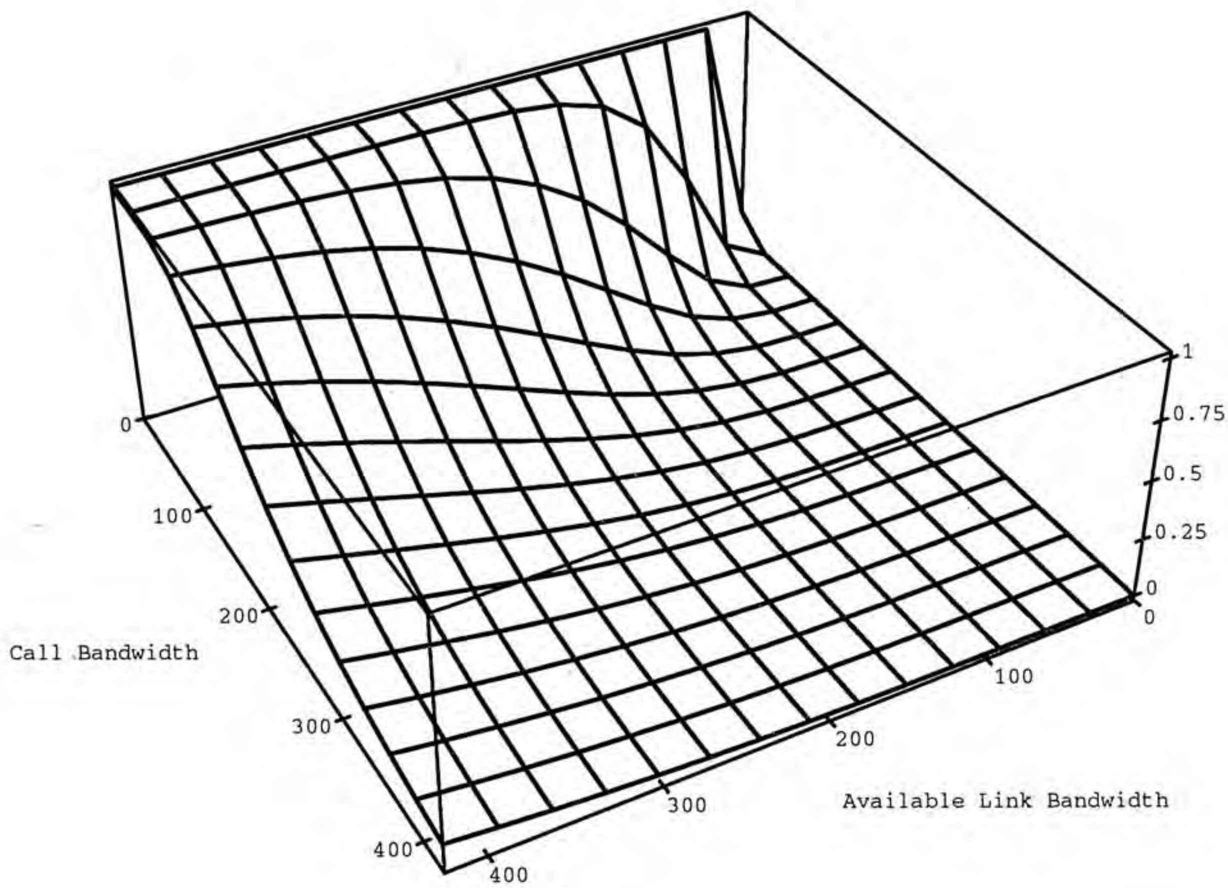
Graph 5.1 : Overrun Probability of Scheme I



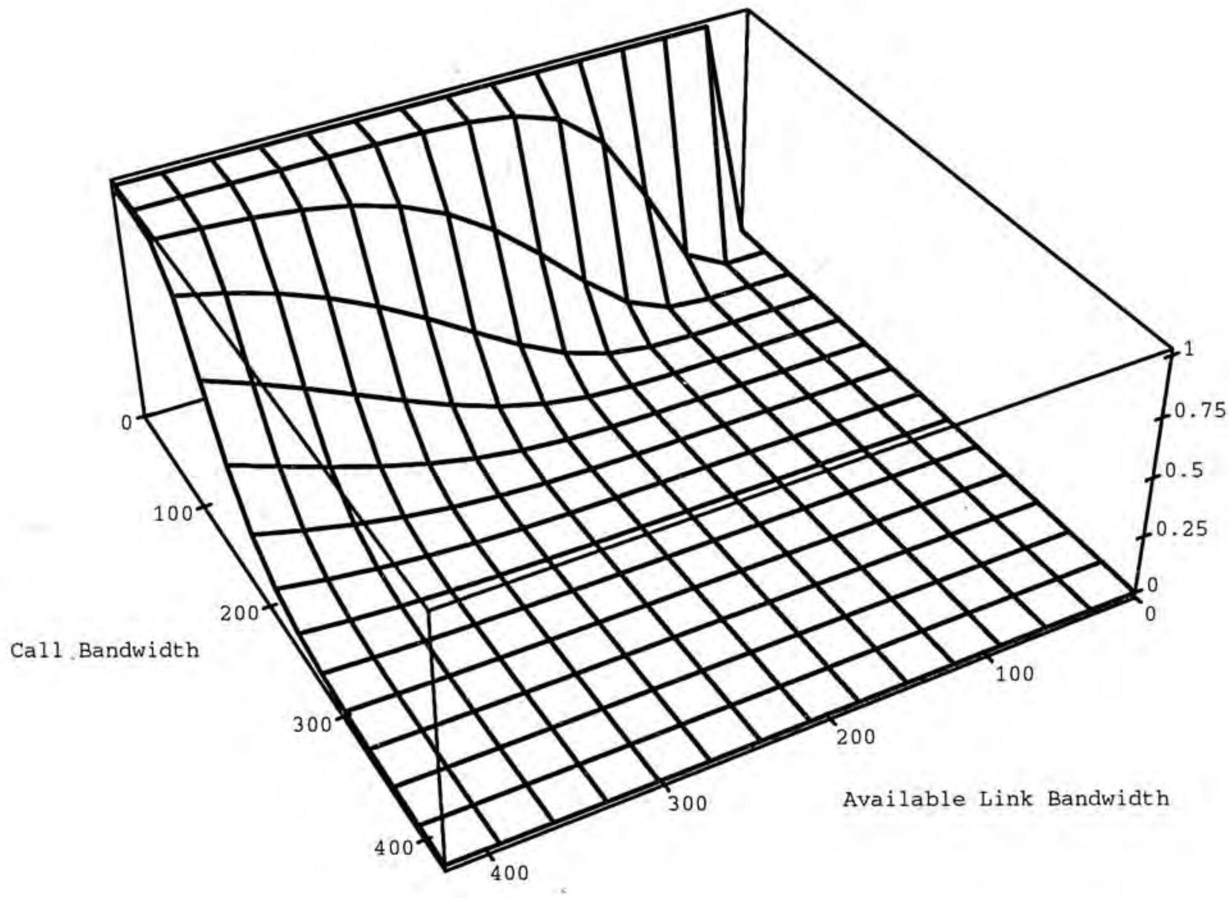
Graph 5.2 : Overrun Prob. of Scheme I using Diff. No. of DS packets



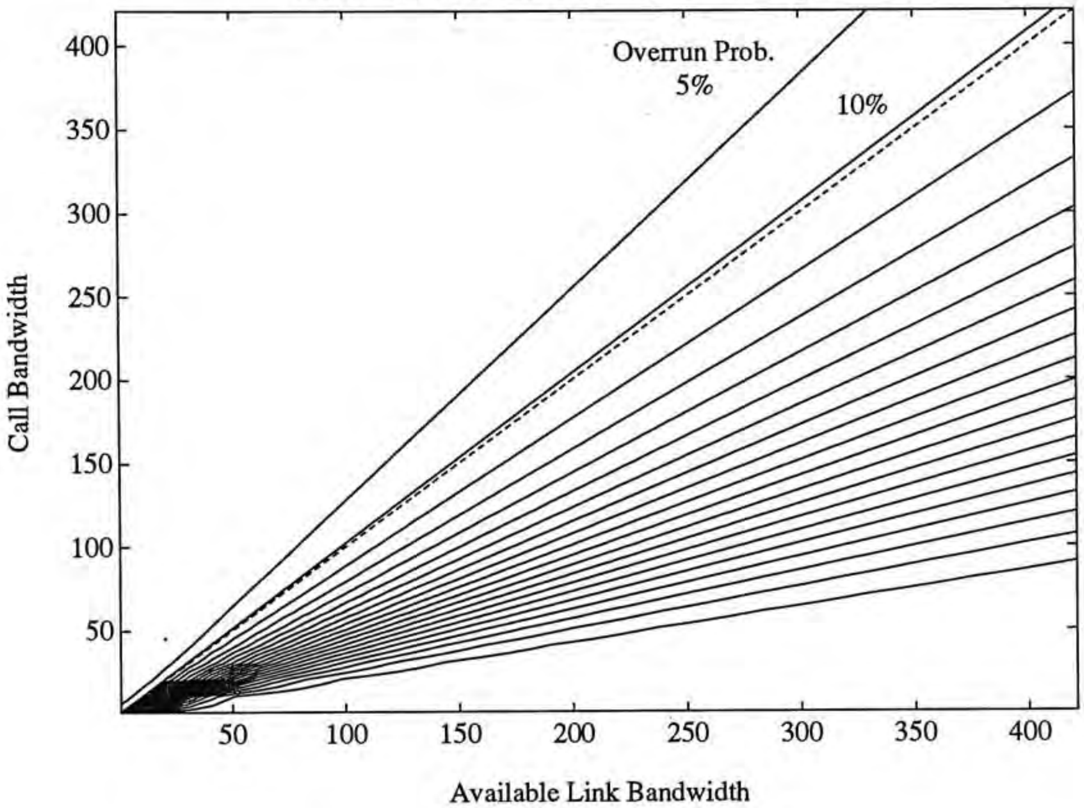
Graph 5.1.1: Prob. of accepting a Call (5 DS packets)



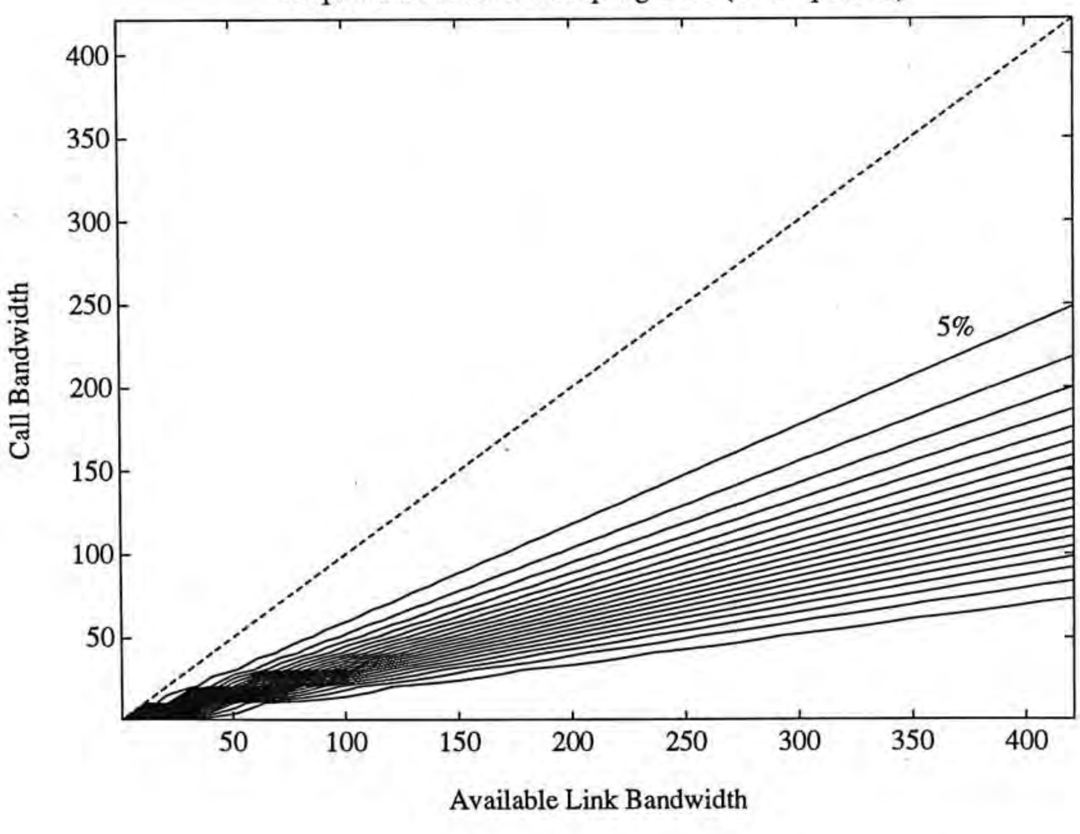
Graph 5.1.2: Prob. of accepting a Call (15 DS packets)



Graph 5.2.1 : Prob. of Accepting a call (5 DS packets)



Graph 5.2.2 : Prob. of Accepting a call (15 DS packets)



traffic is dominated by Type I call, the values of N_{DS} are smaller than that of the evenly distributed traffic except for setting up type I call. It is because type I dominating traffic have stronger periodicity than the evenly distributed traffic. The values of N_{DS} obtained in this section provide testing data for tuning the parameters of the network simulator described in the Chapter 6.

Table 5.1

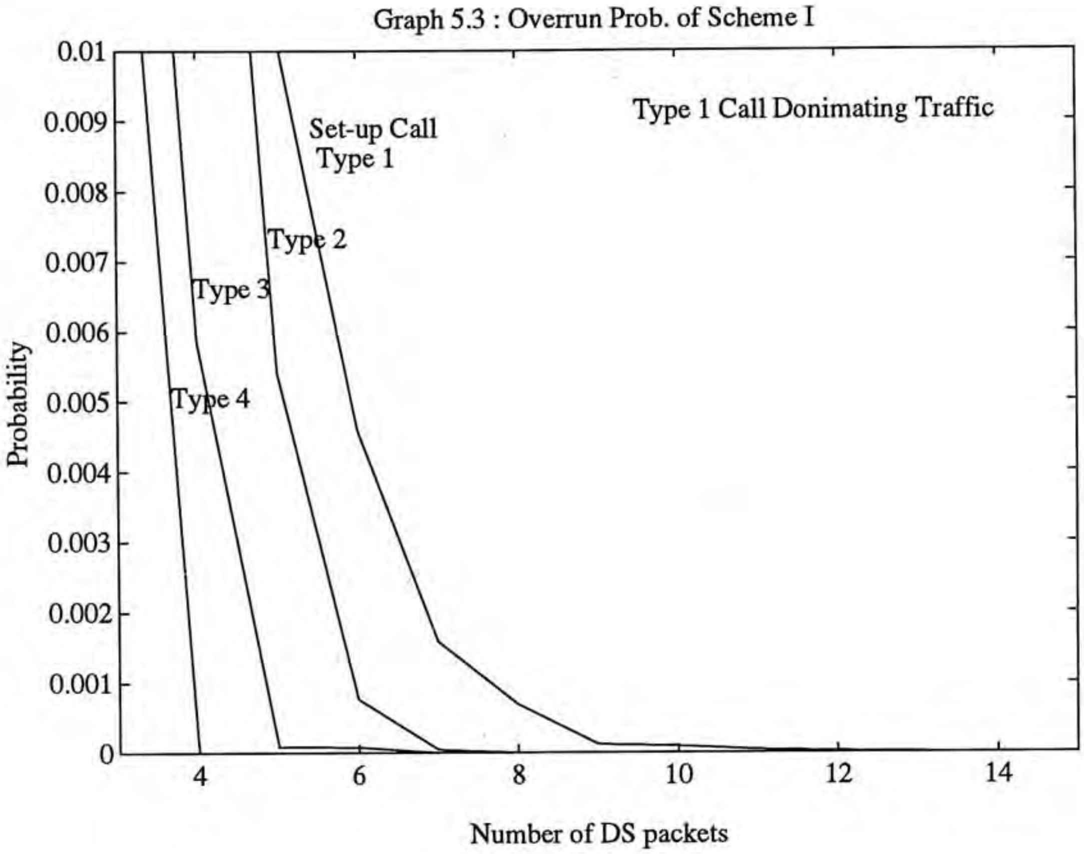
$[T_1, T_2, T_3, T_4] = [125, 72, 51, 37]$, $[N_1, N_2, N_3, N_4]$ of PTM :

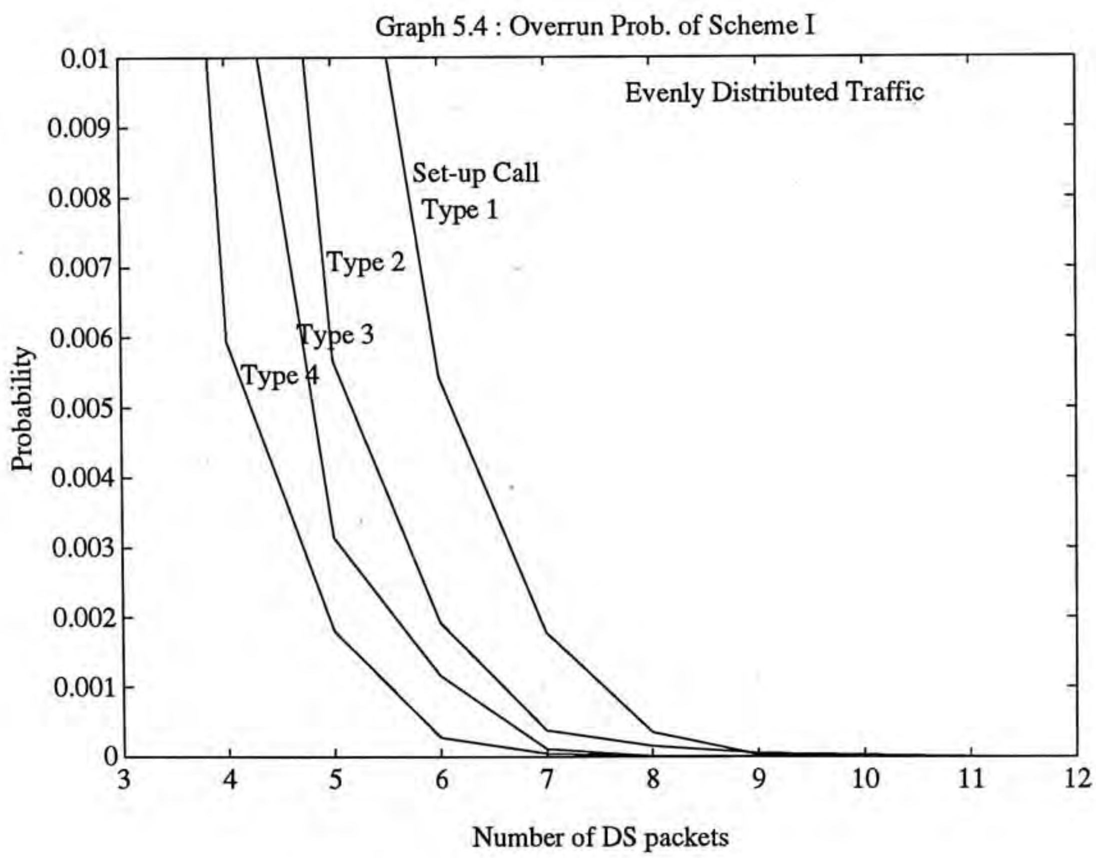
Set-up Call Type	Type I Call Dominating traffic	B_{mean}
1	$PTM11 = [98, 6, 5, 1]$	0.9924
2	$PTM12 = [99, 5, 5, 1]$	0.9865
3	$PTM13 = [99, 6, 4, 1]$	0.9808
4	$PTM14 = [99, 6, 5, 0]$	0.9734

Set-up Call Type	Evenly Distributed traffic	B_{mean}
1	$PTM15 = [29, 18, 15, 8]$	0.9923
2	$PTM16 = [30, 17, 15, 8]$	0.9864
3	$PTM17 = [30, 18, 14, 8]$	0.9807
4	$PTM18 = [30, 18, 15, 7]$	0.9733

5.2 Scheme II : Speed-up Scout Pumping

DS packets from call type j are loaded at a speed-up rate $R_{sp,j}$, so that they will be cumulated at the busy link up to a threshold level $N_T (\geq 2)$ within a predefined period of time T_{detect}. The speed-up rate is formulated as the following.


Graph 5.3 : Overrun Prob. of Scheme I
Type 1 Call Donimating Traffic
Set-up Call
Type 1
Type 2
Type 3
Type 4
Probability
0.01
0.009
0.008
0.007
0.006
0.005
0.004
0.003
0.002
0.001
0
4
6
8
10
12
14
Number of DS packets


Graph 5.4 : Overrun Prob. of Scheme I
Evenly Distributed Traffic
Set-up Call
Type 1
Type 2
Type 3
Type 4
Probability
0.01
0.009
0.008
0.007
0.006
0.005
0.004
0.003
0.002
0.001
0
3
4
5
6
7
8
9
10
11
12
Number of DS packets

$$R_{sp,j} = R_{c,j} + \epsilon + h \cdot \Delta \qquad \text{where } \Delta = \frac{N_T \cdot N_p}{T_{detect}} \tag{5.9}$$

where $R_{sp,j}$ is the Speed-up rate of DS packets of type j call,

$R_{c,j} = \frac{1}{c_j}$ is the rate of the intended call of type j,

ϵ is the bandwidth allowance,

h is the number of hops of the intended call,

N_p is the packet size.

Comparing with scheme I, the pumping rate of scout packets increases by $\epsilon + h \cdot \Delta$. Since the outgoing bit rate of the link may not be constant throughout the time of scout packets transmission, a constant value ϵ is used as the bandwidth allowance. This bandwidth is sacrificed to reduce the overrun probability due to traffic fluctuation within the time of call set up.

The term $h \cdot \Delta$ of the above equation is to guarantee that mean rate of scout packet accumulation is high enough to set the scout counter to N_T within T_{detect}. The bit rate of each link can be regarded as constant at the time of call set-up because the effect of traffic fluctuation is absorbed by the bandwidth allowance ϵ. Therefore, if the outgoing bit rate is smaller than the incoming bit rate by more than Δ (that is not enough bandwidth for setting up the call), the time for accumulation of scout packets up to N_T is guarantee to be less than T_{detect}. Since the retarding speed of scout packet in each node is less than or equal to Δ, the speed-up rate of $h \cdot \Delta$ is to ensure that the average outgoing rate of scout packets at any link along the route of the call is more than or equal to $R_{c,j} + \epsilon$. Figure 5.2 portrays the rate of scout packet at different node for $h = 3$.

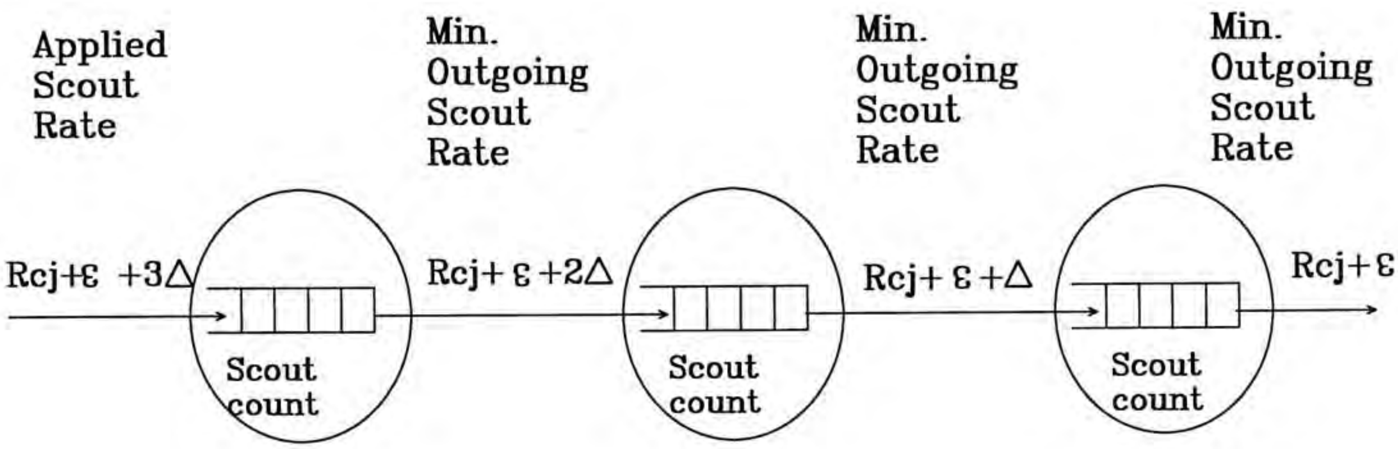


Figure 5.2 Applied Scout Rate to test availiability of the route.

From Figure 5.3, $T_{reserve}$ can be found by the worst case approximation,

$$T_{reserve} = \begin{cases} c_j & \text{if } h = 1 \\ \left\lceil \dfrac{T_1 + T_2 - T_{detect} + h \cdot \delta}{c_j} \right\rceil \cdot c_j & \text{if } h > 1 \end{cases} \tag{5.10}$$

T_1 is the transmission delay time of the first DS packet from the sender to the node that is $(h-1)$ hop away from it. T_2 is the time difference between the first and the last DS packet arrivals at the congested link. The minimum outgoing scout rate at links of different hop can be determined by the values $R_{sp,j}$ and $\triangle$. Therefore, the maximum values of T_1 and T_2 can be deduced.

$$T_1 \le \sum_{i=1}^{h-1} (d_i + \frac{1}{R_{c,j} + \epsilon + i\Delta})$$

$$T_2 \le (\frac{R_{c,j} + \epsilon + h\Delta}{R_{c,j} + \epsilon + \Delta}) \cdot T_{detect}$$

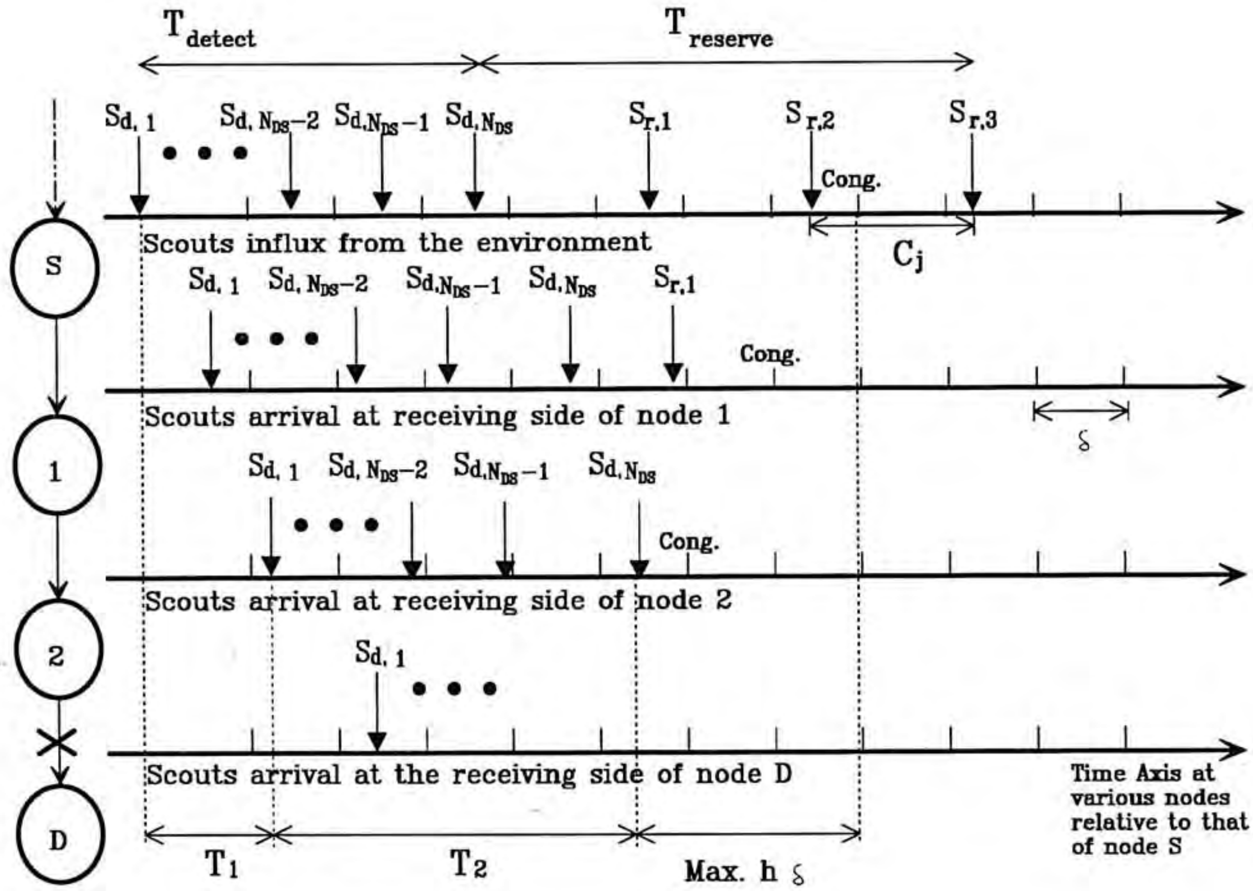


Figure 5.3 Congestion detected at the 3rd hop of type j call by Scheme II. (h=3)

thus,

$$T_{reserve} \leq \left\lceil \frac{\sum_{i=1}^{h-1}(d_i + \frac{1}{R_{c,j} + \epsilon + i\Delta}) + \frac{(h-1)\Delta}{R_{c,j} + \epsilon + \Delta} \cdot T_{detect} + h \cdot \delta}{c_j} \right\rceil \cdot c_j \quad (5.11)$$

where $h > 1$, and

$$T_{scout} \leq \begin{cases} T_{detect} + c_j & \text{if } h = 1 \\ \left\lceil \frac{\sum_{i=1}^{h-1}(d_i + \frac{1}{R_{c,j} + \epsilon + i\Delta}) + \frac{R_{c,j} + \epsilon + h\Delta}{R_{c,j} + \epsilon + \Delta} \cdot T_{detect} + h \cdot \delta}{c_j} \right\rceil \cdot c_j & \text{if } h > 1 \end{cases} \quad (5.12)$$

5.3 Blocking Probability

Applying the assumptions in the Section 5.1, the blocking probabilities of scheme I is going to compare with that of scheme II. Blocking probability is the probability of the intended call being refused under the given traffic loading.

For scheme I, a scout packet stream will be blocked if no idle time slot is present within any interarrival time of two DS packets. The interarrival period of DS packets is equal to the period of the intended HP call c_j, so the blocking probability for scheme I by N_{DS} DS packets is

$$\begin{aligned} P\{\text{Blocking of type } j \text{ call}\} &\geq 1 - (1 - e^{-\lambda c_j})^{N_{DS}} \quad (5.13) \\ &= P_I \end{aligned}$$

For scheme II, since the packet rate of DS packets is increased and the $N_T \geq 2$, it is difficult to get the exact form of blocking probability for scheme II. However, for the purpose of comparison, an upper bound of the blocking probability is approximated. The upper bound is the probability that the scout packet stream consists of at least N_T interarrival periods that are without idle time slots arrivals.

$$\begin{aligned} P\{\text{Blocking type } j \text{ call}\} &\leq 1 - \sum_{r=0}^{N_T-1} \binom{N_{DS}}{r} \left(e^{-\lambda y_j}\right)^r (1 - e^{-\lambda y_j})^{N_{DS}-r} \\ &= P_{II}, \qquad \text{where } y_j = \frac{1}{R_{sp,j}} \end{aligned} \tag{5.14}$$

Remarks.

1. The variables y_j, N_{DS} and N_T are inter-related according to Eqn. (5.9). If N_T increases, N_{DS} (i.e. T_{detect}) has to increase to maintain $R_{sp,j}$. The goal is to get a suitable N_T so that the blocking probability can be minimized but T_{detect} and $R_{sp,j}$ will not be too great.

2. Graph 5.5 shows the values of P_{II} $(y_j = 0.4, N_{DS} = 25)$ of using different threshold levels N_T under different idle time slot arrival rates λ.

3. it can be observed from Graph 5.6 that for the same T_{detect}, P_{II} $(y_j = 0.4, N_{DS} = 25)$ can be adjusted by tuning the variable N_T to 10 so that it is smaller than P_I $(c_j = 1, N_{DS} = 10)$.

4. Scheme II can outperform (by Parameters tuning) Scheme I because it can tolerate traffic fluctuation during the time of call set-up. In other

words, scheme I is much more conservative than Scheme II. However, the complexity requirement of scheme II is higher than that of Scheme I.

5.4 Scout Stream Collision

Collision occurs when DS packets from more than one call are loading onto the same link. Clearly, collision will increase the blocking probability of the calls because of the increase in DS packet rate. This section is going to find out how probable two scout packet streams will collide and how likely two collided calls will be refused for admission. Take one unit time as the time of loading one packet. If DS packets take Y unit time for transmission and two scout packet streams collide for X unit time. Then the probability of refusal for call set-up due to scout packet stream collision P_T is,

$$P_T = \sum_{X=1}^{Y} [P\{\text{set-up refusal} \mid \text{Collision period} = X\} \cdot P\{\text{Collision period} = X\}] \tag{5.15}$$

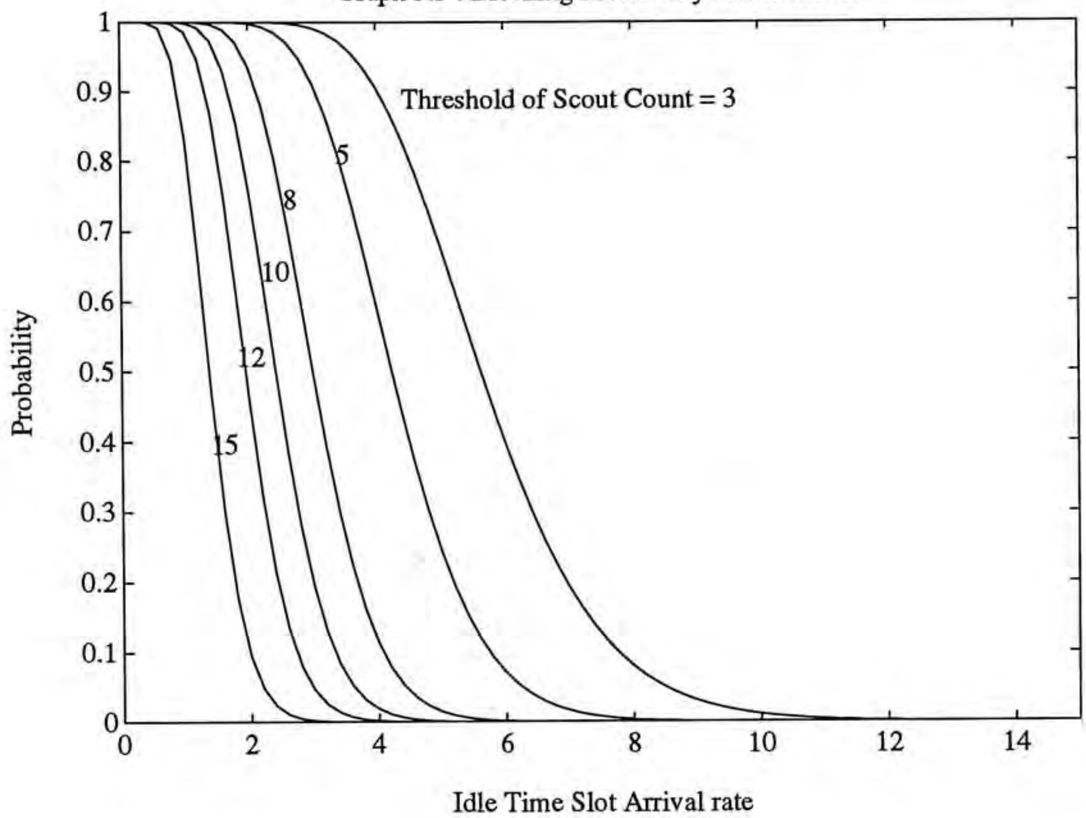
Assume that the interarrival time of the calls at link k is exponentially distributed with mean λ_k. DS packet streams will only collide if their interarrival time is shorter than the length Y.

Thus,

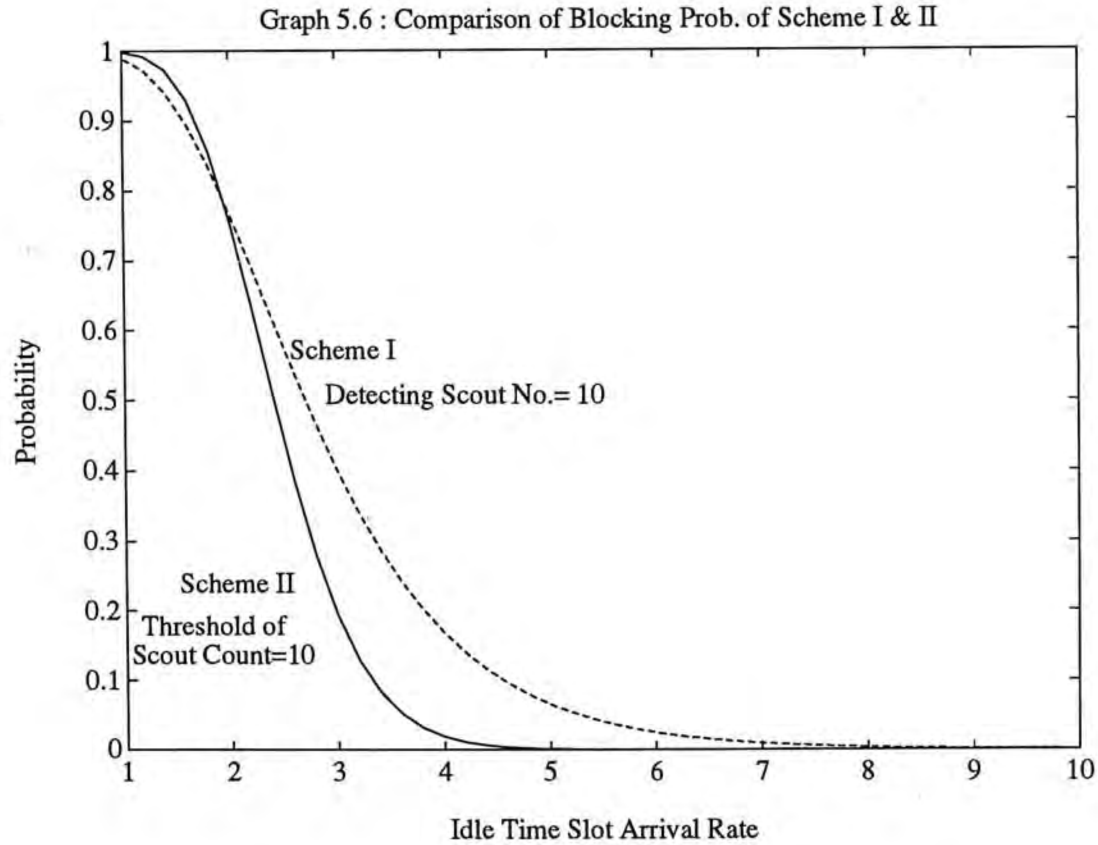
$$P\{\text{Collision period} = X\} = \lambda_k e^{-\lambda_k (Y-X)} \tag{5.16}$$

Consider the case of scheme I under PTM. Figure 5.4 depicts collision of two scout packet streams. $\{x_i\}_{i=1}^{N}$ is a set of inter-scout arrival time and $X = \sum_{i=1}^{N} x_i$.

Graph 5.5 : Blocking Probability of Scheme II



Graph 5.6 : Comparison of Blocking Prob. of Scheme I & II



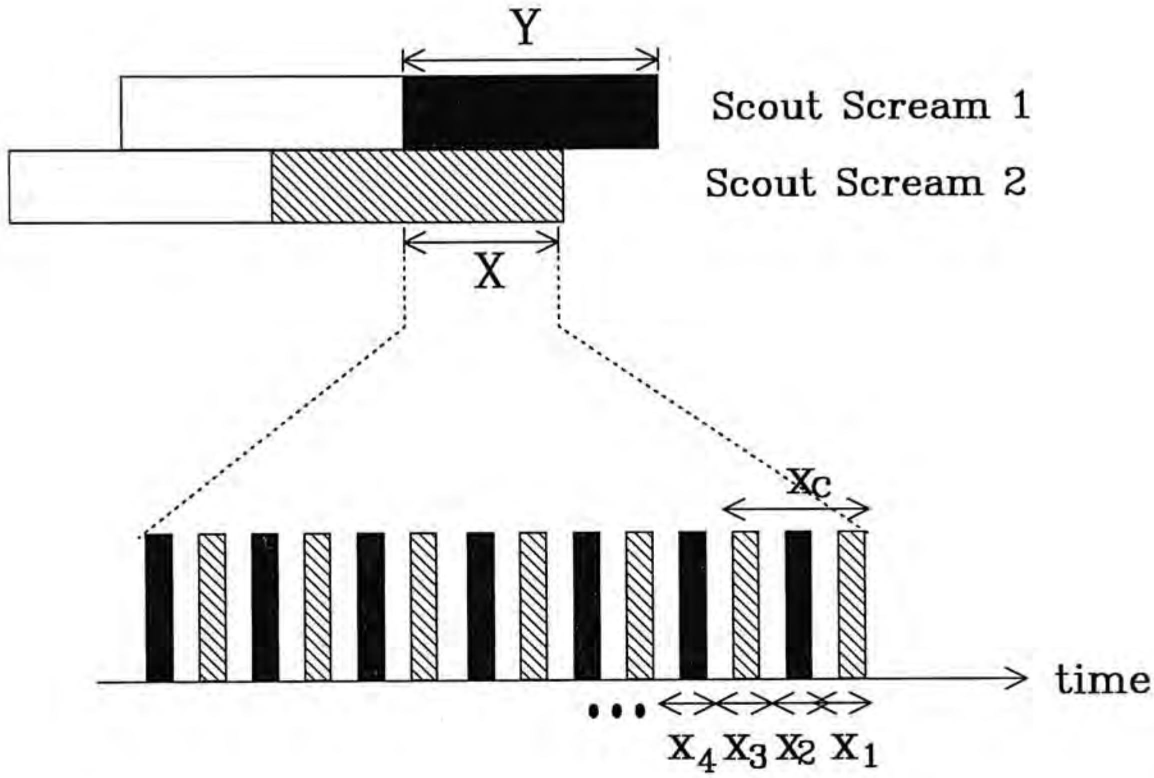


Figure 5.4 Collision between Scout Streams

The calls will both be terminated if there are no idle time slot within any x_i. Applying the assumption that the arrival of idle time slot is a Poisson Process with mean λ at short time interval,

$$P\{\text{Set-up refusal} \mid \text{Collision period} = X\} = 1 - \prod_{i=1}^{N}(1 - e^{-\lambda x_i}) \qquad (5.17)$$

Averaging the above probability for different set of x_i,

$$P\{\text{set-up refusal} \mid \text{Collision period} = X\} = \frac{1}{x_c}\sum_{x_1=1}^{x_c-1}\left[1 - \prod_{i=1}^{N}(1 - e^{-\lambda x_i})\right]$$

$$\approx \frac{1}{x_c} \sum_{x_1=1}^{x_c-1} \sum_{i=1}^{N} e^{-\lambda x_i}$$

Therefore,

$$P_T \approx \sum_{X=1}^{Y} \left[\left(\frac{1}{x_c} \sum_{x_1=1}^{x_c-1} \sum_{i=1}^{N} e^{-\lambda x_i} \right) \left(\lambda_k e^{-\lambda_k (Y-X)} \right) \right] \tag{5.18}$$

If the scout packet streams are of the same type with rate $\frac{1}{x_c}$, then

$$\begin{aligned} N &= \frac{2X}{x_c} \\ \sum_{i=1}^{N} e^{-\lambda x_i} &= \frac{N}{2} (e^{-\lambda x_1} + e^{-\lambda (x_c - x_1)}) \end{aligned} \tag{5.19}$$

and,

$$P_T \approx \sum_{X=1}^{Y} \left[\left(\frac{X}{x_c^2} \sum_{x_1=1}^{x_c-1} (e^{-\lambda x_1} + e^{-\lambda (x_c - x_1)}) \right) \left(\lambda_k e^{-\lambda_k (Y-X)} \right) \right] \tag{5.20}$$

Remarks. It is intuitively clear that calls will be less likely to be refused for admission under scheme II situation. From Eqn.(5.20), the most dominating variable in getting P_T is λ_k. The actual interarrival rate of HP calls in a link is a very small value in comparing with the unit time because HP calls usually last for minutes. For the most pessimistic case for both schemes, it is found that $P_T < Y \cdot \lambda_k$. For example, if $Y = 2000$ unit time and $\lambda_k = 10^{-6}$ per unit time, then $P_T < 0.2\%$. Therefore, the throughput loss due to scout stream collision is not significant.

Chapter 6

Simulation Modelling & Performance Evaluation

6.1 The Network Simulator

Since it is computationally prohibitive to estimate the performance of the proposed multi-hop call set-up and traffic control algorithms using analytic methods, a *network simulator* is built to serve the purpose. Since both packet and call level of traffic control are carried out by the simulator, the simulator has to operate for tremendous amount of simulation time so as to get the statistically stable results. The packet loading time is chosen to be the simulation unit time, because it is the shortest time to schedule an event. For example, if the link bandwidth is 424 Mbps and packet size is the same as size of a ATM cell, then the unit time is $1\mu s$. The challenge of simulating operations of broadband network is the ratio of packet loading time to mean call duration because the call usually takes minutes to terminate.

For the sake of efficiency, instead of using general simulation language, a simulator is designed to speed up the execution of simulation. The network simulator is written in C programming language and operations of broadband network are modelled by the method of discrete event simulation. The network simulator is tested thoroughly to guarantee its correctness and the performance of the network are studied by tuning the network parameters extensively. These validating and tuning processes are extremely time consuming. In order to reduce simulation time at development phase of the simulator, the size of packets is set to be larger than the size of ATM cell, say 4240 bits. Thus, the simulation time decrease proportion to the increase in packet size. Afterwards, packets resume to size of ATM cell for further testing.

The simulation jobs are replicated and assigned to a number of over 30 workstations (including SUN SPAR-10, DEC 3100, DEC 5000/25, /125, /200, /133) with different input parameters. The simulation time for each simulation run is lengthy. The ratio of the simulation time to the actual time on the fastest available workstation, i.e. SUN SPAR-10, is found to be about 1500.

6.1.1 Simulation Event Scheduling

Event scheduling is one of the time consuming process in simulation and a correct choice of data structure can speed up the process. The network model is evaluated in discrete time domain by using a short calendar of consecutive unit time to schedule every simulation event. The calendar is represented by a circular array of event lists which are numbered as $1, 2, \ldots, L_{calen}$. Each event list consists of event structures that are scheduled at the same unit time, see

Figure 6.1. The system clock advances cyclically to the next non-empty event list.

For modelling of periodic traffic calls, events are scheduled periodically according to the packet rate of the calls, so short calendar is usually sufficient. For example, if the peak rate of the lowest bandwidth call (voice-based call) is 3.392 Mbps, then for a 424 Mbps bandwidth link, a packet is loaded in every $125\mu s$. So the length of the calendar (L_{calen}) is 125 unit time. However, a long calendar is needed to model the off-period of on-off traffic sources. To keep the calendar short, an extra field is defined in the each event structure to indicate when the event is ready to be processed. This field is known as the scheduled period (S) of the event (Figure 6.2). When an event is scheduled, the event information and value of the scheduled period (S) is assigned to the event structure. Then it is inserted to the rear of the E^{th} event list.

$$E = (\text{Scheduled Time} - \text{Current Time}) \quad \text{Modulus} \quad L_{calen}$$

$$S = \left\lfloor \frac{(\text{Scheduled Time} - \text{Current Time})}{L_{calen}} \right\rfloor$$

where $\lfloor x \rfloor$ denotes the largest integer smaller than x.

Current events in the list are examined, if S is equal to zero, the event is processed instantly. System variables will then be updated and new events may be scheduled, otherwise the event will remain in the list with the value of S subtracted by 1. After all current events are processed, next non-empty event list will be processed similarly. No outdated events remain in the calendar, only current and future events are stored. Although the granularity of the simulator

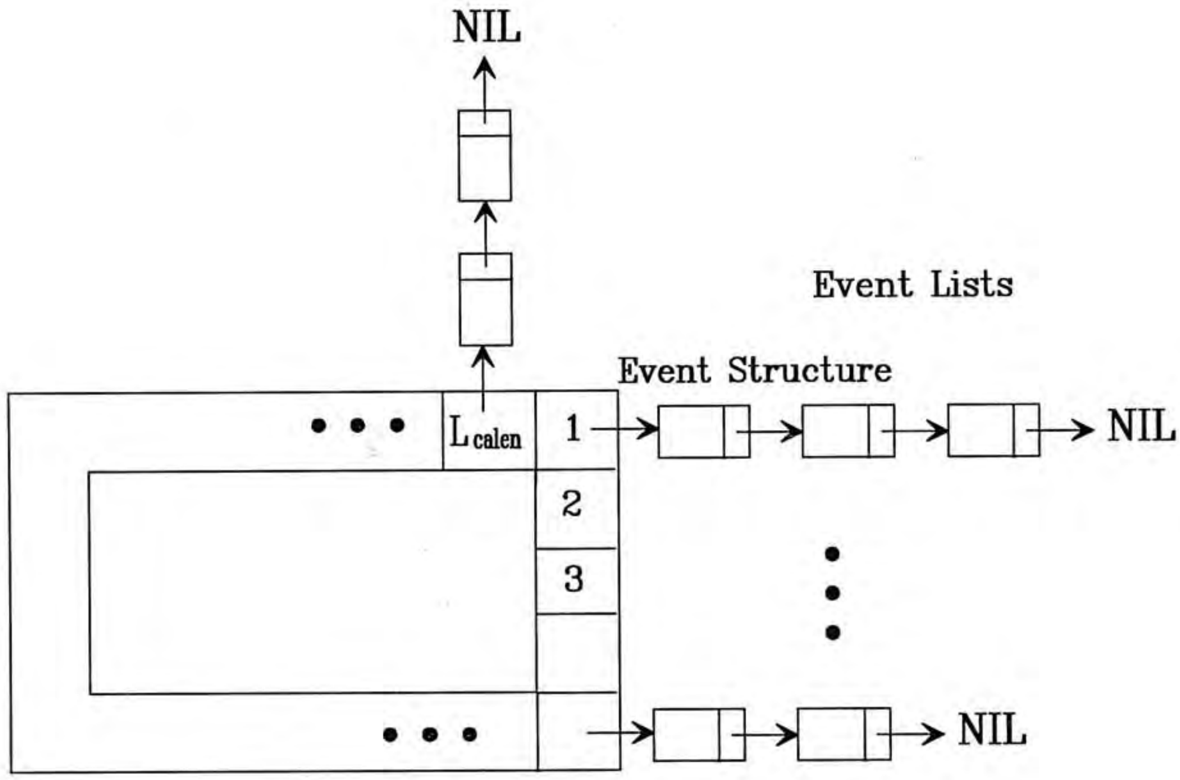


Figure 6.1. Event Calender of Network Simulator

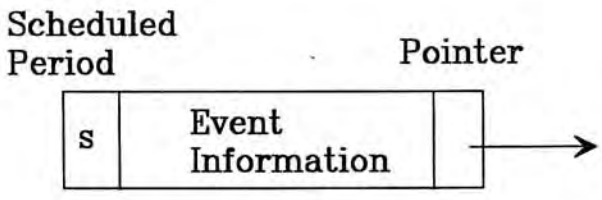


Figure 6.2. Event Structure

is confined by the simulation unit time, the obvious merit of this data structure is to avoid using excessive time for searching the latest event.

6.1.2 Input Traffic Regulation

A balanced load is important to investigate the performance of the network under the call set-up algorithms. Balanced load means that the input traffic loading of each link is the same. However, the topology of the network is arbitrary and the calls are multi-hop, so the average input traffic can only be balanced by an *input traffic regulation matrix*. The matrix which is defined as below can alter proportion of traffic intensity on each route. Assume that all the links in the network have the same bandwidth B.

Input Traffic Regulation Matrix for HP calls of a N_{nd}-node network:

$$M_{in}^{h} = \begin{bmatrix} 0 & \alpha_{12}^{h} & \alpha_{13}^{h} & \cdots & \alpha_{1n}^{h} \\ \alpha_{21}^{h} & 0 & \alpha_{23}^{h} & \cdots & \alpha_{2n}^{h} \\ \alpha_{31}^{h} & \alpha_{32}^{h} & 0 & \cdots & \alpha_{3n}^{h} \\ \vdots & & & \ddots & \vdots \\ \alpha_{n1}^{h} & \alpha_{n2}^{h} & \alpha_{n3}^{h} & \cdots & 0 \end{bmatrix}$$

where α_{sd}^{h} = mean rate of Poisson arrival of HP call from node s to d, $s \neq d$

$s = 1, 2, \ldots, N_{nd}$ and $d = 1, 2, \ldots, N_{nd}$

thus, the mean call arrival rate at link k :

$$\lambda_k^h = \sum_{(s,d)\in c_k} (\alpha_{sd}^{h} \cdot L_{call}) \tag{6.1}$$

where L_{call} is the mean call length,

$c_k = \{$pairs of source and destination (s,d), s.t. $r_{sd,k} = 1, s \neq d\}$,

$r_{sd,k}$ is an element of route vector Such that

$$\text{and} \quad r_{sd,k} = \begin{cases} 1 & \text{if the route from node s to d via link k} \\ 0 & \text{otherwise} \end{cases}$$

There is a similar input traffic regulation matrix for LP packets M^l_{in} with mean packet arrival rate at link k, λ^l_k.

$$\lambda^l_k = \sum_{(s,d)\in c_k} (\alpha^l_{sd} \cdot L_{call}) \tag{6.2}$$

To obtain a balanced load on every link k, M^h_{in} and M^l_{in} have to designed in a way that $\lambda^h_k = \lambda^h$, $\lambda^l_k = \lambda^l$ for all k. If the offered loading at each link is 100%, then $\lambda^h + \lambda^l = \lambda = B$. In addition, the ratios of call arrival for different hops can be determined by M^h_{in}.

$$R_{hi} = \sum_{\text{hop distant } = \text{ i}} (\frac{\alpha^h_{sd}}{B} \cdot L_{call}) \tag{6.3}$$

6.1.3 Actual Offered Load

The actual offered load of a link is the load which can successfully impose to the link. Thus, its value is equal to the offered load λ mentioned in the previous section subtracting the loads which is blocked due to fluctuation in traffic intensity. The length of calls is geometrically distributed and the inter-arrival time of calls is also geometrically distributed with mean λ. If the integrated traffic is dominated by one type of traffic call, the situation is simplified by considering only the dominating type of call. The link can be at most occupied by a finite number of calls, say m, each with bandwidth μ. If not enough bandwidth remains, then some of the call arrivals are blocked. The situation can be approximated by M/M/m/m queueing system (Erlang Loss System) as shown in Figure 6.3. Applying Erlang B formula to calculate the proportion of time where the stationary system spend in state m, that is all the bandwidth has been occupied by m calls,

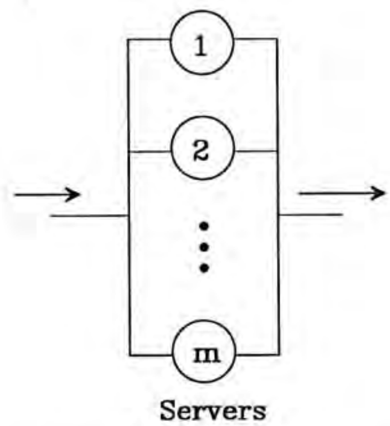


Figure 6.3. M/M/m/m Queueing System

$$P_m = \frac{\frac{1}{m!}(\frac{\lambda}{\mu})^m}{1 + \sum_{n=1}^{m} \frac{1}{n!}(\frac{\lambda}{\mu})^n} \tag{6.4}$$

where $\lambda = B = m\mu$, so

$$P_m = \frac{\frac{m^m}{m!}}{1 + \sum_{n=1}^{m} \frac{m^n}{n!}}$$

$$= \frac{\frac{m^m}{m!}}{1+m+\frac{m^2}{2!}+\ldots+\frac{m^m}{m!}}$$
$$\approx \frac{m^m}{e^m m!} \quad (6.5)$$

Finally,

$$\begin{aligned}\lambda' &= \lambda(1-P_m)\\ &= B(1-\frac{m^m}{e^m m!}) \quad (6.6)\end{aligned}$$

Numerically, the relation of the actual offered load and m is

Table 6.1

m	40	50	60	70	80	90	100	110	120
λ'	0.937B	0.944B	0.949B	0.952B	0.955B	0.958B	0.960B	0.962B	0.964B

6.1.4 Static and Dynamic Parameters

Static Network Parameters :

1. Network Configurations

   (a) No. of Nodes, $N = 7$ (Figure 6.4)

   The choice of the network topology is arbitrary but the number of nodes in the network is limited by the complexity of the network simulator. Thus, a small network with maximum possible number of hops three is chosen.

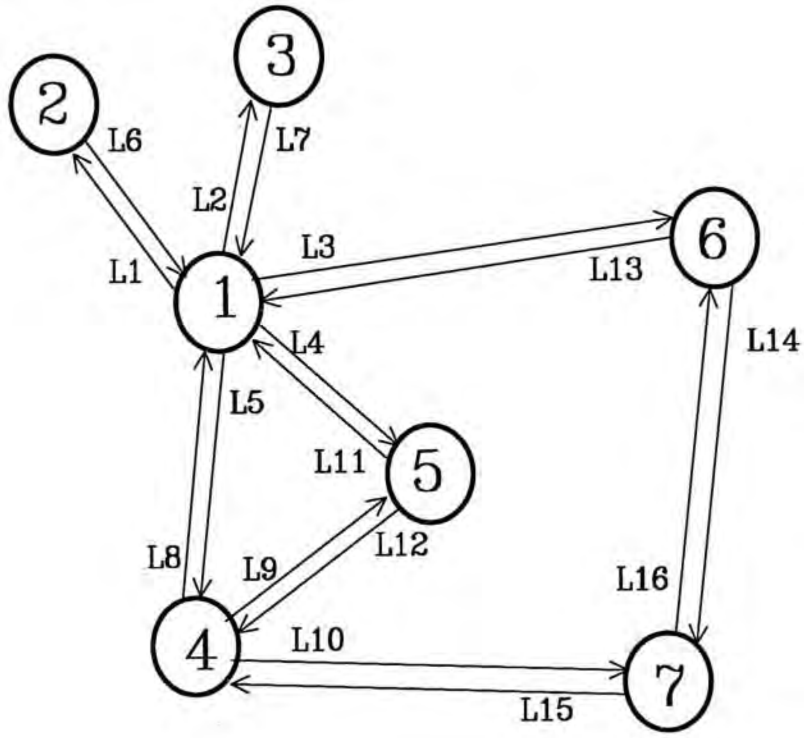


Figure 6.4 Nodes and Links of the Network

(b) No. of uni-directional links, $N_{lk} = 16$

(c) Propagation delay time matrix, M_d

$$M_d = \begin{bmatrix} \infty & 2 & 3 & 10 & 9 & 4 & \infty \\ 2 & \infty & \infty & \infty & \infty & \infty & \infty \\ 3 & \infty & \infty & \infty & \infty & \infty & \infty \\ 10 & \infty & \infty & \infty & 1 & \infty & 2 \\ 9 & \infty & \infty & 1 & \infty & \infty & \infty \\ 4 & \infty & \infty & 1 & \infty & \infty & 3 \\ \infty & \infty & \infty & 2 & \infty & 3 & \infty \end{bmatrix}$$

Every element of the above matrix represents the propagation delay time of link from the source node (row) to destination node (column).

The matrix is symmetric which implies that the incoming link and the corresponding outgoing link have the same physical length.

2. Simulation unit time = $1\mu s$

3. Length of event calendar, $L_{calen} = 125\mu s$

4. Bandwidth of each link, $B = 424$ Mbps

5. Packet Size, $N_p = 424$ bits (ATM cell)

6. Route vectors are generated by shortest path algorithm.

7. Fixed routing method is applied so no alternate routes are provided.

8. Fixed control vectors are used.

Dynamic Parameters and typical values :

In order to achieve high network throughput, the values of the dynamic parameters are obtained based on the static parameters by a series of tuning processes.

1. Length of a simulation run , (1200 millions unit time = 20 minutes of simulated time)

2. Control vector Interval δ, (12 unit time)

3. Buffer Size and Threshold Level

   (a) Scout threshold level N_T, (2 for scheme I, 8 for scheme II)

   (b) Upper threshold level for LP packet queues N_{UT}, (50)

(c) Lower threshold level for LP packet queues N_{LT}, (48)

(d) High priority buffer size, (300)

4. Duration of scout screams

(a) Period for DS packets T_{detect}, ($\sim 1000 - 2000$ unit time, for scheme I, N_{DS} for various types of call are 8,6,3,3.)

(b) Period for RS packets $T_{reserve}$, (refer to Eqn. 5.6 for scheme I, Eqn. 5.11 for scheme II)

(c) Speed-up scout rate for scheme II R_{sp}, (refer to Eqn. 5.9)

(d) Bandwidth Allowance for scheme II ϵ,

5. Traffic Loading

(a) Traffic models of HP packets, (Superposition of periodic or on-off sources, refer to Eqns (6.1) and (6.2))

(b) Maximum bit rate of type j HP call T_j, (125,72,51,37 unit time)

(c) Mean number of packets in a HP call, (500000)

(d) Input traffic regulation matrices M_{in}^h and M_{in}^l,

Traffic loading by periodic sources in the network can be represented by the following vectors :

$[T_1, T_2, T_3, T_4]$	:	Packet Interarrival time of 4 types of HP calls,
$[R_{t1}, R_{t2}, R_{t3}, R_{t4}]$	:	Ratio of call arrival rate for different call types (refer to Eqn. (6.3))
$[R_{h1}, R_{h2}, R_{h3}]$	:	Ratio of call arrival rate for different call hops

An extra vector is needed to describe the on-off sources :

$[P_{on}, P_{off}]$: Mean on-off period of source

6.2 Simulation Results

Since there are 16 links in the network (Figure 6.4), the mean link utilization evaluated in this section is the mean values of proportion of time that links are loaded with LP or HP packets. The deviation of utilization of all the links is not greater than 6%. Ideally, if full traffic loading is applied, 100% link utilization would be expected. However the actual offered loading may not be the same as the input traffic loading (refer to Section 6.1.3). Besides, bandwidth overheads due to transmission of scout packets, control vectors and algorithmic loss are imposed onto the network. The simulation results show that the scout packet overhead is less than 0.003% of the link bandwidth. Since scout packets superimpose over LP packets, if the LP packet loading is high, the actual overhead by scout packets becomes negligible. The control vector overhead is calculated as $\frac{N_{lk}}{\delta}$, where N_{lk} is the number of bits of the control vector (i.e. the number of links in the network) and δ is the control vector interval. If the size of the network is not very big, this overhead would not contribute significantly to the link utilization. Algorithmic loss is the blocking of calls due to response delay of control vectors. This loss can be obtained by the simulation results in this section.

The performance of scheme I and II under different traffic heterogeneities, loading and burstiness is examined. The following results are based on the static and dynamic parameters mentioned in the Section 6.1.4.

Table 6.1 shows the mean link utilization of scheme I and II under different homogeneous traffic inputs. Because of the *bin-packing effect*, the algorithm discriminates against setting up calls of large bandwidth during busy time. Thus, the link utilization is highest if all the traffic sources are of Type 1 (i.e. the smallest bandwidth call). Both schemes can obtain high network throughput but it is found that scheme I performs slightly better than scheme II due to the strong periodicity of the homogeneous traffic.

Traffic Parameters for Table 6.1 :

1. Traffic Source Model : Periodic sources
2. Call Hop Ratio : $[R_{h1}, R_{h2}, R_{h3}] = [0.800, 0.169, 0.031]$
3. Mean Offered Loaded : 100% loading by HP calls (i.e. $\lambda^h = B$)

Table 6.1

Call Type Ratio	Mean Link Utilization	
$[R_{t1}, R_{t2}, R_{t3}, R_{t4}]$	Scheme I	Scheme II
$[1,0,0,0]$	0.9215	0.9163
$[0,1,0,0]$	0.9084	0.8972
$[0,0,1,0]$	0.8837	0.8779
$[0,0,0,1]$	0.8755	0.8514

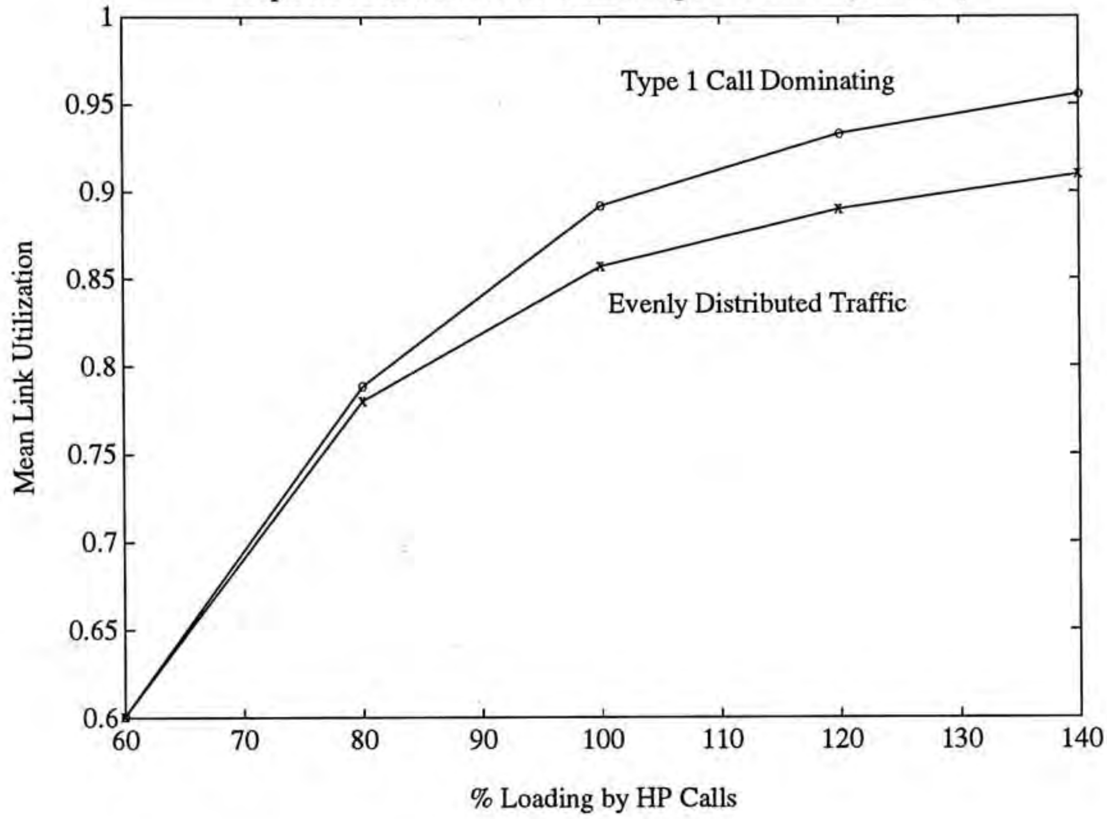
Network performance under two traffic patterns that are composed of different call type ratios R_{ti} are evaluated in Graphs 6.1-2. Type 1 call is the dominating call in the first traffic pattern and the arrival rate of various types of call are equally probable in the second traffic patterns. It is found that when the HP call loading is low, the network performs similarly under different traffic of call type ratios. But as the loading increases, higher link utilization can be achieved under Type 1 call dominating traffic. In real-life situation, it is more probable that the traffic of an integrated services network is dominated by the calls with smaller bandwidth such as the voice-based calls.

The blocking probabilities of HP calls of various call type for scheme I & II can be compared in Table 6.2-3. The blocking probability is the proportion of calls that are refused from admission to the total number of calls sent in the same type. It is observed that the calls with larger bandwidth are more vulnerable to admission refusal. Comparing the results of scheme I and II, scheme II favours larger bandwidth call and scheme I favours smaller bandwidth calls.

Traffic Parameters for Graphs 6.1-2 and Tables 6.2-3 :

1. Traffic Source Model : Periodic sources
2. Call Hop Ratio : $[R_{h1}, R_{h2}, R_{h3}] = [0.800, 0.169, 0.031]$
3. Dominated by Type 1 call : $[R_{t1}, R_{t2}, R_{t3}, R_{t4}] = [0.781, 0.156, 0.047, 0.016]$
4. Evenly distributed traffic : $[R_{t1}, R_{t2}, R_{t3}, R_{t4}] = [0.25, 0.25, 0.25, 0.25]$

Graph 6.1 : Performance under Inhomogenous Traffic (Scheme I)



Graph 6.2 : Performance under Inhomogenous Traffic (Scheme II)

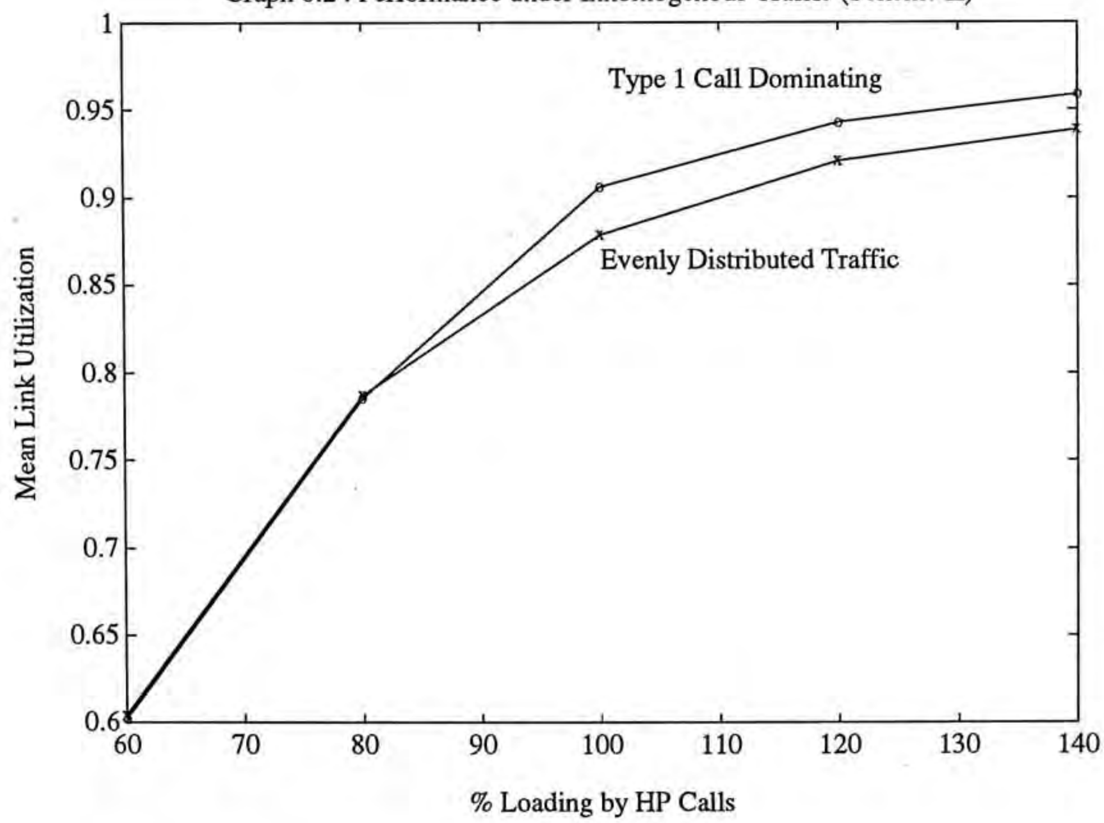


Table 6.2 : Blocking probabilities of HP calls for various call types under Type 1 call dominating traffic

HP Call Loading (%)	Scheme I Call Type				Scheme II Call Type			
	1	2	3	4	1	2	3	4
60	0.0002	0.0008	0.0013	0.0038	0.0000	0.0000	0.0000	0.0000
80	0.0327	0.0506	0.1324	0.2022	0.0050	0.0080	0.0102	0.0258
100	0.0239	0.3043	0.4160	0.6332	0.0687	0.1256	0.1793	0.2296
120	0.0934	0.5902	0.6462	0.8000	0.1479	0.2320	0.3362	0.4642
140	0.1661	0.7539	0.8016	0.9075	0.2342	0.3786	0.5016	0.6057

Table 6.3 : Blocking probabilities of HP calls for various call types under Evenly Distributed Traffic

HP Call Loading (%)	Scheme I Call Type				Scheme II Call Type			
	1	2	3	4	1	2	3	4
60	0.0000	0.0002	0.0005	0.0033	0.0000	0.0000	0.0000	0.0000
80	0.0020	0.0172	0.0426	0.1026	0.0090	0.0153	0.0246	0.0423
100	0.0120	0.0903	0.1534	0.3104	0.0425	0.0803	0.1198	0.1688
120	0.0303	0.1847	0.2700	0.4552	0.0885	0.1819	0.2351	0.3225
140	0.0337	0.2991	0.3990	0.5989	0.1418	0.2409	0.3371	0.4458

Graphs 6.3-4 depict the effect of call hop ratio R_{hi}. Traffic Pattern 1 is dominated by single-hop calls and the call hop ratios are more evenly distributed

in Traffic Pattern 2. It is found that the performance of the network degrades in both schemes if the number of hops of the calls increases. It is because blocking of multi-hop calls have greater impact on network performance than that of single-hop calls. In addition, Tables 6.4-5 show that the algorithm biased toward setting up calls with smaller hops especially when the traffic loading is heavy. Thus, the single-hop calls are mostly likely to be set up for congested traffic.

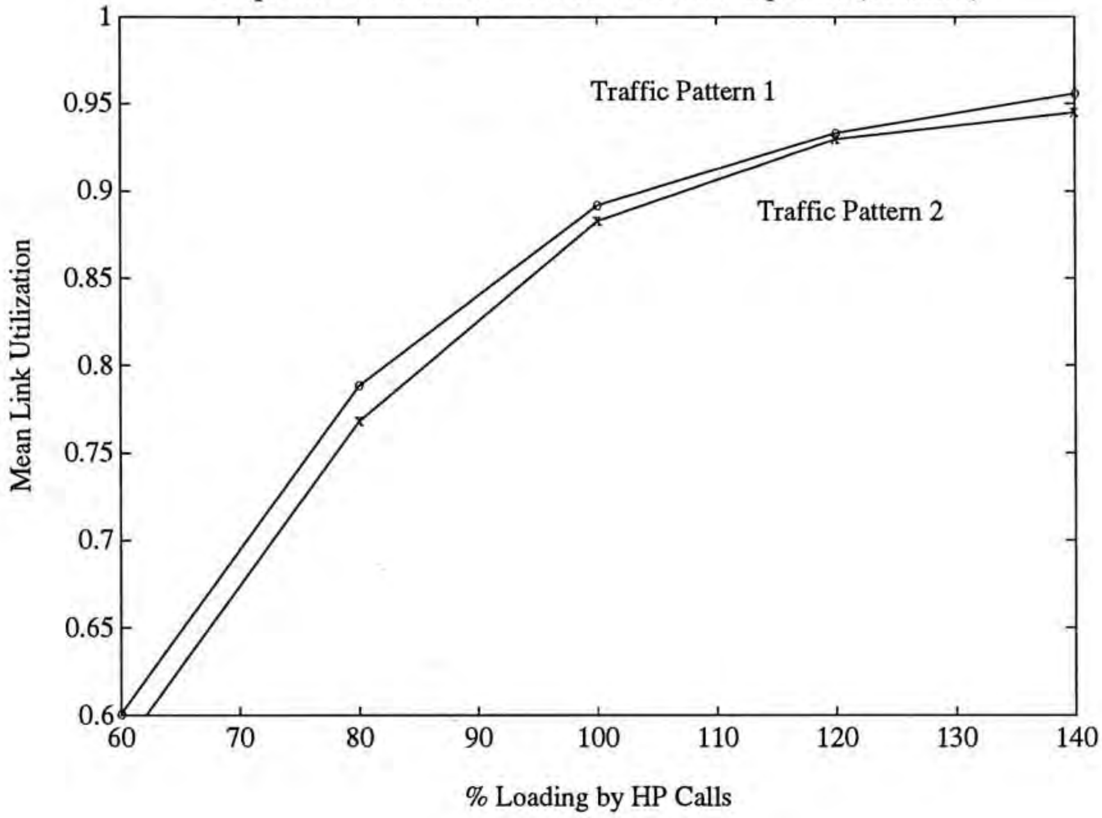
Traffic Parameters for Graphs 6.3-4 and Tables 6.4-5 :

1. Traffic Source Model : Periodic sources
2. Call type ratio : $[R_{t1}, R_{t2}, R_{t3}, R_{t4}] = [0.781, 0.156, 0.047, 0.016]$
3. Traffic Pattern 1 : $[R_{h1}, R_{h2}, R_{h3}] = [0.800, 0.169, 0.031]$
4. Traffic Pattern 2 : $[R_{h1}, R_{h2}, R_{h3}] = [0.529, 0.373, 0.098]$

Table 6.4 : Blocking probabilities of HP calls for various call hop types under Traffic Pattern 1

HP Call Loading (%)	Scheme I Hop Type			Scheme II Hop Type		
	1	2	3	1	2	3
60	0.0004	0.0004	0.0000	0.0000	0.0000	0.0000
80	0.0429	0.0444	0.0379	0.0045	0.0114	0.0186
100	0.1213	0.2089	0.2528	0.0642	0.1569	0.2375
120	0.1810	0.3078	0.3408	0.1334	0.3187	0.4626
140	0.2683	0.4105	0.4845	0.2167	0.4797	0.6939

Graph 6.3 : Performance under Different Call Hop Ratio (Scheme I)



Graph 6.4 : Performance under Different Call Hop Ratio (Scheme II)

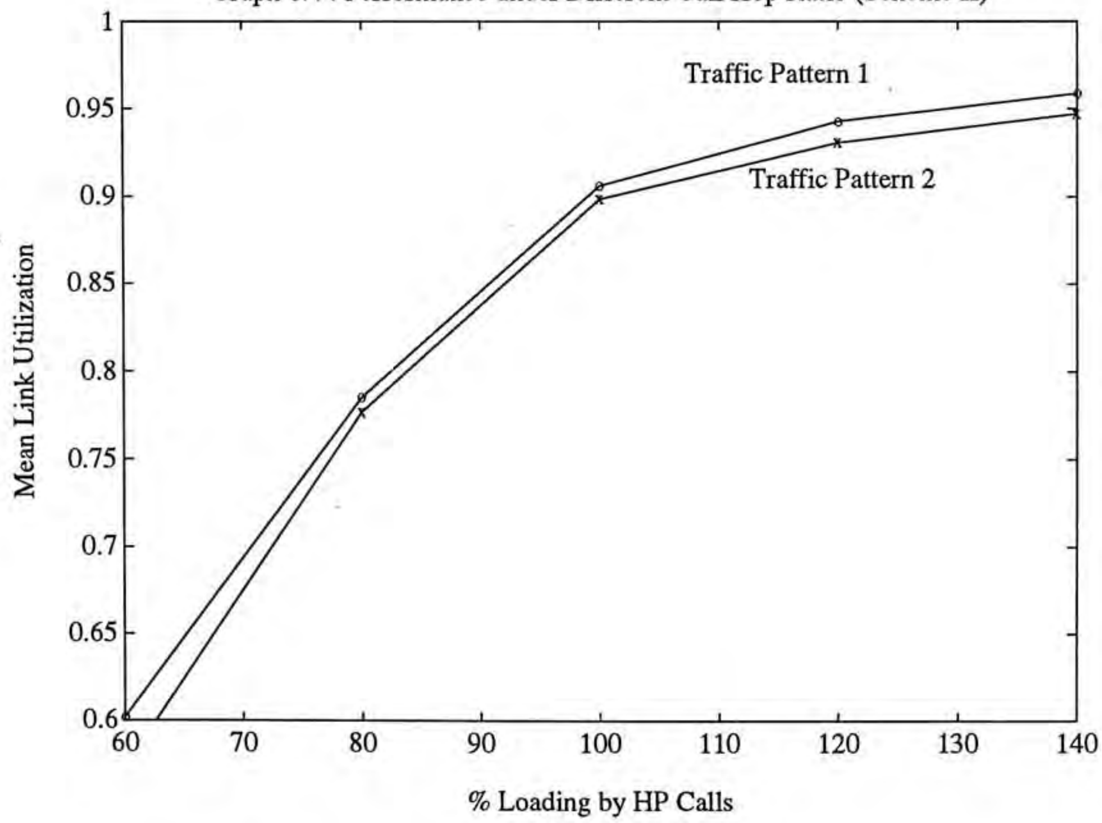


Table 6.5 : Blocking probabilities of HP calls for various call hop types under Traffic Pattern 2

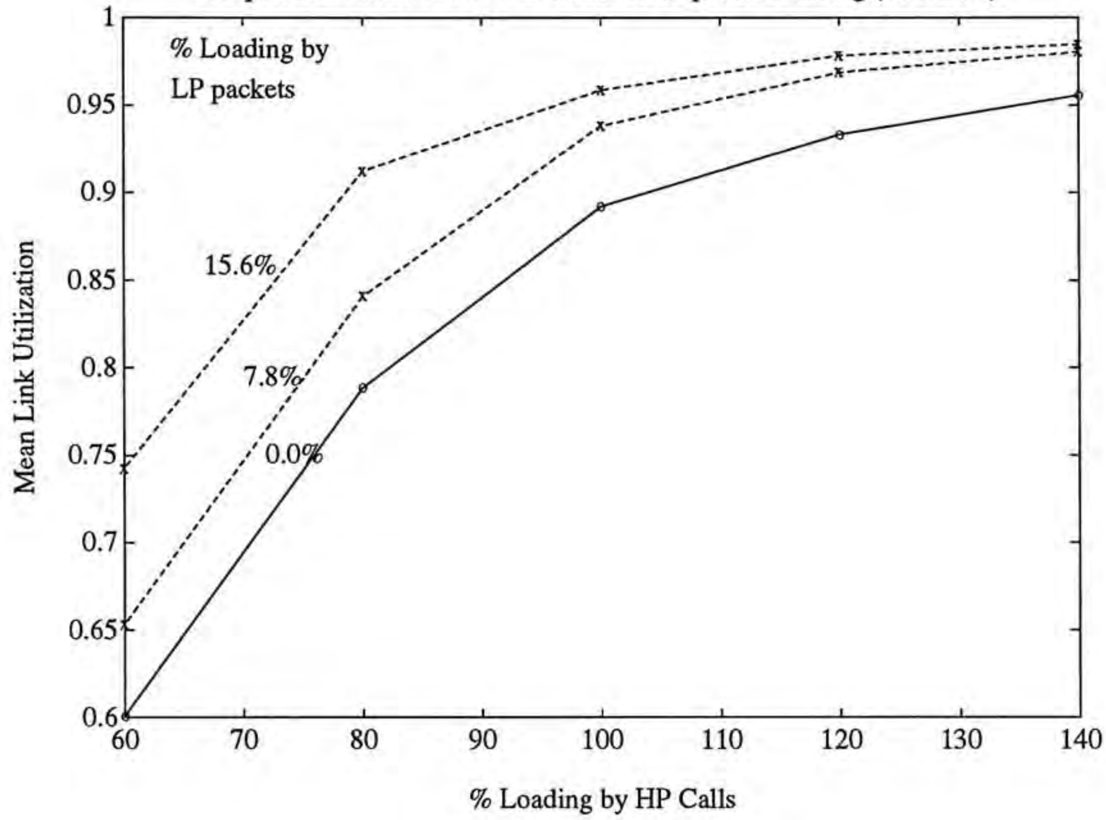
HP Call Loading (%)	Scheme I Hop Type			Scheme II Hop Type		
	1	2	3	1	2	3
60	0.0000	0.0000	0.0009	0.0000	0.0000	0.0000
80	0.0000	0.0026	0.0000	0.0013	0.0067	0.0042
100	0.0677	0.1222	0.1365	0.0038	0.0185	0.0204
120	0.1676	0.2741	0.3135	0.1168	0.2565	0.3022
140	0.2528	0.3496	0.3863	0.1919	0.3617	0.4626

From Graphs 6.5.1-2, the network performance of different loading by HP calls and LP packets can be examined. If the loading of LP packets is fixed, the performance levels off at heavy HP call loading. The mean link utilization can rise up to more than 97% in both schemes. On one hand, LP packet loading can occupy more bandwidth by scheme II than scheme I (Graphs 6.5.3-4). On the other hand, congestion due to LP packet transmission affects the admission of HP calls and this degrading effect is heavier in scheme II.

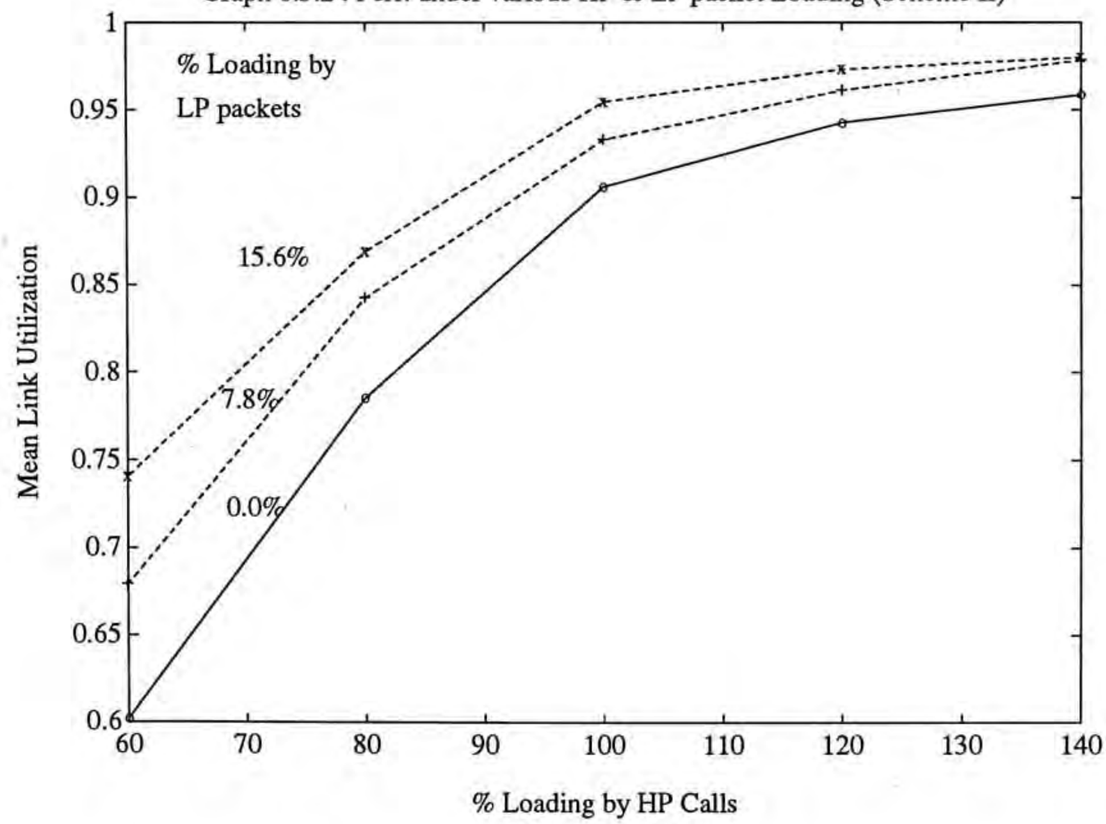
Traffic Parameters for Graphs 6.5.1-4 :

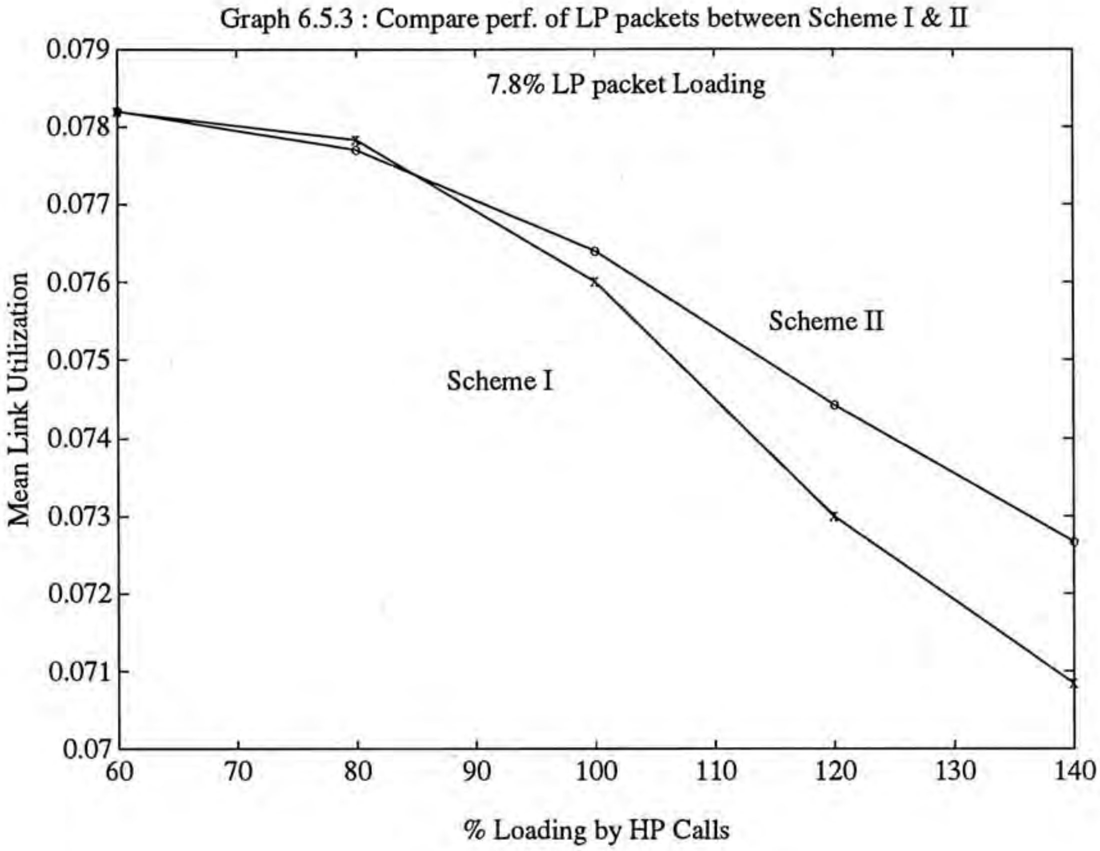
1. Traffic Source Model : Periodic sources
2. Call Hop Ratio : $[R_{h1}, R_{h2}, R_{h3}] = [0.800, 0.169, 0.031]$
3. Call Type Ratio : $[R_{t1}, R_{t2}, R_{t3}, R_{t4}] = [0.781, 0.156, 0.047, 0.016]$

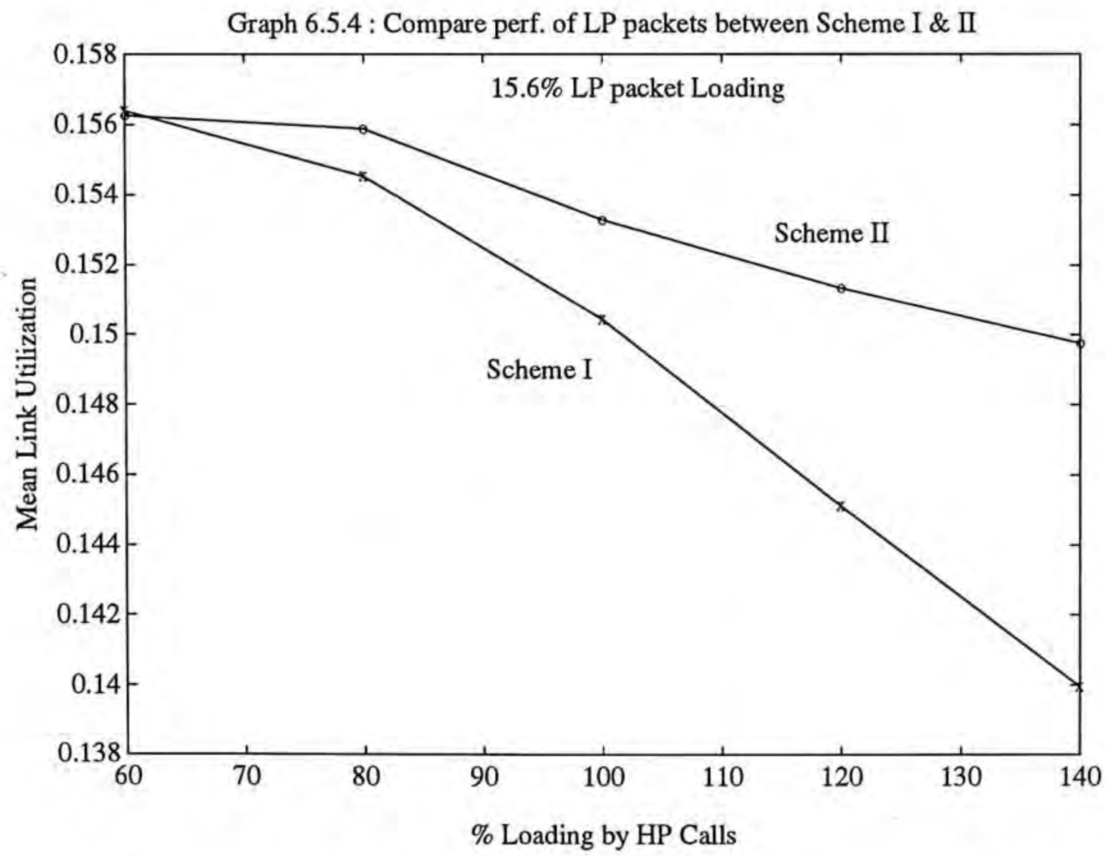
Graph 6.5.1 : Perf. under various HP & LP packet Loading (Scheme I)



Graph 6.5.2 : Perf. under various HP & LP packet Loading (Scheme II)




Graph 6.5.3 : Compare perf. of LP packets between Scheme I & II
7.8% LP packet Loading
Scheme II
Scheme I
Mean Link Utilization
0.079
0.078
0.077
0.076
0.075
0.074
0.073
0.072
0.071
0.07
60
70
80
90
100
110
120
130
140
% Loading by HP Calls


Graph 6.5.4 : Compare perf. of LP packets between Scheme I & II
15.6% LP packet Loading
Scheme II
Scheme I
Mean Link Utilization
0.158
0.156
0.154
0.152
0.15
0.148
0.146
0.144
0.142
0.14
0.138
60
70
80
90
100
110
120
130
140
% Loading by HP Calls

For scheme I, as stated in Section 5.1, the overrun probability under PTM can be altered by N_{DS}. From Table 6.6, N_{DS} for different call types are adjusted to investigate its relation with the mean link utilization and packet loss probability. Therefore, the values of N_{DS} for various call types can be increased to satisfy the requirement of packet loss probability. Scheme II outperforms scheme I for the traffic due to superposition of periodic sources. The packet loss probability of scheme II is less than 10^{-9} and the maximum HP buffer size requirement is only 25.

Traffic Parameters for Table 6.6 :

1. Traffic Source Model : Periodic sources

2. Call Hop Ratio : $[R_{h1}, R_{h2}, R_{h3}] = [0.800, 0.169, 0.031]$

3. Call Type Ratio : $[R_{t1}, R_{t2}, R_{t3}, R_{t4}] = [0.781, 0.156, 0.047, 0.016]$

4. 100% HP Call Loading

5. HP buffer size = 300

Table 6.6 : Packet loss Probabilities of Scheme I
using various number of DS packets

N_{DS} for call type				Pack. Loss Prob.	Mean Link Utilization (%)
1	2	3	4		
7	5	3	3	1.312xE-7	89.28
6	4	3	3	5.458xE-7	89.74
5	3	3	3	6.894xE-7	89.89
4	3	3	3	8.849xE-7	90.06
3	3	3	3	1.446xE-6	90.14

If on-off sources are taken as the traffic sources, the relation between packet loss probability and mean off-period is shown in Graph 6.6. From the results, the mean link utilization for scheme I and II are 88% and 90% respectively. Although scheme II performs better than scheme I, as the traffic burstiness increases, the packet loss probability increase rapidly. For the same situation, the packet loss probability of scheme I increases less rapidly than that of scheme II.

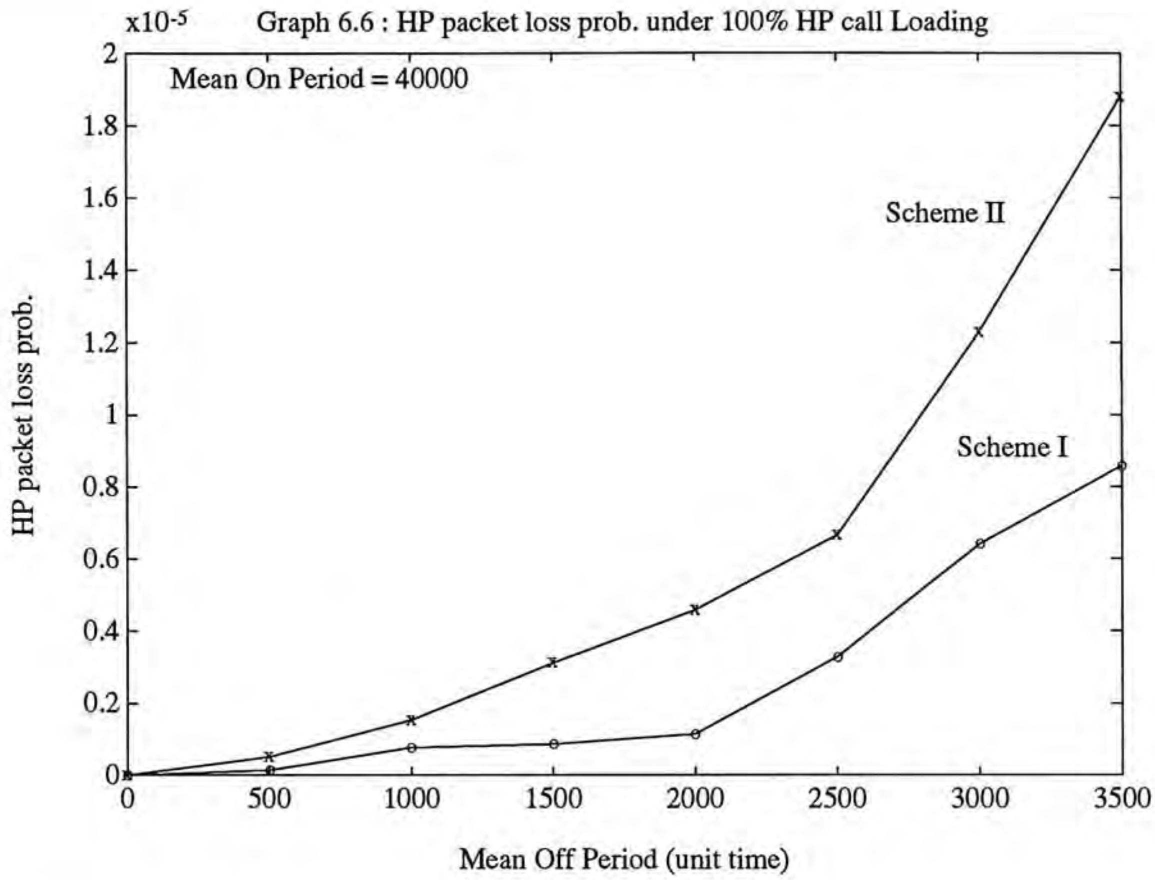
Graphs 6.7-8 portray the HP packet loss probability under traffic loading of different proportion of periodic sources and on-off Sources. Acceptable packet loss probability can be obtained by increasing the ratio of bandwidth allowance to link bandwidth of scheme II, i.e. $\frac{\epsilon}{B}$. However, the performance will be slightly affected (Graphs 6.9-10). Lastly, if the ratio of call to link bandwidth decreases, the aggregate traffic burstiness will be lowered and a better performance of the call set-up algorithm is expected.

Traffic Parameters for Graph 6.6 :

1. Traffic Source Model : On-off Sources
2. Call Hop Ratio : $[R_{h1}, R_{h2}, R_{h3}] = [0.800, 0.169, 0.031]$
3. Call Type Ratio : $[R_{t1}, R_{t2}, R_{t3}, R_{t4}] = [0.781, 0.156, 0.047, 0.016]$
4. Mean on and off period : $[P_{on}, P_{off}] = [40000, 500 - 3500]$
5. 100% HP call Loading
6. HP buffer size = 300
7. ratio of bandwidth allowance to link bandwidth for scheme II, $\frac{\epsilon}{B} = 0.005$

Traffic Parameters for Graphs 6.7-8 :

1. Traffic Source Model : Periodic sources + on-off Sources
2. Call Hop Ratio : $[R_{h1}, R_{h2}, R_{h3}] = [0.800, 0.169, 0.031]$
3. Call Type Ratio : $[R_{t1}, R_{t2}, R_{t3}, R_{t4}] = [0.781, 0.156, 0.047, 0.016]$
4. Mean on and off period : $[P_{on}, P_{off}] = [40000, 40000]$
5. 80% and 90% Traffic Loading
6. HP buffer size = 300
7. ratio of bandwidth allowance to link bandwidth for scheme II, $\frac{\epsilon}{B} = 0.005$



Traffic Parameters for Graphs 6.9-10 :

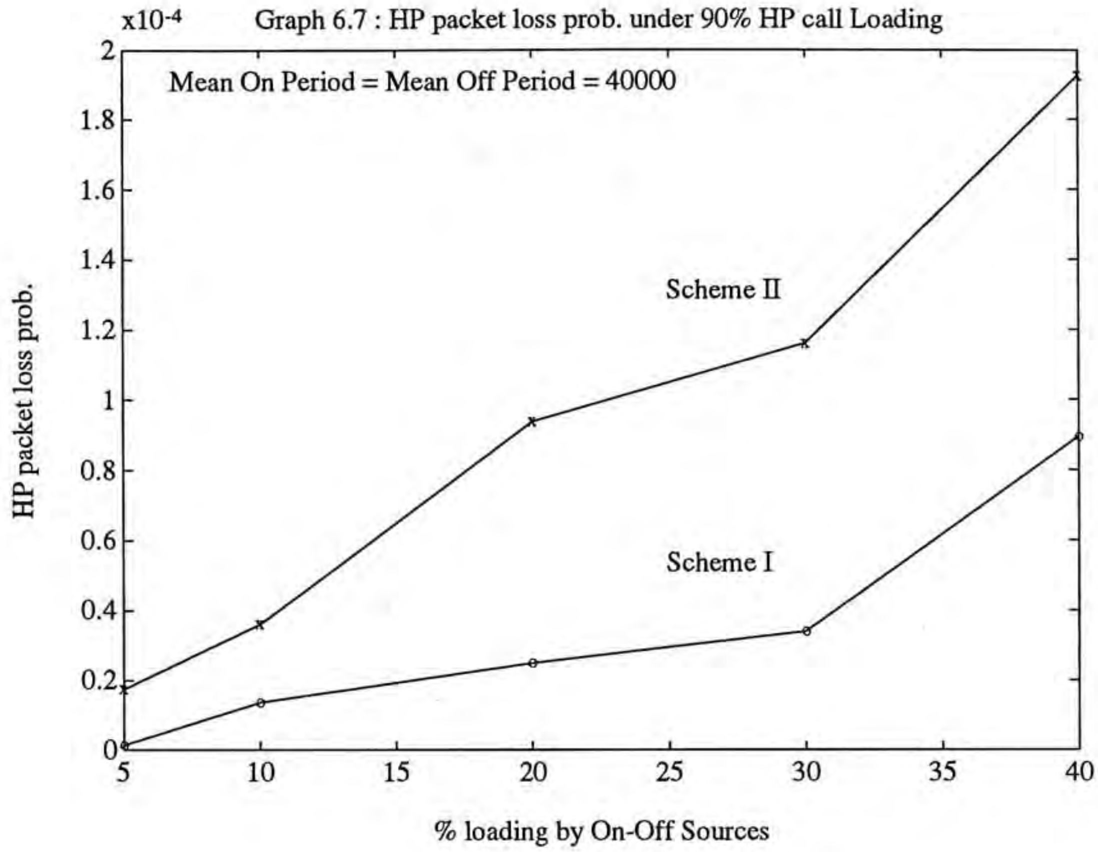
1. Traffic Source Model : 90% periodic sources + 10% on-off sources (90% Traffic Loading)

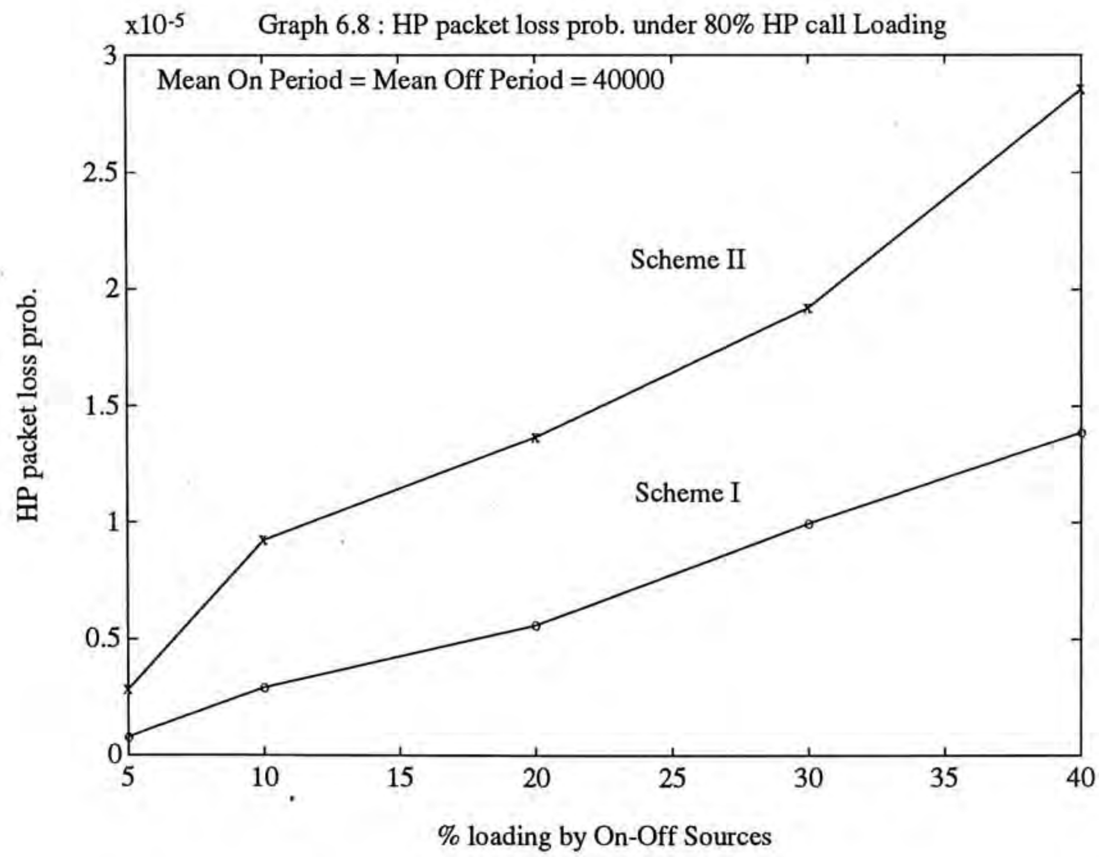
2. Call Hop Ratio : $[R_{h1}, R_{h2}, R_{h3}] = [0.800, 0.169, 0.031]$

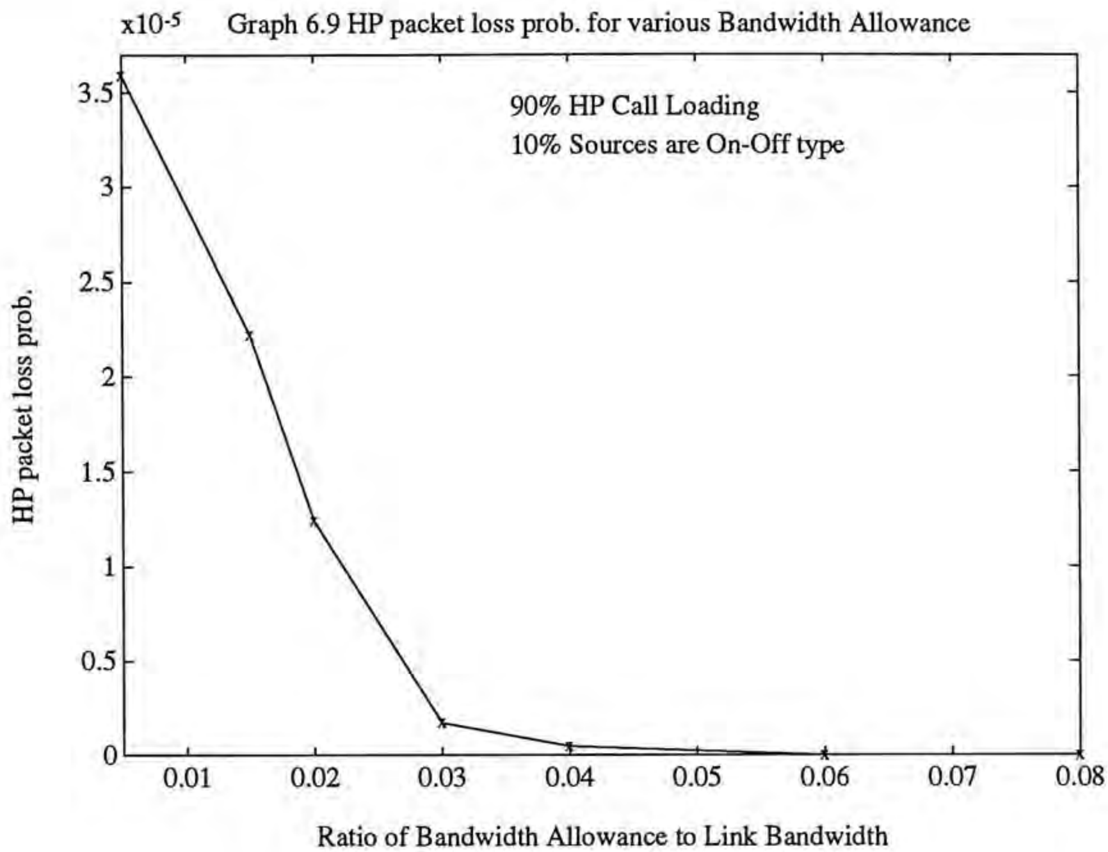
3. Call Type Ratio : $[R_{t1}, R_{t2}, R_{t3}, R_{t4}] = [0.781, 0.156, 0.047, 0.016]$

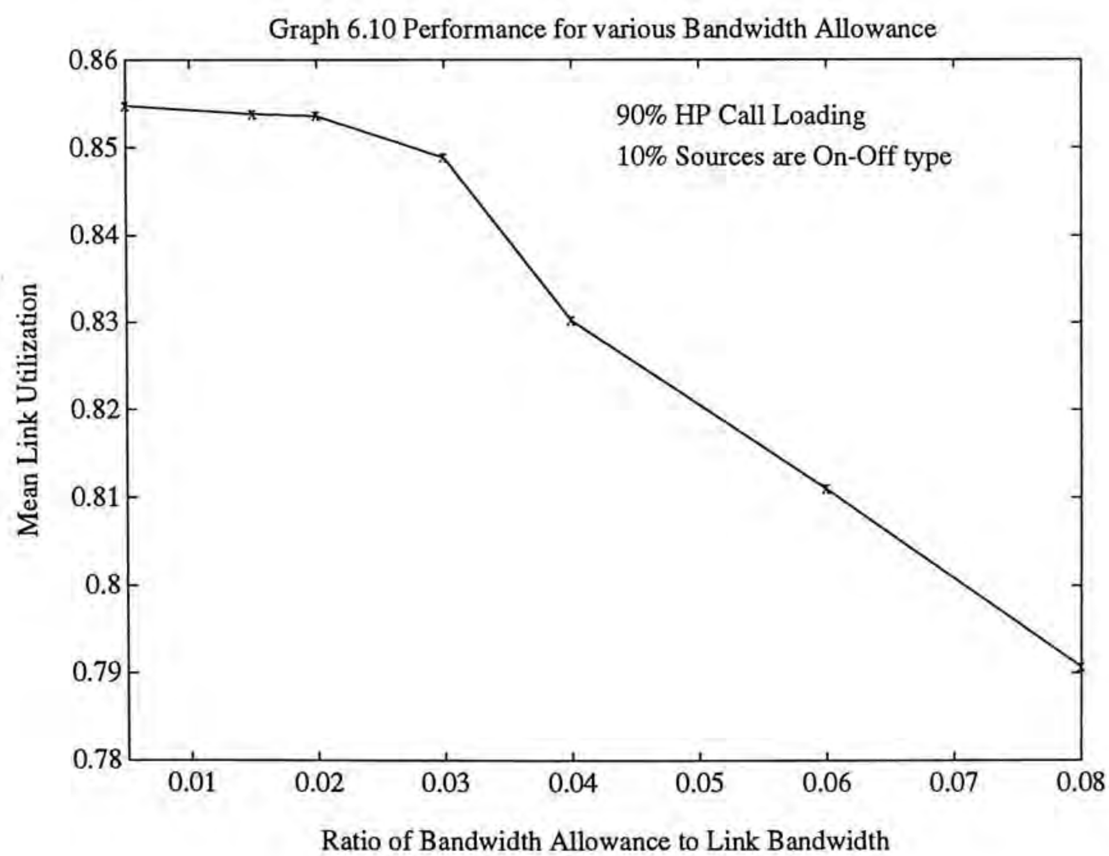
4. Mean on and off period : $[P_{on}, P_{off}] = [40000, 40000]$

5. HP buffer size = 300


x10-4
Graph 6.7 : HP packet loss prob. under 90% HP call Loading
Mean On Period = Mean Off Period = 40000
Scheme II
Scheme I
HP packet loss prob.
% loading by On-Off Sources
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
5
10
15
20
25
30
35
40


x10-5
Graph 6.8 : HP packet loss prob. under 80% HP call Loading
Mean On Period = Mean Off Period = 40000
Scheme II
Scheme I
HP packet loss prob.
% loading by On-Off Sources
0
0.5
1
1.5
2
2.5
3
5
10
15
20
25
30
35
40


x10-5
Graph 6.9 HP packet loss prob. for various Bandwidth Allowance
90% HP Call Loading
10% Sources are On-Off type
HP packet loss prob.
3.5
3
2.5
2
1.5
1
0.5
0
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08
Ratio of Bandwidth Allowance to Link Bandwidth


Graph 6.10 Performance for various Bandwidth Allowance
90% HP Call Loading
10% Sources are On-Off type
Mean Link Utilization
0.86
0.85
0.84
0.83
0.82
0.81
0.8
0.79
0.78
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08
Ratio of Bandwidth Allowance to Link Bandwidth

Chapter 7

Conclusions

Distributed algorithms for traffic control and call set-up for broadband ISDN are designed in detail. These algorithms can be realized by simple hardware in which only primitive operations like READ from high speed memory, logical AND/OR are performed. The properties of being starvation-free and deadlock-free of the traffic control are presented. In setting up an one way multi-hop call, only the call originating node involves. No inter-node communication and special mechanism for tearing down the call are required.

Parameters in the call set-up algorithm are fine tuned with respect to various traffic models of integrated services. A network simulator for the broadband network of a general topology has been designed and implemented to investigate the performance of the traffic control and call set-up algorithms. Network performance under different traffic heterogeneities, loading and burstiness is examined. The results have been shown that high throughput and negligible packet loss probability can be achieved for different patterns of aggregate traffic due to superposition of periodic sources. In addition, if the traffic is mixed with on-off sources which will increase the traffic burstiness, the requirement of packet loss probability can be satisfied by adjusting parameters of the algorithms and trading off certain levels of bandwidth utilization.

There are a few directions for future work. First is to investigate the network performance by simulation under other traffic models with different traffic properties and levels of burstiness. The second is to expand the size of the testing network for simulation. The third is to design the call set-up algorithm for two way communication calls and analyse its performance.

Bibliography

[AHM 88] Ahmadi H., et al., "A high-performance Switch fabric for integrated circuit and packet switching", Proc. INFOCOM 88, New Orleans, pp.9-18.

[AKA 90] Akata M., et al., "A 250-Mb/s 32x32 CMOS Crosspoint LSI for ATM Switching Systems," IEEE Journal of Solid-State circuit, Vol.25, pp.1433-1439, No.6 Dec 1990.

[ARA 90] Arakawa N., et al., "ATM Switch for Multi-Media Switching System", International Switching Symposium 90, A7.2, May 1990.

[BAT 68] Batcher K.E., "Sorting networks and their applications", Proc. 1968 Spring Joint Computer Conf.

[CID 88] Cidon I., et al., "PARIS: An Approach to Integrated High-speed Private Networks," International J. of Digital and Analog Cabled Systems, Vol. 1, 1988, pp.77-85.

[CLO 91] Cloetens L., et al., "A 600Mb/s 16x16 Switching Element Chipset for Broadband ISDN," IEEE International Solid-State Circuits Conference 1991 pp.240-241.

[DEC 90] Decina M., et al., "Bandwidth Assignment and Virtual Call Blocking in ATM Networks", IEEE INFOCOM'90, pp.881-888.

[DEV 88] Devault M., "The Prelude ATD experiment: assessments and future prospects", IEEE J. Selected Areas in Comm. Vol.6, no. 9, pp.1528-1537, Dec. 1988.

[DIT 88] Dittmann L., et al., "Statistical Multiplexing of Identical Bursty Sources in an ATM Network", IEEE Globecomm 1988 pp. 39.6.1-39.6.5.

[FUC 70] Fuchs E., et al., "Estimates of Distributions of Random Var iables for Certain Computer Communications Traffic Models", Comm. of the ACM, Vol 13, No. 12, Dec. 1970.

[GAL 89] Gallassi G., et al., "ATM: Bandwidth Assignment and Enforcement Policies", IEEE Globecom '89, pp. 1788-1793, 1989.

[GIE 81] Giessler, A et al., "Flow Control Based on Buffer Classes," IEEE Trans. on Comm., Vol. COM-27, No.4, pp 436-443, April 1981.

[GUN 81] Gunther K.D., "Prevention of Deadlocks in Packet-Switched Data Transport Systems", IEEE Tans. on Comm., Vol. COM-29, No. 4, April 1981.

[HAJ 88] Hajikano K., et al., "Asychronous Transfer Mode switching Architecture for Broadband ISDN - Multi-Stage Self-Routing (MSSR)", ICC '88 29.3 (June 1988).

[HI1 89] Hiramatsu A., "A self-learning control of ATM service quality using neural network", Tans. IEICE, J72-B-I, pp564-573, 1989.

[HI2 89] Hirano M., et al., "Characteristics of a cell multiplexer for bursty ATM traffic", Proc. IEEE ICC '89, pp.13.2.1-13.2.5.

[HUA 84] Huang J., "Starlite : A Wideband Digital Switch", Globecom, '84, 5.3 pp.121-125 (Nov. 1984).

[HUI 87] Hui J.Y., et al., "A broadband packet switch for integrated transport", IEEE J. Select. Areas Comm., Vol.SAC-5, pp.1264-1273, Oct. 1987.

[HYM 91] Hyman J.M., et al., "Real-time Scheduling with Quality of Service Constraints", IEEE J. Select. Areas in Comm., Nov. 1991.

[INT 88] International Telegraph and Telephone Consultative Committee, "Broadband Aspects of ISDN," CCITT Recommendation I.121, 1988.

[JAI 91] Jaime J.B. and Tatsuya S., "Survey of Traffic Control Schemes and Protocols in ATM Network," Proceedings of the IEEE, Vol. 79, No. 2, Feb. 1991.

[JOO 89] Joos P., et al., "A Statistical Bandwidth Allocation and Usage Monitoring Algorithm for ATM Networks", IEEE ICC'89, pp.13.5.1-8.

[KAR 87] Karol M.J., et al., "Input Verus Output Queueing on a Space - Division Packet Switch", IEEE Trans. on Comm., Vol. COM-35, No.12, Dec. 1987.

[KES 91] Keshav S., "A Control-Theoretic Approach to Flow Control", ACM Sigcomm, Sept. 1991, pp.3-15.

[KER 79] Kermani P., et al, "Virtual Cut-through : A new computer Communication Switching Technique", Computer Networks, Vol.3, pp.267-286, 1979.

[KLE 76] Kleinrock, L., "Queueing Systems, Volume 2: Computer Applications," Wiley-Interscience, New York, 1976.

[KRA 85] Kraimeche B., et al., "Analysis of Traffic Access Control Strategies in Integrated Service Networks", IEEE Trans. on Comm., Vol. COM-33, No. 10, Oct. 1985, pp.1085-1093.

[KUL 84] Kulzer J., et al., "Statistical switching architecture for future services", Proc. ISS '84, pp.43A.1.1-43A.1.6.

[KUW 89] Kuwahara H., et al., "A Shared Buffer Memory Switch for an ATM Exchange", ICC '89 pp.118-122, June 1989.

[LAU 90] Lau P.S.Y., et al., "Design and Performance analysis of a b uffer subsystem for the Batcher-Banyan Switch," GLOBECOM'90 905.3.1-905.3.5.

[LAZ 83] Lazar, A.A., "Optimal Flow Control of a class of Queueing Networks in Equilibrium," IEEE Trans. on Automatic Control, Vol. AC-28, No. 8, pp 1001-1007, Aug. 1983.

[LAZ 90] Lazar A.A., et al., "Real-Time Traffic Measurements on MAG NET II", IEEE J. Selected Areas in Comm. Vol. 8. No. 3., April 1990,467-483.

[LES 91] Leslie I., et al., "Fairisle: An ATM Network for the Local Area", ACM Sigcomm 91, Sept. 1991, pp.327-336.

[LI 90] Li, S.-Y. R., "Algorithms for Flow Control and Call Set-up in Multi-Hop Broadband ISDN," INFOCOM '90, June 1990.

[LI2 90] Li, S.-Y. R., "Considerations of Periodic Traffic in Packet Switching," IEEE Trans. on Comm., Vol. 38, No.9, pp 1625-1629, September 1990.

[LI 92] Li, S.-Y. R., "Traffic Flow Control and Call Set-up in Multi-hop Broadband Networks", United States Patent, Patent No.5, 128, 932, July 1992.

[MAR 90] Marcus W.S., "A CMOS Batcher and Banyan Chip Set for B-ISD N Packet Switching", IEEE J. Solid-Sate Circuit, Vol.25, no.6, Dec. 1990.

[MAT 81] Matsumoto, J. and Hiromichi, M, "Flow Control in Packet-Switched Networks by Gradual Restrictions of Virtual Calls," IEEE Trans. on Comm., Vol. 29, No.4, April 1981.

[NAK 89] Nakamaki K., et al., "Traffic Control for ATM Networks", IEEE ICC'89 June, pp.713-717.

[NEW 88] Newman P., "A Fast Packet Switch for the Integrated Services Backbone Network", IEEE J. on Selected Areas in Comm., Vol. 6, No. 9, pp.1468-1479, Dec. 1988.

[NOJ 87] Nojima S., et al., "Integrated services packet network using bus matrix switch", IEEE J. Select. Areas in Communications Vol.5, pp.1284-1292, Oct. 1987.

[PEN 75] Pennotti, M.C. and Schwartz, M., "Congestion Control in Store and Forward Tandem Links," IEEE Trans. on Comm., Vol. COM-23, No.12, pp 1434-1443, December 1975.

[RO1 89] Ross K.W., et al., "Optimal Circuit Access Policies in an ISDN Environment: A Markov Decision Approach", IEEE Trans. on Comm., Vol. 37, No.9, pp.934-939, Sept. 1989.

[RO2 89] Ross K.W., et al., "The Stochastic Knapsack Problem", IEEE Trans. on Comm., Vol. 37, No.7, pp.740-747, July 1989.

[SAI 91] Saito H., et al., "Traffic Control Technologies in ATM Networks", IEICE Trans., Vol. E 74, No.4, pp.761-771, April 1991.

[SCH 88] Schoute F.C., "Simple decision rules for acceptance of mixed traffic streams", Philips TDS Review, 46, 2, pp.35-48, 1988.

[SPR 81] Sproule, D.E. and Mellor F., "Routing, Flow, and Congestion Control in the Datapac Network," IEEE Trans. on Comm., Vol. COM-27, No.4, pp 386-391, April 1981.

[SRI 86] Sriram K., et al., "Characterizing Superposition Arrival Process in Packet Multiplexers for Voice and Data", IEEE J. Select. Areas Comm. Vol.SAC-4, pp.833-846, Sep. 1986.

[SUZ 89] Suzuki H, et al., "Output Buffer Switch Architecture for A synchronous Transfer Mode", International Conference on Communications (ICC) 89 4.1 (June 1989)

[SUZ 90] Suzuki H, et al., "A Bursty Traffic Control Strategy for ATM Networks", IEEE ICC'90, pp.505.6.1-5.

[TAN 88] Tanenbaum A.S.,"Computer Networks", Second Ed., Prentice-Hall, Englewood Cliffs, N.J., 1988.

[TAN 91] Tanaka S., et al., "A 400Mb/s 8x8 BiCMOS ATM LSI with 128k b On-Chip Shared Memory", IEEE International Solid-state Circuit Conference, 1991.

[TOB 91] Tobagi F.A., Kwok T.C., "The Tandem Banyan Switching Fabric: A Simple High-Performance Fast Packet Switch," INFOCOM.91, pp.1245-1253.

[TYM 81] Tymes, L., "Routing and Flow Control in TYMNET," IEEE Trans. on Comm., Vol. COM-27, No. 4, pp 693-702, April 1981.

[YEH 87] Yeh Y.-S., et al., "The Knockout switch: a simple, modular architecture for high performance packet switching", IEEE J. Selected Areas in Comm . Vol.5, no. 8, pp.1274-1283, Oct. 1987.

Appendix A

List of Symbols

Symbols	**Description**
Q_{sd}	low priority queue of route from node s to d
δ	control cector interval
B_{mean}	mean bit rate of the traffic mix
T_i	packet interarrival time of call type i
N_i	mean number of sources of call type i
N_{UT}	upper threshold level for low priority queues
N_{LT}	lower threshold level for low priority queues
N_T	threshold level for detecting scout packet count
N_{DS}	number of detecting scout packets of the intended call
N_{RS}	number of reserving scout packets of the intended call
N_s	sum of N_{DS} and N_{RS}
N_p	packet size
N_{nd}	number of nodes in the network
$N(t)$	counting process associated with the arrival process of packets
N_{lk}	number of links in the network
$P_{on,i}$	mean on-period of call type i
$P_{off,i}$	mean off-period of call type i
$I(t)$	index of dispersion for counts at time t
$C_{s,t}$	control vector of node s at time t
S_k	link status of link k
$L_{s,t}$	link status vector of the outgoing links of node s at time t

Symbols	Description
l_k	link status of link k
R_{sd}	route vector from node s to d
$r_{sd,k}$	route status from node s to d includes link k
$M_{i,j}$	vector mask of the incoming link from node i to j
$C^p_{0k,t}$	incoming variable-length control vector from node 0 to k at time t
$\oplus$	control vectors concatenation
$O_{sd,t}$	result of influx vector screening to the route from node s to d at time t
$I_{sd,t}$	result of load vector screening to the route from node s to d at time t
$q_{ij}(t)$	number of packets queued in queue Q_{ij} of the congest ed link at time t
b_r	mean residue bandwidth of the congested link
b_j	mean influx rate of the queue Q_{ij}
c_j	periodicity of high priority packet call type j
p_j	periodicity of scout packet for call type j
T_{scout}	time for scout packet transmission of the intended call
$T_{reserve}$	transmission time of reserving scout packets of the intended call
T_{detect}	transmission time of detecting scout packets of the intended call
T_{delay}	delay time due to the response time of control vector
$R_{sp,j}$	speed-up rate of detecting scout packets call of type j
$R_{c,j}$	rate of the intended call of type j
λ	mean arrival rate of idle time slot
ϵ	bandwidth allowance
h	number of hops of the intended call
d_i	propagation delay of link at the i^{th} hop
P_T	probability of calls termination due to scout stream collision

Symbols	**Description**
E	current event list
S	scheduled period for the current event structure
B	link bandwidth
M_{in}^{h}	input traffic regulation matrix for high priority calls
M_{in}^{l}	input traffic regulation matrix for low priority packets
M_d	propagation delay time matrix
α_{sd}^{h}	mean rate of Poisson arrival of high priority call from node s to d
α_{sd}^{l}	mean rate of Poisson arrival of low priority packets from node s to d
L_{calen}	length of the calendar
L_{call}	mean call length
λ_k^h	mean call arrival rate at link k
λ_k^l	mean packet arrival rate at link k
R_{hi}	ratio of call arrival for different types
R_{tj}	ratio of call arrival for different hops