National Library
of Canada

Acquisitions and
Bibliographic Services Branch

395 Wellington Street
Ottawa, Ontario
K1A 0N4

Bibliothèque nationale
du Canada

Direction des acquisitions et
des services bibliographiques

395, rue Wellington
Ottawa (Ontario)
K1A 0N4

Your file Votre référence

Our file Notre référence

NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

Canada

Analysis of an Integrated Services Multibeam Satellite with a New Multiaccess Technique and an On-Board Prioritized Baseband Switch

Stephan Bohm

A Thesis

in

The Department

of

Electrical and Computer Engineering

Presented in Partial Fulfilment of the Requirements
for the Degree of Master of Applied Science at
Concordia University
Montreal, Quebec, Canada

May 1992

ISBN 0-315-81026-2

Canada

Abstract

Analysis of an Integrated Services Multibeam Satellite with a New Multiaccess Technique and an On-Board Prioritized Baseband Switch

Stephan Bohm

A future integrated services satellite system with a new multiaccess scheme and a prioritized on-board baseband switch is analyzed. The uplink technique serves both connection video, voice and file data users as well as connectionless interactive data users using a movable boundary/random-demand assigned multiaccess scheme (MB/R-DAMA), while in the downlink a TDM technique is employed. The frame parameters are designed to maximize user populations for each of the services with acceptable call blocking and user access delays. On-board the satellite, a baseband switch with four priority levels is presented and analyzed under both uniform and hotspot traffic. The on-board switching system is designed to minimize the loss while providing a high throughput and incorporates both finite input and output buffering and speedup. As well, a head of line resolution algorithm is shown to be easily implemented to improve the throughput of the two lowest priority users. The relationship between head of line resolution, switch speedup and buffering are examined to determine the delay and loss characteristics of each service.

Acknowledgements

I would very much like to thank my supervisor Prof. A.K. Elhakeem for his words of encouragement, patience, constructive criticism, and guidance throughout the course of this thesis.

Dedicated to my parents

Table of Contents

List of Figures

List of Tables

List of Symbols

B_W : Video call blocking probability.

DLR : Ratio of the downlink to uplink rate.

DR_V : Mean reservation delay for a voice call.

C/N_0 : Carrier to noise ratio.

E_b/N_0 : Energy per bit to noise ratio.

E_i' : Final distribution of empty spaces.

EIRP : Effective isotropic radiated power.

E_{FD_i} : Distribution of the number of empty spaces left for file sister packets.

Eh_{FD_i} : Probability distribution of the number of empty spaces left at the hotspot output for file sister packets.

Eu_{FD_i} : Probability distribution of the number of empty spaces left at the uniform outputs for file sister packets.

FD : File data user.

F_p : Frame period.

F_{W_k} : Probability distribution of the number of video calls on the frame.

GR_W : Total offered traffic for video Slotted Aloha reservation.

G/T : Gain to noise temperature.

H_V : Hotspot traffic probability for voice traffic.

IBD_W : Video input buffer delay.

IBL_W : Video input buffer loss.

ID : Interactive data user.

L : Switch speedup.

M_W : Video user population.

$M_W^{'}$: Mapped total number of video users.

NxN : Switch size.

NC : Number of connection user slots on the frame.

NC_W : Mapped number of video user connection slots per frame.

NF : Number of slots per frame.

NR_{FD} : Number of file data reservation slots per frame.

OBD_{DLR} : Output buffer delay for a given DLR.

OBL_{DLR} : Output buffer loss for a given DLR.

p : Probability that a packet arrives per time slot at an input port.

P_k : Distribution of the number of packets arriving at an output for the single user case.

P_V : Probability that a video packet arrives at an input buffer per time slot.

Pb_{ID} : Interactive data switch blocking probability.

Pmb_W : Mean blocking probability for video under hotspot traffic.

Pob_k : Distribution of the number of packet arriving to the output buffer.

Prs_W : Video reservation success probability.

Ps_{FD} : Probability that there is a file sister packet available.

Psb_{FD} : File sister packet blocking probability.

$Psb_{h_{FD}}$: Probability that file sister packets are blocked at a hotspot output.

$Psb_{u_{FD}}$: Probability that file sister packets are blocked at a uniform output under hotspot traffic.

$Pmsb_{FD}$: Mean file sister packet blocking probability under hotspot traffic.

R_W : Video traffic ratio.

Rb_W : Video bit rate.

RQ_{FD} : Rate at which file data calls are generated per frame.

RT : Satellite round trip propagation delay.

SR_V : Voice reservation Slotted Aloha throughput.

T : Output buffer transition matrix.

TE : Mean total number of slots available to interactive users per second.

TM : Mean number of equivalent connection slots for interactive users per second.

S : Output buffer system length.

SI : Input buffer system length.

U_{ID} : Interactive data input buffer service probability per time slot.

V : Voice user.

W : Video user.

W_i : Probability distribution of video users at an output port.

Wh_i : Video user distribution at the hotspot output port.

Wu_i : Video user distribution at a uniform output port under hotspot traffic.

Φ_{W_i} : Probability distribution of the video input buffer.

Ψ : Total traffic from the input buffers to the input switch input port.

Ψ_h, Ψ_u : Total traffic at an input port to the switch destined to the hotspot and uniform output respectively.

$\Psi_{h_{FD}}, \Psi_{u_{FD}}$: File sister packet traffic generated at each input to the hotspot and uniform output ports.

λ_W : Arrival rate of video calls.

Θ_i : Output buffer probability distribution.

μ_W : Service rate of a video call.

Chapter 1

Introduction

1.1. Multipurpose On-Board Switching Satellites

The introduction of the Intelsat VI satellite in 1989 for the first time saw the incorporation of a microwave switching matrix on-board a satellite, and represents the start of a new era in satellite communications. By placing the switch on-board the satellite the delay is reduced by half over that of conventional satellite systems which require a ground based hub switching station. With the reduction in the delay and the ability to connect users over very large distances, satellites now have the means to provide voice grade service directly to individual user terminals instead of only providing trunk service for the public telephone system. In addition, the use of higher Ku-band (12.5-18 GHz) frequencies along with multiple spot beams has made possible the use of very small aperture terminals (VSAT), which typically require only 1.8 m antennas to transmit at a rate of 56 Kbps. With the availability of small inexpensive terminals, satellite communications has become feasible for small business and the home user. The combination of on-board switching and VSATs now place satellites in a position to fully complement and compete with terrestrial voice and data networks.

At present domestically, satellites are still mainly used for television distribution while point-to-point telephony is almost exclusively limited to overseas calls. Intelsat, which carries about 60% of all international communications traffic, is composed of 90% point-to-point telephony with the other 10% made up of television distribution and data

communications. In the U.S. domestic market however, television distribution presently represents almost 60% of all the traffic, while point-to-point telephony and data communications represent the other 40%. Currently, only 5% of the $45 billion long distance market in the U.S. is being supplied by satellite communications. While broadcast television will remain a dominant user of satellites, the next ten years will also see vigorous competition for a share of the long distance market. With the ability to access private networks, businesses and even individual users switching satellites are expected to capture up to 20% of this market [1.1]. In addition, with the ability to establish a connection anywhere in the continental U.S., switching satellites are uniquely positioned to take advantage of the up and coming mobile telecommunications market. Indeed, satellite communication systems like terrestrial systems are moving toward the integration of services. Future satellite systems will, as shown in Fig. 1.1, most likely integrate on one platform or connect by satellite links, fixed satellite services such as point-to-point voice and data, broadcast services, as well as mobile services on land at sea and in the air. Such systems would be very versatile and provide the means to reliably interconnect calls from airplanes, ships and cars with the public telephone system or future ground based ISDNs. However, in order to integrate all these services future on-board switches will have to be based on baseband packet switching.

NASA will shortly deploy an experimental baseband switching satellite. Their advanced communications technology satellite (ACTS), scheduled to be launched in early 1993, will incorporate a baseband switch as well as a microwave switch. The ACTS program is meant to test the technology for the next generation of satellites in order to

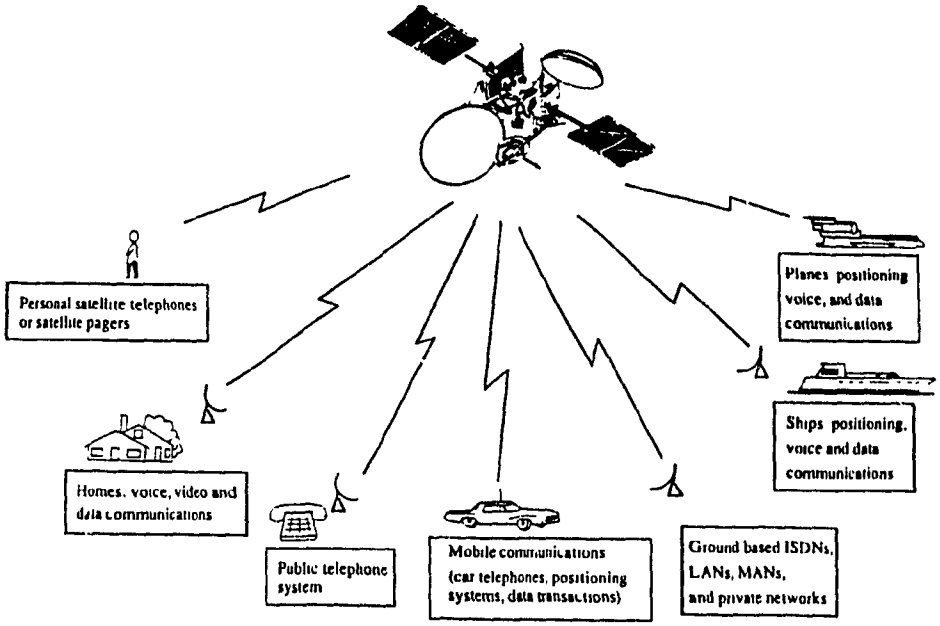


Fig. 1.1. Multipurpose on-board baseband switching satellite system.

lay the groundwork for commercial use. The ACTS system will demonstrate the feasibility of Ka-band (26.5-40 GHz) transmission which will be needed due to congestion in the C-band (4-8 GHz) as well as in Ku-band. Exploiting the higher Ka-band which is still unused offers the potential for a very large system bandwidth and will result in tens to hundreds of thousands of 64 Kbps users on a single satellite beam. The use of spot beams at Ka-band frequencies will potentially also reduce the required antenna size to 1.2 m for 64 Kbps terminals and make possible the implementation of 2.4 Kbps hand held portable telephones [1.2]. Such portable telephones will make telephone communications possible from anywhere in the world.

The ACTS system will also demonstrate the use of very high speed intersatellite

laser links which will provide the means to connect future switching satellites in order to form a completely global satellite network. Such a satellite network will have a distinct advantage over ground-based systems which are plagued by governmental regulations and equipment incompatibilities. Indeed, there is some doubt whether a completely global ground-based ISDN will ever be realized. Instead separate regional or national ISDNs seem to be more likely. A satellite network in such a situation may act as an interface to connect various networks or altogether provide an alternative to ground based networking.

1.2. Scope of the Thesis and Research Contributions

In order for future integrated service satellites to incorporate many different types of users, multiple access schemes will have to be designed that satisfy the delay requirements of the different services and at the same time manage the link capacity, so that the number of users is maximized. In addition, the on-board baseband switch will have to be able to implement a priority structure in order to ensure low delays for video and voice users for which delayed packets have to be discarded. Also of concern is the switch's fault tolerance, weight and power requirements. These become especially important as the number of switching ports increases. Incorporating a fully connected nonblocking switch may not be possible as the switch size increases and a blocking switch will have to be designed for. The blocking may be especially severe under nonuniform traffic conditions and such a switch will have to incorporate both input and output buffering in order to reduce the loss.

To this end, in Chapter 2, a fully integrated services satellite system is presented

which incorporates a blocking baseband switch with input and output buffering along with a new movable boundary/random-demand assigned multiple access (MB/R-DAMA) scheme for integrated services. The accessing scheme serves both connection and connectionless users using a moving boundary policy to ensure that no capacity is wasted. The accessing scheme is analyzed under different conditions to determine the number of users that can be accommodated.

In Chapter 3, a fault-tolerant priority baseband switch is presented for use on-board a fully integrated services satellite. The on-board switch is analyzed under uniform traffic conditions to determine the effect of priority on the buffer requirements. To reduce the blocking, a head of line resolution algorithm (HLR) is also shown to be easily implemented. The relationships between HLR, switch speedup and buffering are examined to determine the delay and throughput characteristics for each service.

Nonuniform traffic conditions are also expected to arise frequently in an integrated services environment and may severely degrade the performance of the switch system. To design for this eventuality, the prioritized switch system is analyzed in Chapter 4 under one of the most severe types of nonuniform traffics, namely hotspot traffic. Proper buffer design and switch speedup are examined in conjunction with HLR to maximize the throughput. In addition, the downlink is engineered to reduce the loss at the output buffer. The effect of increasing the switch size on the performance with HLR and input and output buffering is also examined.

Finally, Chapter 5 presents a summary of the conclusions along with suggestions for further research.

Chapter 2

A New Accessing Scheme for an Integrated Services Satellite

In this chapter the technical aspects of future Ka-band baseband switching systems are examined. As well, a new system architecture for integrated services is presented which incorporates an on-board priority switch and a new accessing scheme. The integrated services accessing scheme is analyzed to determine the throughput and delay characteristics for different frame parameters.

2.1. Overview of Ka-Band Satellites with Multiple Spot Beams

The increased number of communication satellites has lead to congestion in the C-band and will shortly lead to congestion in the Ku-band as well. Therefore, future satellites will have to exploit the Ka-band frequencies which are still unused. Transmitting at these higher frequencies will enable future satellites to substantially increase their bandwidth, which will greatly increase the potential number of users, and will lead to the implementation of a fully integrated services satellite network. Transmitting at Ka-band has the added advantage of reducing the antenna size whose gain is proportional to the square of the frequency. However, much of this advantage is curtailed in heavier rain regions by the high attenuation at Ka-band which may be as high as 20 dB [3]. In order to counter this loss and provide a versatile network configuration future satellites will employ multiple spot beam antennas, as shown in Fig. 2.1. Multiple spot beams will allow

Ka-band transmission using 1.2 m dishes over most of the continental United States (CONUS), while 1.8 m dishes may be required in regions with heavier rainfalls. With the use of spot beams, the satellite power is focused within a small region, thereby achieving higher gains which allow the use of smaller, less expensive user terminals. The carrier to noise ratio C/N_0 on the satellite link is dependent mostly on the gain to noise temperature ratio, G/T, of the receiver and the effective isotropic radiated power, EIRP, of the transmission. The EIRP, which is measured in dBW, is the power of the transmitter multiplied by the gain of the transmitting antenna. In the uplink, the large aperture of the spot beam satellite antenna achieves a high G/T making it possible to limit the power requirements of the users high power amplifier (HPA) as well as the size of the antenna. On the other hand, the small user antennas result in a low G/T on the downlink. This, however, is compensated for by the high EIRP of the satellite. Even with limited power, the satellite maintains a high EIRP due to the high gain of the satellite antenna.

In future integrated services satellite systems, the uplink will most likely be either configured as a time division multiple access channel (TDMA) or a combination of both multiple frequency and TDMA (MF/TDMA). Multiple carriers will be used in order to increase the transmission rate while satisfying the power limitations of the user terminals. By using TDMA, different services will be able to access the satellite either through random access, demand assigned access schemes or a combination of both. In addition, future systems will employ hopping spot beams which will enable the satellite to configure the system to varying traffic needs and at the same time increase the power on demand to counter rain attenuation. Hopping beams will reduce the number of transmitters

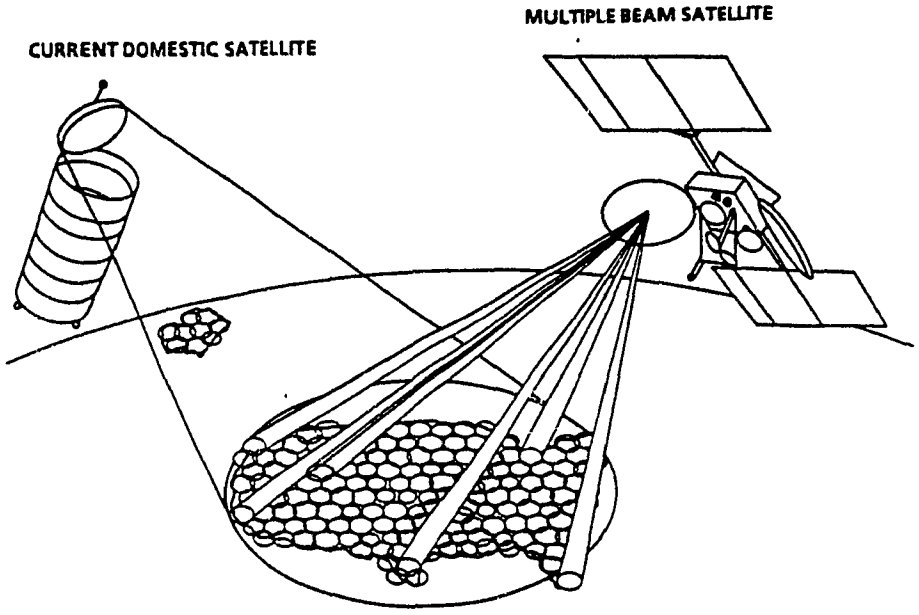


Fig. 2.1. Multiple spot beams versus a single beam system [4].

needed on-board the satellite by hopping the beam from one location to another depending on the load. This will allow for frequency reuse reducing the required capacity of the overall system. In addition any extra power which is available may be directed to rain attenuated areas in order to increase the gain [5].

2.1.1. On-Board Processing

On-board packet switching will also be needed to integrate a wide variety of different services and to reduce the delay by half from present day hub-switched systems. In present day systems, users first transmit to a ground switching hub via the geostationary satellite. The hub performs the switching and then retransmits the transmission via the satellite to the destination. In this way, two round trip propagation delays are incurred for a total delay of 540 ms. Such delays are especially annoying for voice users necessitating the use of an on-board switch.

A typical on-board baseband switch system is shown in Fig. 2.2 [6,7,8]. As seen, the incoming signals are first demodulated/decoded, after which the packets are routed to their destination ports and then remodulated/encoded and transmitted to their destinations. The advantage of baseband switching over switching at the intermediate frequency (IF) is that with baseband switching, a full time-space switch is possible in the uplink. Users place packets on the frame, as shown in Fig. 2.3., irrespective of the destinations of the packets and the baseband switch then switches these to their respective destinations based on the packets destination address. In contrast, IF switching implements only a space switch and the time dimension has to be controlled in the uplinks. This is done by assigning a particular time slot in the frame to each user for each destination beam, as shown in Fig. 2.4. A user in one beam area then waits until the slot comes around to a particular destination and transmits a burst in the sub-slot assigned to him. The problem with this method is that it wastes capacity at low traffic and incurs a high delay, especially as the user population increases.

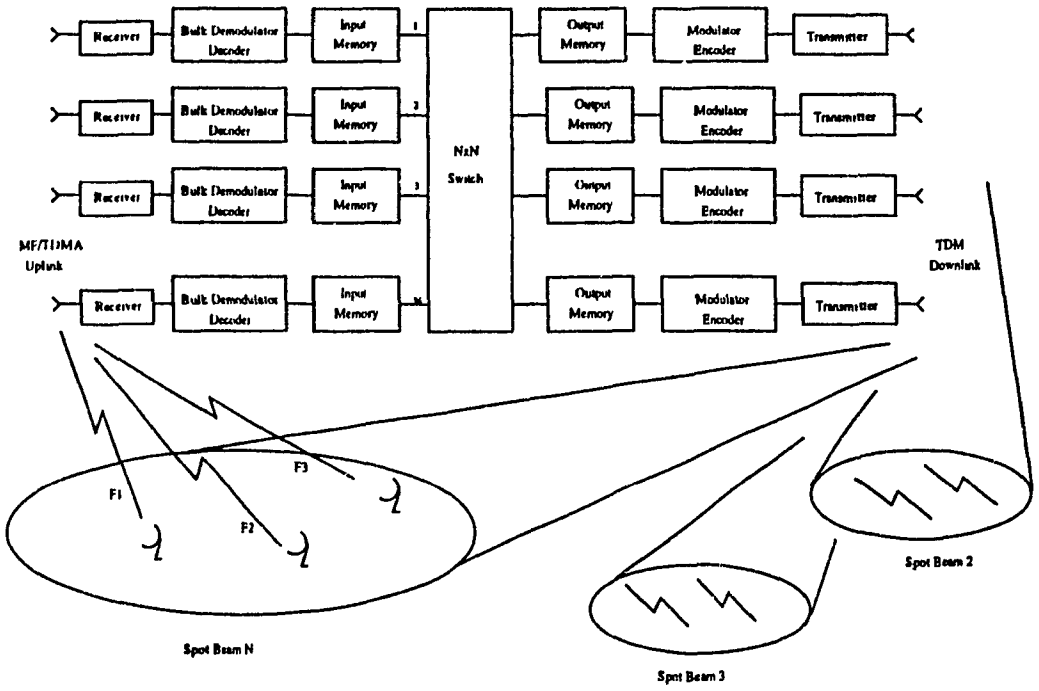


Fig. 2.2. On-board baseband switch system.

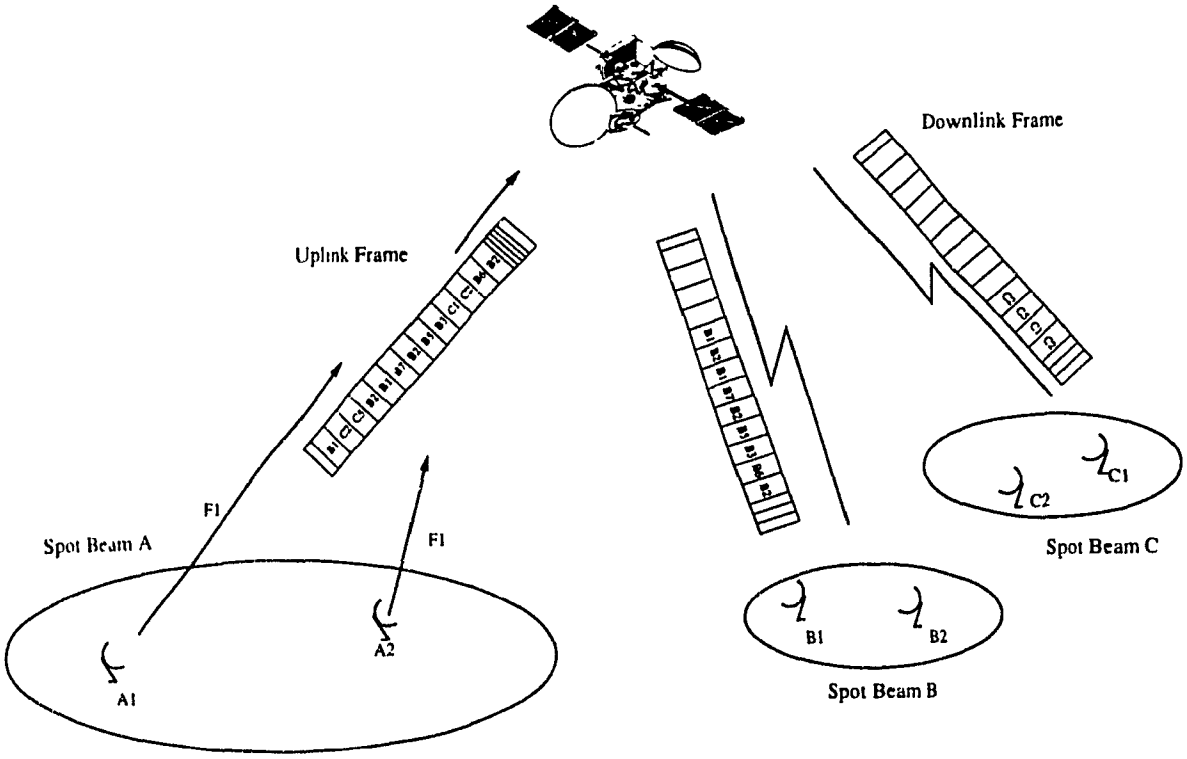


Fig. 2.3. Full time-space uplink access with baseband switch.

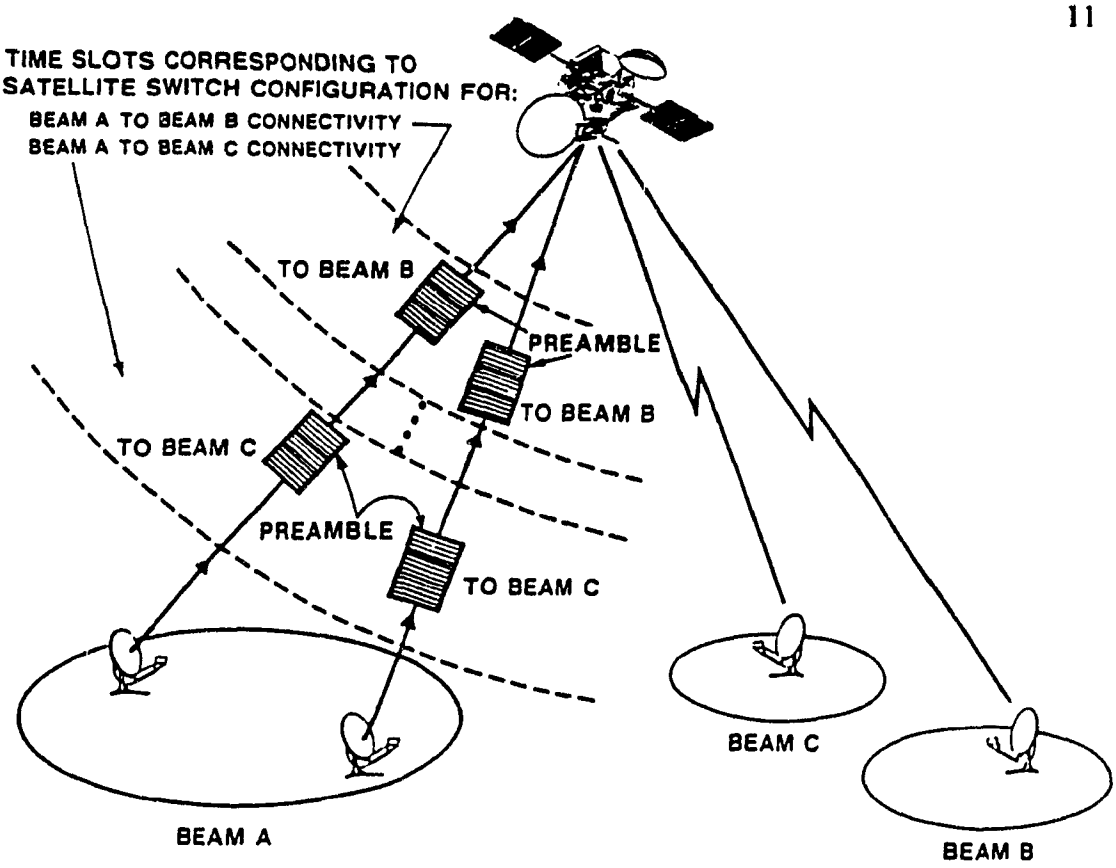


Fig. 2.4. Uplink bursts in an IF switching system [4].

The use of on-board baseband packet switching will also allow many different types of services to be accommodated on a packet level and will allow for the implementation of a priority structure. In addition, packet switching is completely independent of any external ground control while IF switches require monitoring to set up the switch configuration. The disadvantage of baseband switching however, is that it will necessitate the use of a demodulator/modulator for each switch port, which increases the power requirements especially as the switch size is increased. This, however, does provide a regenerative link improvement of a few dB. The number of demodulators needed is especially critical in an MF/TDMA system where a demodulator is required for

each frequency. In the case of a 20 beam system for example, with 20 frequency multiplexed 5 Mbps channels per uplink beam, the satellite requires a total of 400 demodulators. To counter the excessive amounts of power and added weight that such a large number of demodulators would impose, bulk demodulators have been developed. Bulk demodulators, as their name implies, demodulate a number of signals simultaneously and can be implemented in analog form using surface acoustic wave devices (SAW) or digitally using DSP techniques. It has recently been shown that up to $1x10^4$ channels may be demodulated with only 100 bulk demodulators, each with a bandwidth of 10 MHz [9,10].

Thus future integrated service satellites will have to exploit Ka-band frequencies and utilize hopping spot beams to make efficient use of the channel capacity. Hopping beams will reduce the required bandwidth while providing a means of increasing the satellite power on demand to counter rain attenuation. In addition MF/TDMA will be used in order to increase the throughput of the .2 m user terminals which have a limited power and gain. MF/TDMA, will however require bulk demodulators in order to limit the power and weight requirements on-board the satellite. In the following section, two next generation on-board switching satellites which use some of these techniques are examined.

2.1.2. Next Generation Systems

2.1.2.1. The NASA ACTS System

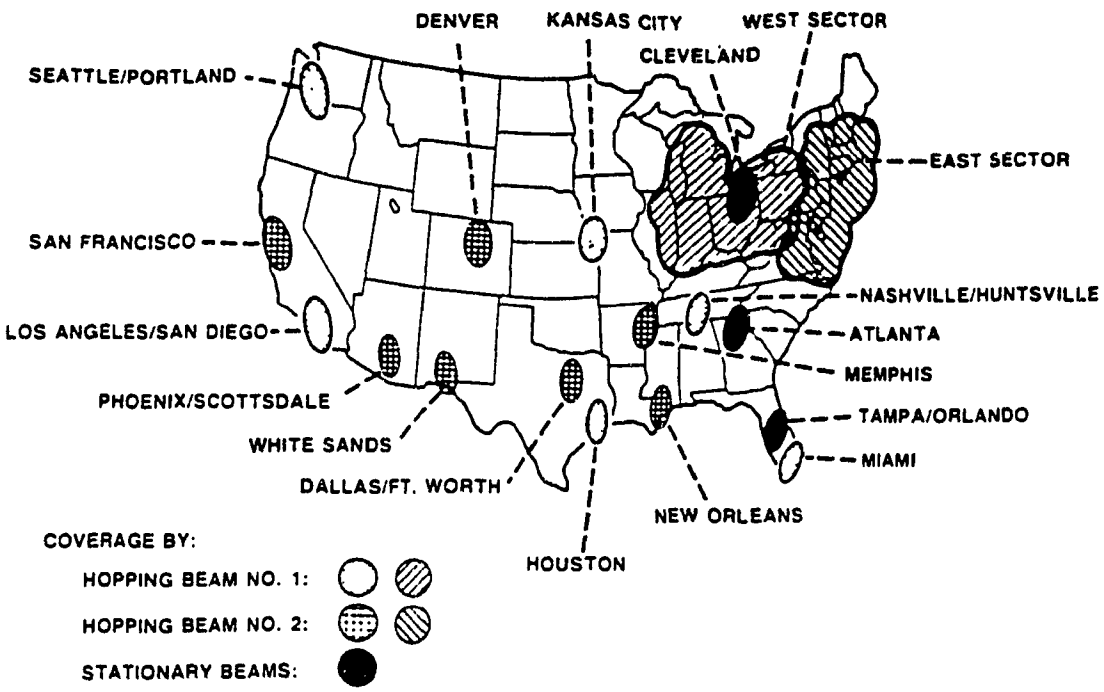
The first system that is examined is NASA's experimental advanced communications satellite (ACTS), which is scheduled to be launched in early 1993. The ACTS uses 30/20 GHz frequencies for uplink and downlink transmission and incorporates both an intermediate frequency (IF) microwave switch and a baseband switch. The 3x3 IF switch is to be used with fixed spot beams to service high burst rate users (HBR). The baseband switch is a 2x2 switch and is to be used with low burst rate users (LBR) which will utilize hopping beams. Both the HBR and LBR users use 220 Mbps TDMA channels. The LBR users, however, may be configured as either two 110 Mbps or eight 27.5 Mbps MF/TDMA channels. On the downlink, TDM is assumed at 220 Mbps. Both the uplink and the downlink use serial minimum shift keying (SMSK). On the ground, five and three meter antennas are to be used along with smaller 1.2 m antennas which will also be examined for 64 Kbps terminals. On-board the satellite, 2.2 m and 3.3 m antennas are used for the uplink and downlink, respectively. The EIRP of the ACTS system is designed to be 60 dBW, which 20 dBW greater than that of current satellites [11,12].

DAMA will be used in ACTS to dynamically configure the system to different traffic loads. However, for experimental purposes control will be maintained on the ground by one station which is designated the master control station (MCS). Users request 64 bit slots in the frame by using special control channels. Requests are processed by the MCS which then assigns slots to users for the durations of their calls. At the end of transmission the MCS is notified and the slots are returned to the common pool [13].

In the ACTS experimental system, spot beams will remain fixed on Cleveland, Atlanta, and Tampa while two other spot beams will hop between a total of 13 discrete locations and any two regions known as the east and west sectors, as shown in Fig. 2.5. In total, a little more than 20% of CONUS is served by ACTS, which is deemed enough for experimental purposes. To cover all of CONUS in future systems it is expected that 10 beams will be needed. The hopping beams are formed by selecting an isolated horn or a three horn subset of a horn array. Each of the uplink and downlink hopping beams are independently controlled with a switching time of less than 1 μs and may be switched to many different locations within each 1 ms TDMA frame.

Under rain fades, the LBR system can reduce the rate by half and apply a 1/2-rate forward error-correcting code (FEC) which adds an extra 10 dB margin to the 5 dB margin under clear weather conditions. In the downlink, travelling-wave tube amplifiers (TWTA) may also be switched between 8 and 40 watts to counter rain fades. This ability to change levels in the TWTAs directs more power only to the areas in need and saves power in the others [14].

The ACTS will also test the next generation of laser intersatellite links. A 870 nm gallium arsenide laser will be used to transmit around 220 Mbps between the satellite and the space shuttle. This is the start of satellite laser communications which will eventually employ links with rates from 2 to 4 Gbps in order to effectively implement a complete global communications network [1].



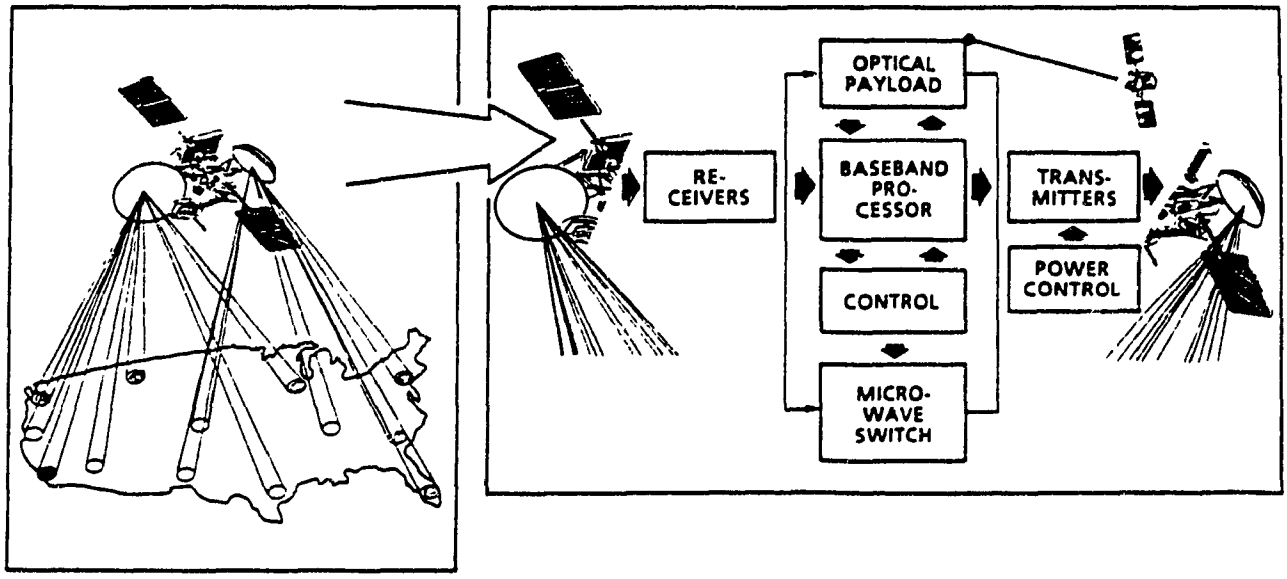


Fig. 2.5. ACTS system [4].

2.1.2.2. TRW Design

While the ACTS satellite incorporates many techniques, it is an experimental system which may be configured in many different ways. A more detailed study of a specific system with a 1000 Mhz composite bandwidth and the ability to service many thousands of 64 Kbps terminals has been undertaken by TRW [10]. The design does not assume any particular user type nor does it deal with an accessing scheme. The design is meant to be generic, concentrating rather on the physical link.

In their design, each 64 Kbps terminal requires only two watts of power to transmit at 30 GHz over a 1.2 m antenna. QPSK is used on the uplink with differential detection at a penalty of 2.3 dB with respect to coherent detection. In the future, chip-level implementation of coherent QPSK may reduce the complexity and cost of coherent detection which will further reduce power requirements of the transmitter terminal. A 3/4 FEC is used resulting in a symbol rate of 42.67 Ksps. Assuming that for bulk demodulation, the carrier separation must be twice the symbol rate, 11718 continuous users can be accommodated within the 1000 MHz bandwidth. The actual user population may however be many times this amount depending on the activity of the users. The bulk demodulators are each assumed to have a bandwidth of 10 Mhz and be capable of demodulating up to 100 channels. For the composite bandwidth, 100 such bulk demodulators will be needed. In order to achieve a bit error rate (BER) of 10^{-7}, an energy per bit to noise ratio (E_b/N_0) of 14 dB is designed for, as shown in Table 2.1. As seen, the satellite receiver antenna gain must be at least 33 dB. A minimum of eight spot beams are needed to cover CONUS with this gain. On the downlink, BPSK is chosen over QPSK

because of the larger E_b/N_0 requirement of differential QPSK and the need to minimize the cost and complexity of the user terminals. Four 180 Mbps TDM downlink beams are assumed which also use a 3/4 FEC. The downlink characteristics are given in Table 2.2. As can be seen the satellite requires an EIRP of 63.1 dBW, while the ground station antenna requires a gain of 45.8 dB at 20 GHz.

Table. 2.1.
Uplink Parameters for the TRW Design [10]

Transmitter power, dBW	3.0
Line loss, dB	−1.0
Antenna gain (1.2m), dB	49.2
Pointing loss, dB	−1.0
EIRP, dBW	50.2
Path loss, dB	−213.7
Atmospheric loss, dB	−0.6
Rain loss, dB	−8.0
S/C antenna gain (edge of beam), dB	33.3
Received carrier power, dBW	−138.8
System noise temperature, dB-K	27.6
Boltzmann's constant, dB(W/K-Hz)	-228.6
N_0, dB(W/Hz)	−201.0
Bit rate (64 kbps), dB(sec^{-1})	48.1
E_b/N_0, dB	14.1

Table. 2.2.
Downlink Parameters for the TRW Design [10]

EIRP, dBW	63.1
Path loss, dB	−210.2
Atmospheric loss, dB	−0.8
Rain loss, dB	−4.0
E/S antenna gain, dB	45.8
Pointing loss, dB	−0.8
Line loss, dB	−1.0
Received carrier power, dBW	−107.9
System noise temperature, dB-K	25.7
Boltzmann's constant, dB(W/K-Hz)	−228.6
N_0, dB (W/Hz)	-202.9
Bit rate (180 Mbps), dB(sec^{-1})	82.6
E_b/N_0, dB	12.4

2.2. Satellite Accessing Schemes for Different Services

The satellite accessing schemes that have been examined to date have dealt mainly with single traffic types and fall into three main categories, fixed assignment, random access, and demand-assigned multiple access (DAMA). These three cases have evolved to meet the needs of constant high traffic with long durations, sporadic traffic with short to medium durations, and sporadic traffic with long durations respectively. Fixed assignment schemes have traditionally been used for satellite trunk systems and include FDMA and TDMA. Both work well under heavy traffic environments with throughputs around 0.8, but waist capacity when the traffic becomes bursty. In addition, they are not versatile in that the frame or the bandwidth has to be reconfigured each time users are

added. Random access, on the other hand, is ideal for future interactive services which are bursty and which generate small amounts of traffic.

Random access schemes include Aloha, Slotted Aloha and Select Reject Aloha (SREJ) which all use random retransmission policies, as well as SREJ/FCFS (First Come First Served) and Announced Retransmissions Random Access (ARRA) which use more innovative methods to avoid retransmission collisions. Due to partial overlap, the throughput of Aloha is limited to between 0.13 and 0.18 depending on the message length distribution. Forcing users to transmit in slots as in Slotted Aloha theoretically doubles the throughput to 0.368 but due to packetization and synchronization, this is reduced to between 0.2 and 0.3. The trade-off between message length and throughput is that larger packet lengths require more overhead which reduces the throughput, while decreasing the packet length offsets the gain over that of pure Aloha, since packets carry less information. To reduce the dependence on the message length distribution, SREJ packets messages but does not slot them [15]. In this way, partial overlaps constitute collisions but packetization requires only the colliding packets to retransmit and not the entire message. It has been shown that the performance of the system approaches 0.368 irrespective of the message length but due to overhead the throughput is limited to around 0.3. To reduce the prospect of repeated collisions which limit the throughput SREJ/FCFS and ARRA have been examined. In SREJ/FCFS, packets are sequentially not randomly retransmitted so that collisions may not occur between retransmitted packets. Collisions, however, may still occur between retransmitted and new packets. At the expense of requiring central control, the throughput may be increased to between 0.45 and 0.6

depending on the overhead. ARRA, uses a slotted system in contrast to SREJ/FCFS along with a low-rate announcement subchannel so that collisions between new and retransmitted packets are avoided [16]. The low rate subchannel may be implemented by adding a minislot pool to the frame in which a minislot corresponds to one data slot in the frame. In ARRA, users pick their retransmission slots at random in advance in anticipation of a collision and transmit this information along with the packet. If no collision occurs, the minislot pool is not effected but if there is a collision the pool is updated so that in the next frame new users will know the location of the retransmitted packets and will not transmit in these slots. In addition, by using the minislot pool retransmitting packets that collide for a second time do not transmit in the next frame, thereby allowing potential collision slots to be used by new users. The throughput achieved using ARRA ranges between 0.5 and 0.6 depending on the overhead.

DAMA systems in present satellite systems are used to service traffic such as video and data as well as telephone trunks which dynamically change their bandwidth for changing traffics. DAMA schemes try to balance the low delays of random access and the high throughput of TDMA by first using random access to make a reservation, after which slots on the frame are dedicated to the user for the duration of the call [17]. DAMA systems dedicate a small portion of the frame capacity for random access reservations while the rest of the frame is divided into information slots. Random access is used as the reservation accessing scheme since the generation rate of new connection requests is assumed to be bursty in nature. In using a reservation mechanism however DAMA techniques incur an initial round trip propagation delay. Such delays are

acceptable for most connections including future private voice terminals since once a connection is established packets are assured of a constant delay just like in fixed assignment schemes. DAMA also has the potential to serve an almost infinite population like random access. Thus, DAMA combines the best of both random access and fixed assignment achieving a throughput of between 0.6 and 0.8 [18].

Table. 2.3.
Performance of Satellite Accessing Schemes [18]

Multiaccess Protocol	Capacity (Max. Throughput)	Delay	Stability	Robustness	VSAT Cost/ Complexity
TDMA	0.7-0.8	med-high	good	med	med
FDMA SCPC	0.7-0.8	med	good	high	v. low
Fixed Assigned Techniques					
ALOHA	0.13-0.18	low	poor	high	v. low
Slotted ALOHA	0.25-0.368	low	moderate	high	low-med
SREJ-ALOHA	0.2-0.3	low	moderate	high	low
ARRA	0.5-0.6	low	moderate	med	med-high
Random Access Techniques					
DAMA TDMA	0.6-0.8	med high	good	poor	high
Controlled Access Techniques					

Future accessing schemes will have to deal with many different services and it is clear that no single accessing scheme will be able to fulfil the needs of all the different services. While DAMA systems may balance fairly similar types of service such as voice and data communications using one access technique future systems will incorporate bursty services as well and will have use a combination of random access and DAMA to provide adequate service.

One design by Suda et at. [19] serves both bursty and long duration users by dividing the frame into two portions, a DAMA channel and a Slotted Aloha channel. Users from each user group access the channel independently of each other. Bursty users access the Slotted Aloha channel while reservation users access the DAMA portion of the channel by first making a reservation via Slotted Aloha. The problem with this system is that it uses a fixed frame policy and it physically divides the frame into two separate parts, which serve two separate populations. In this way capacity on the frame may be wasted. Situations may arise in which one type of service is not using the full capacity while the other is overloaded. Another scheme proposed by Lee and Mark [20] assumes that DAMA users have priority over bursty users in all the slots in the frame. In this scheme bursty users may compete only in any of the free slots not being used by the connection users. This method is more equitable under low to medium connection user traffic, but under high connection user traffic scenarios, bursty users suffer at the expense of connection users. In both these designs however only one type of connection user traffic was considered. In the next section a movable boundary/random-DAMA (MB/R-DAMA) scheme is proposed to integrate four different services namely, video, voice, file

data and interactive data. A detailed description of the MB/R-DAMA accessing scheme is given in the following section as part of the design of a future integrated services satellite.

2.3. System Configuration for a Future Integrated Services Satellite

In order to meet the delay and throughput requirements of a wide variety of services, a new integrated satellite system is proposed and analyzed in this and the next sections as well as in the following chapters. The new system incorporates an on-board packet switch with a priority structure as well as input and output queuing. In addition, the system uses a new accessing method which incorporates a movable boundary strategy in order to ensure that no capacity is wasted in the uplink.

2.3.1. Integrated System Description

The proposed satellite system is shown in Fig. 2.6. Both connection and connectionless types of service are provided on the frame, however each utilizes packets which are switched on-board the satellite to their respective destinations. After demodulation and decoding, the packets in the frame are buffered at the input to the switch in order to reduce the switch loss which might be quite severe, especially under nonuniform traffic conditions. Packets that are switched are then buffered at the output of the switch since more than one packet may be switched to an output and the downlink rate is assumed to be fixed. As shown, the connection user requests are removed by the uplink controller for processing by the on-board switch manager processor. The on-board

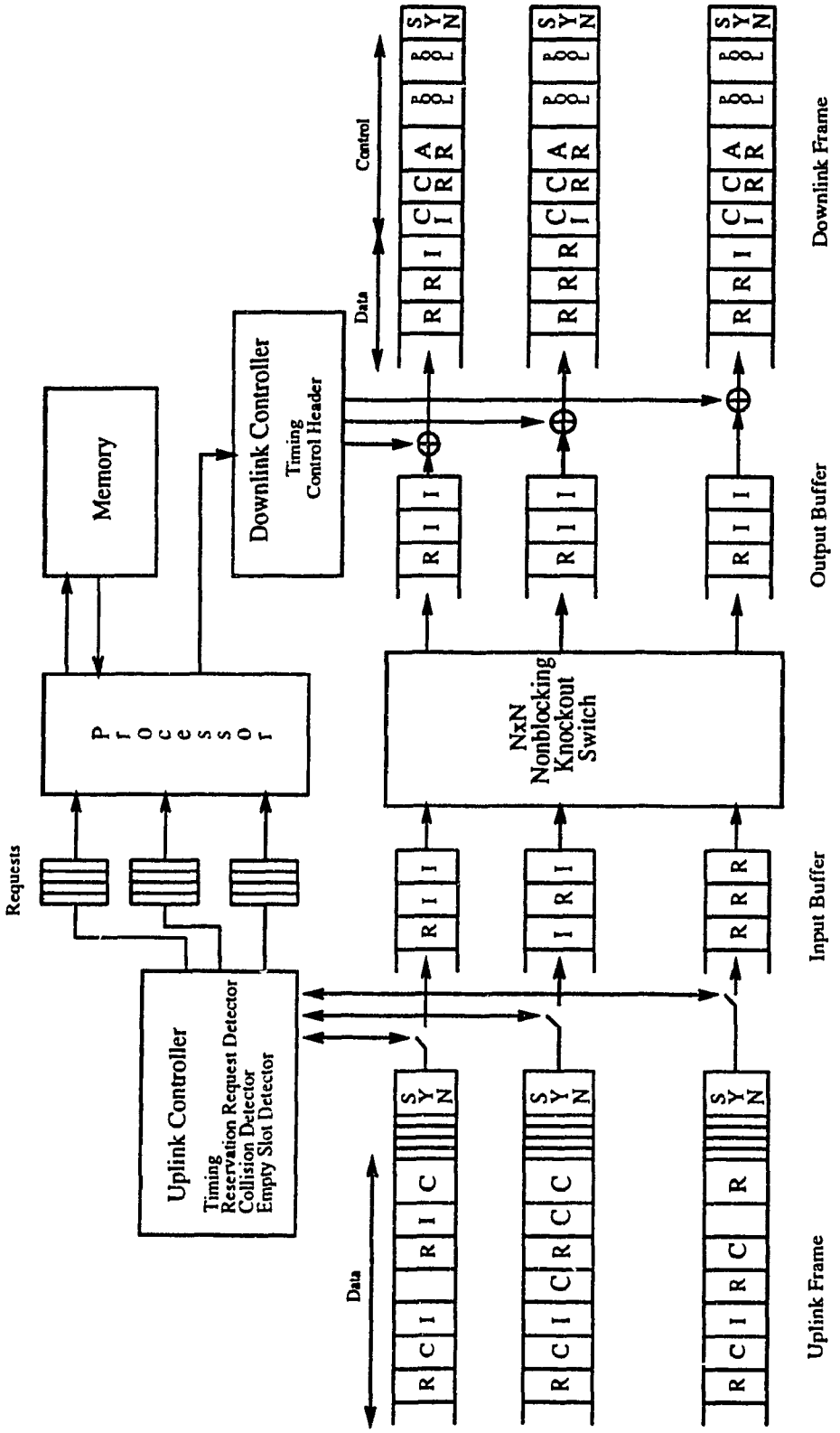


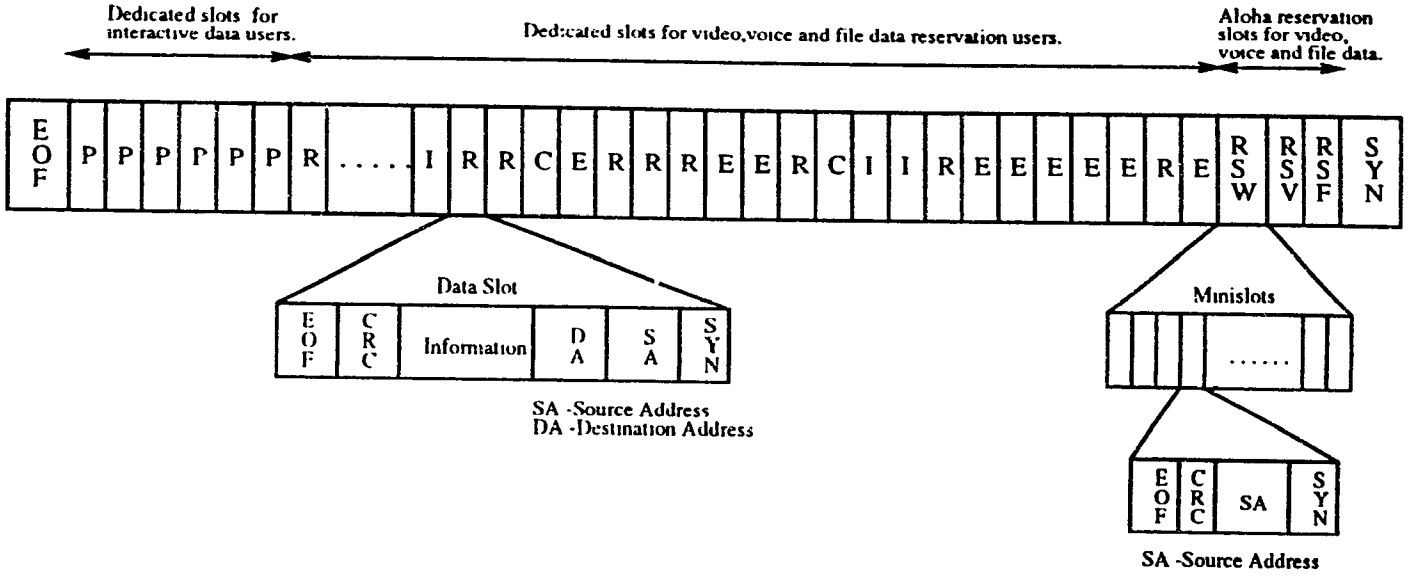
Fig. 2.6. Integrated services on-board processing satellite system.

processor assigns one or more slots on the frame for each request depending on the user type and updates the available pool of empty slots. Acknowledgments are then inserted into the downlink frame along with the switched data packets. Users monitor the TDM downlink stream and select the packets which are destined to them.

2.3.2. MB/R-DAMA Accessing Scheme for Integrated Services

The uplink frame of the MB/R-DAMA accessing scheme, as shown in Fig. 2.7, tries to balance the needs of video, voice and file data users which have relatively long call durations with those of bursty interactive data users. To this end video, voice and file data users are provided with a connection type of service while interactive data users are provided with a connectionless service. The bursty interactive data users compete in a movable boundary scheme using Slotted Aloha random access in dedicated slots, to insure a minimum service, as well as in any unused slots that are not being used by connection users. In this way, when the connection traffic is low the extra capacity reduces the delay of the interactive data users. In contrast, the call durations of the connection users are relatively long and would not be well served by a random access scheme. The connection users use a DAMA scheme instead, with Slotted Aloha used for the reservations. Slotted Aloha is used because of its simplicity in order to minimize the complexity of user ground terminals.

When a connection type terminal becomes active and wants to generate a call, it places a request in one of the reservation minislots denoted by RSW, RSV, and RSF for video, voice and file data respectively. Each traffic thus has its own reservation slots and



R - Denotes an occupied reserved slot for a connection type of user (video, voice, and file data). The slot is dedicated to the user for the duration of his call.

C - Denotes a slot in which interactive users have collided.

I - Denotes a successfull interactive data packet.

E - Denotes an empty slot in the frame not being used by any service.

P - Denotes one of the dedicated slots that are set aside in the frame for interactive data users to provide a minimum service.

RSW - Denotes the aloha reservation minislots for video users.

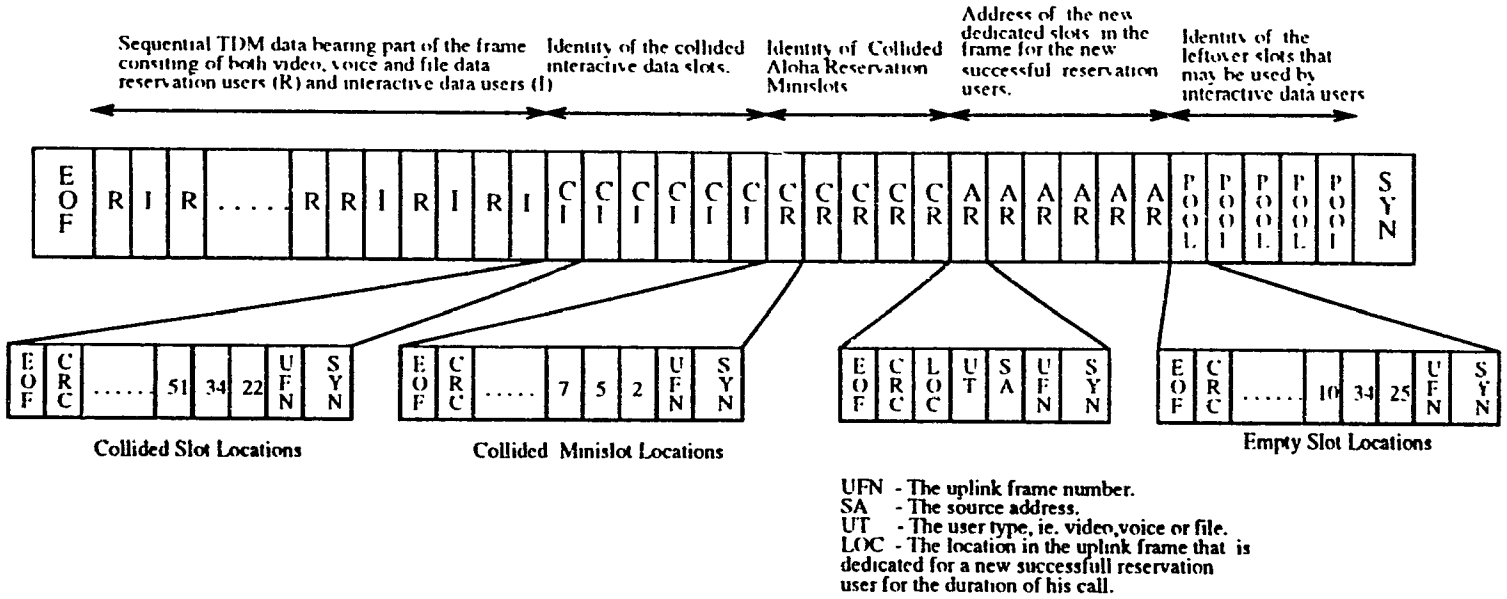
RSV - Denotes the aloha reservation minislots for voice users.

RSF - Denotes the aloha reservation minislots for file data users

Fig. 2.7. Uplink MB/R-DAMA frame.

competes only with itself. Successful reservations are admitted to the frame in a random fashion with no priority structure. If there is insufficient space available on the frame for further calls the reservations are assumed to be cleared. Acknowledgements of the collided reservation minislots are placed in the control part of the TDM downlink frame as shown in Fig. 2.8. Acknowledgements of the successful reservations follow along with the slot assignment information, i.e., the slot number(s) in the frame that are dedicated to the successful user for the duration of the call. On the ground, each connection user monitors the TDM downlink frame for their acknowledgment which arrives after one round trip propagation delay of 270 ms. Each of the AR fields includes the frame number in which the requests were sent. It is assumed that the frame numbering sequence is sufficiently long to span the round trip propagation delay plus the delays on-board the satellite, so that there is no ambiguity in which frame is being received. If the request collided as indicated by the CR field, the user then retransmits the request in the next frame in a random fashion that satisfies the Slotted Aloha accessing scheme or in the case where the request was successful, begins transmitting the call in the assigned slots for the duration of the call. Each colliding request costs one round trip delay which is the major drawback of a reservation method and so it is imperative to design the frame with sufficient reservation minislots to ensure a low collision rate.

When an interactive user terminal is active it generates packets in a bursty fashion and each packet competes using Slotted Aloha random access in any free slot in the frame, which includes the dedicated interactive P slots and the empty E connection slots that are not being used by the connection users. The empty connection slots are



POOL - Denotes the leftover reservation user slots that may be used by interactive data users.

AR - Denotes the acknowledgement of new dedicated slots in the frame for new reservation users.

CR - Denotes the identity of collided aloha reservation minislots.

CI - Denotes the identity of collided interactive data users.

R - Denotes the connection reservation users information slots.

I - Denotes the connectionless interactive users information slots.

Fig. 2.8. TDM downlink frame.

announced to the interactive users in the downlink part of the frame labelled POOL. The on-board switch manager continuously updates the POOL field in each frame based on the number of new call reservations granted. The interactive users monitor the downlink frame and combine the capacity announced by the POOL with that given by the P field, which is totally dedicated to interactive users. In addition to monitoring the downlink for available slots the interactive users also monitor the CI field to determine if previous packets sent have been successful. The CI field announces the frame number along with the slot numbers in which collisions occurred. This acknowledgement arrives at the interactive users after one round trip propagation delay. The interactive users compare their transmission record with the CI field and if a collision occurred a user randomly selects one slot out of the available slots in the next frame for retransmission in such a way as to guarantee stability for Slotted Aloha. It is also assumed that the number of users is limited with an acceptable safety margin so that Slotted Aloha is always operating in steady state. In the next section the proposed integrated services accessing scheme is analyzed with the main objective of determining the user populations of the various services that can be supported within the bounds of acceptable delays and call blocking probabilities.

2.4. Analysis of the MB/R-DAMA Integrated Services Accessing Scheme

In analyzing the performance of the multiple access scheme, it is assumed that each service has poisson arrivals and an exponential service time. The mean values that are assumed for the call arrival rate, service rate and bit rate are shown in Table 2.4.

Table 2.4.
Integrated Traffic Parameters

USER	ARRIVAL RATE	SERVICE RATE	BIT RATE
Video	$\lambda_W = 3 \frac{\text{calls}}{\text{day}}$	$\mu_W = \frac{1}{30} \frac{\text{services}}{\text{min.}}$	$Rb_W = 45$ Mbps
Voice	$\lambda_V = 3 \frac{\text{calls}}{\text{hour}}$	$\mu_V = \frac{1}{3} \frac{\text{services}}{\text{min.}}$	$Rb_V = 64$ Kbps
File	$\lambda_{FD} = 5 \frac{\text{calls}}{\text{day}}$	$\mu_{FD} = \frac{1}{20} \frac{\text{services}}{\text{min.}}$	$Rb_{FD} = 64$ Kbps
Interactive	$\lambda_{ID} = 3 \frac{\text{calls}}{\text{day}}$	$\mu_{ID} = \frac{1}{30} \frac{\text{services}}{\text{min.}}$	$Rb_{ID} = 1$ Kbps

Theses values show the wide diversity in the traffic characteristics for the different services. The precise merit of these statistics is an on-going discussion in the literature [21,22,23] and these values have been chosen to represent one possible scenario. It is also assumed that the channel rate of the uplink is 150 Mbps and each slot in the frame contains one packet of 1024 bits. The frame period F_p is designed around the connection voice and file data users which generate at 64 Kbps. To ensure that only one packet is sent per frame for voice and file users, the frame duration is set to 16 ms.

$$F_p = \frac{1024 \text{ bits/packet}}{64 \text{ Kbps}} = 16 \text{ ms.}$$

Correspondingly, there are 2343 slots per frame which is just the ratio of the uplink

$$NF = \frac{150 \text{ Mbps}}{64 \text{ Kbps}} = 2343$$

capacity to the voice and file data rate and represents the number of 64 Kbps users channels that are available. To guarantee the connectionless interactive users a basic service, *NI* slots are dedicated exclusively for interactive users. In the analysis, *NI* is taken to be one hundred. Thus there are *NC=NF-NI* slots available for connection type of users.

It is assumed in the following analysis that the Slotted Aloha reservation mechanism is operating in steady state so that from the frame's perspective, the Slotted Aloha reservation mechanism is transparent. This is to say that the number of connection users arriving at the frame is the same as that generated on the ground. From the perspective of a user, delays may be incurred in making a reservation due to Slotted Aloha collisions but the throughput of the Slotted Aloha scheme is the same as the traffic on the ground and eventually the user will get through. The model of the uplink system is shown in Fig. 2.9 with X denoting the traffic on the ground. It is assumed that blocked calls on the frame are cleared. Call blocking on the frame happens when the frame is full and there are no slots left to accommodate any more connection users. One of the criteria of the following analysis is to find the maximum user population for a certain call blocking probability. Ideally the call blocking is designed to be very small so that a user may never experience a blocked call. Under this assumption the uplink may be modeled as an open loop system.

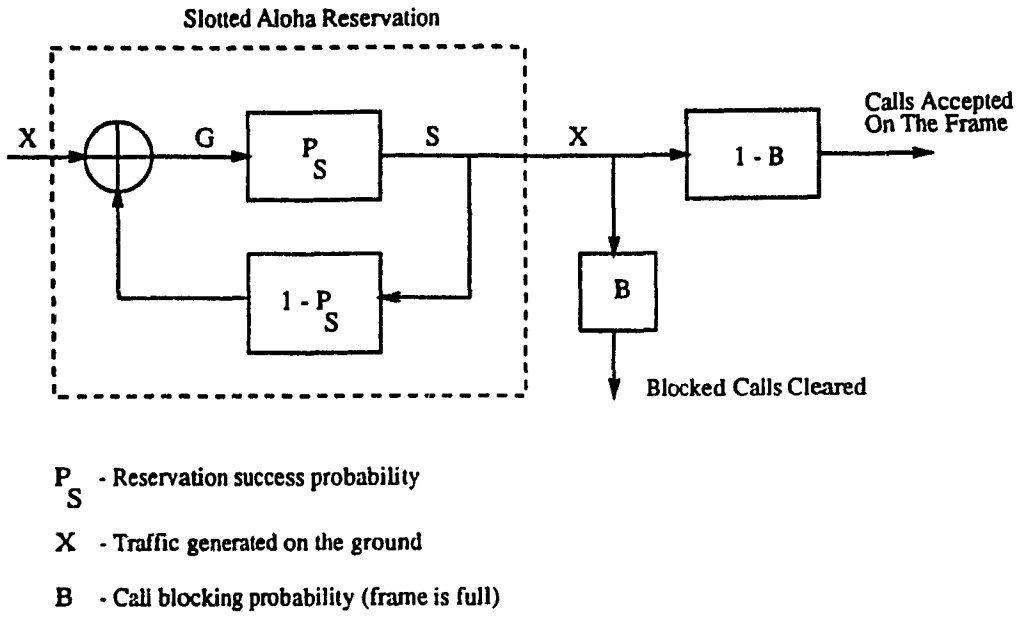


Fig. 2.9. Call blocked cleared model of the uplink system.

2.4.1. Connection Users Call Blocking Probabilities

In the following, the frame call blocking probability is found for different populations of each of the connection type services along with the Slotted Aloha reservation delay in terms of the number of reservation slots available. The call blocking probability is found from the distribution of the number of users on the frame. Each of the four types of users has a different call blocking probability due to the different number of slots a call occupies, its arrival rate and its holding time. In addition the call blocking of any user is dependent on the traffic which is on the frame from the other

users. For example, one video call may be equivalent to Y number of file calls in terms of the blocking seen by a certain number of voice users. To examine the relationship between the other traffics and the one for which the call blocking is to be found a one dimensional model is examined in which all the other traffics are linearly mapped into equivalent users for the traffic in question. By mapping all the users into one user type the multiple user case is reduced to a single user case for which the call blocking can be found using a one dimensional markov chain.

Since the Aloha reservation mechanism is operating in steady state, the frame distribution can be modeled by an M-finite population, m-server, K-finite storage queue denoted by M/M/m/K/M. Since we are considering the blocked call cleared case both K and m are equal since no buffering is assumed. The corresponding Markov chain is shown in Fig. 2.10. For voice and file users K and m are equal to *NC*, since each user takes up

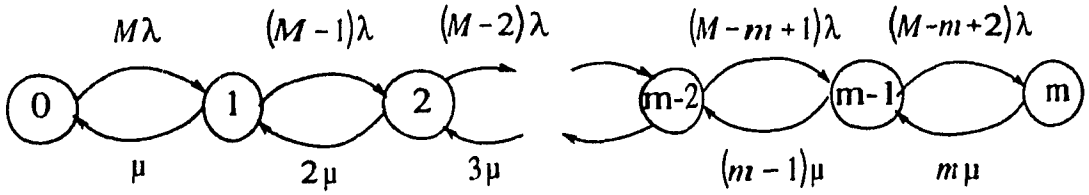


Fig. 2.10. Markov chain of the number of connection users on the frame.

exactly one slot on the frame. For video, the number of slots has to be scaled so that a slot corresponds to one video user. In addition, the other user types have to be scaled in turn to the user type for which the frame distribution and the call blocking is being calculated. In the case of video users for example, all the other traffics are mapped into corresponding video traffic using a linear mapping. The corresponding total number of mapped video users is then equal to

$$\begin{aligned} M_W^{\cdot} &= M_W + M_V VW + M_{FD} FW \\ VW &= \left(\frac{\lambda_V}{\lambda_W}\right) \cdot \left(\frac{Rb_V}{Rb_W}\right) \cdot \left(\frac{\mu_V}{\mu_W}\right) \\ FW &= \left(\frac{\lambda_{FD}}{\lambda_W}\right) \cdot \left(\frac{Rb_{FD}}{Rb_W}\right) \cdot \left(\frac{\mu_{FD}}{\mu_W}\right) \end{aligned} \tag{2.1}$$

where M_W, M_V and M_F denote the number of video, voice and file users on the ground. Since a video call takes more than one slot in the frame the number of slots has to be scaled so that one slot corresponds to one video user. The resulting number of scaled video slots is

$$NC_W = \left[\frac{R_W}{R_F} NC\right] = \left[\frac{45 \text{ Mbps}}{64 \text{ Kbps}} 2243\right] = 3. \tag{2.2}$$

These parameters can now be used to solve the probability distribution of the M/M/m/K/M queue using the solution given in [24]. The probability distribution in terms of video calls on the frame is then given by

$$FW_k = \frac{\binom{M_W'}{k}\left(\frac{\lambda_W}{\mu_W}\right)^k}{\sum_{i=0}^{NC_W}\binom{M_W'}{i}\left(\frac{\lambda_W}{\mu_W}\right)^i} \qquad k=0,1,2,\ldots,NC_W \,. \tag{2.3}$$

As an aside, eq. 2.3 cannot be used to calculate the distribution for large populations of users because of overflow difficulties in computing the binomial. However is has been shown in [25] that the distribution can be solved recursively. The recursive expression for the distribution for the finite user case can be written as

$$FW_k = \frac{1}{1+\frac{k}{\left(M_W'-(k-1)\right)\left(\frac{\lambda_W}{\mu_W}\right)FW_{k-1}}} \,. \tag{2.4}$$

The blocking for the blocked call cleared case of the M/M/m/K/M queue is the probability that all the servers are busy, i.e., the frame is full. Thus the call blocking probability for video service is written as

$$B_W = FW_{(NC_W)} \,. \tag{2.5}$$

The call blocking probabilities for voice and file data connection users can be found similarly. Now, the number of servers is just *NC* however since a voice and file user use only one slot per frame. To find the voice call blocking, the other traffic types are mapped into equivalent voice traffic. The total number of equivalent voice users is given by

$$M_V^{'} = M_V + M_W WV + M_{FD} FV$$

$$WV = \left(\frac{\lambda_W}{\lambda_V}\right) \cdot \left(\frac{Rb_W}{Rb_V}\right) \cdot \left(\frac{\mu_W}{\mu_V}\right) \tag{2.6}$$

$$FV = \left(\frac{\lambda_{FD}}{\lambda_V}\right) \cdot \left(\frac{Rb_{FD}}{Rb_V}\right) \cdot \left(\frac{\mu_{FD}}{\mu_V}\right).$$

The distribution of the number of users on the frame in terms of voice users is then

$$FV_k = \frac{\binom{M_V^{'}}{k}\left(\frac{\lambda_V}{\mu_V}\right)^k}{\sum_{i=0}^{NF}\binom{M_V^{'}}{i}\left(\frac{\lambda_V}{\mu_V}\right)^i} \qquad k=0,1,2,\ldots,NF \tag{2.7}$$

and the voice call blocking is given by

$$B_V = FV_{NF}. \tag{2.8}$$

Similarly, to find the file data call blocking all the other traffics are mapped into file and the resulting total number of file users is then

$$M_{FD}^{'} = M_{FD} + M_W WF + M_V VF$$

$$WF = \left(\frac{\lambda_W}{\lambda_{FD}}\right) \cdot \left(\frac{Rb_W}{Rb_{FD}}\right) \cdot \left(\frac{\mu_W}{\mu_{FD}}\right) \tag{2.9}$$

$$VF = \left(\frac{\lambda_V}{\lambda_{FD}}\right) \cdot \left(\frac{Rb_V}{Rb_{FD}}\right) \cdot \left(\frac{\mu_V}{\mu_{FD}}\right).$$

The distribution in terms of the number of file users is then

$$FFD_k = \frac{\binom{M_{FD}}{k}\left(\frac{\lambda_{FD}}{\mu_{FD}}\right)^k}{\sum_{i=0}^{NF}\binom{M_{FD}}{i}\left(\frac{\lambda_{FD}}{\mu_{FD}}\right)^i} \qquad k=0,1,2,...,NF \tag{2.10}$$

and the call blocking is given by

$$B_{FD} = FFD_{NF}. \tag{2.11}$$

2.4.1.1. Connection Users Reservation Delays

The performance of the Slotted Aloha call reservation mechanism is now analyzed in order to design the frame with the appropriate number of reservation minislots so that the reservation delay is minimized and stability ensured for each user population. The number of minislots for each traffic is governed by the number of users that the frame can support for a certain call blocking probability. It is assumed that the Slotted Aloha system always operates under steady-state conditions. Since each service only competes with itself for the reservation minislots, the delays of each service are independent. For video, the throughput of the Slotted Aloha reservation scheme can be written as

$$SR_W = \frac{M_W RQ_W}{NR_W} = GR_W e^{-GR_W} \tag{2.12}$$

where

$$RQ_W = \frac{\lambda_W \text{ (calls/min)}}{\frac{60}{F_p} \text{ (frames/min)}} = 5.54\text{x}10^{-7} \text{ (calls/frame)}$$

is the rate at which video calls are being generated per user per frame, NR_W is the number of video reservation minislots and GR_W is the total channel traffic consisting of newly

generated video packets plus retransmitted packets. The maximum throughput under stability for Slotted Aloha is limited to 0.368 with GR_W=1. Assuming stability, eq. 2.12 is solved numerically for GR_W, using the bisection method in order to find the success probability for a video user which is then used to find the delay. The probability of a video user being successful in making a reservation is

$$Ps_W = \frac{SR_W}{GR_W}. \tag{2.13}$$

When a user's reservation collides, the user is assumed to randomly schedule another request in the reservation minislots of the frame preceding the acknowledgment. The delay that is incurred each time a reservation is unsuccessful is the round trip propagation delay, RT, of 270 ms plus the processing delay on-board the satellite and one frame period to process the acknowledgment and randomly send another request. Since the frame duration is only 16 ms and the on-board processing delay is assumed to be designed on the order of slots not frames, the delay incurred by a collision may be approximated by just the round trip propagation delay. Then the mean delay incurred by a video before its reservation request is successful can be approximated by

$$DR_W = RT \sum_{i=0}^{\infty} i Ps_W \left(1-Ps_W\right)^{i-1} = \frac{RT}{Ps_W}. \tag{2.14}$$

The delay is thus the mean number of times that a request collides, $1/Ps_W$, multiplied by the round trip propagation delay incurred per collision.

Similarly, the Slotted Aloha reservation delays for voice and file data can be found by solving for the success probabilities for each of the traffic types, respectively. The

throughput for voice reservation is given by

$$SR_V = \frac{M_V RQ_V}{NR_V} = GR_V e^{-GR_V} \tag{2.15}$$

where

$$RQ_V = \frac{\lambda_V \text{ (calls/min)}}{\frac{60}{F_P} \text{ (frames/min)}} = 1.33\text{x}10^{-5} \text{ (calls/frame)}.$$

Here, RQ_V is the rate at which voice calls are being generated per user per frame, NR_V is the number of voice reservation minislots and GR_V is the total channel traffic. Solving for GR_V the success probability for a voice user in making a request is given by

$$Ps_V = \frac{SR_V}{GR_V} \tag{2.16}$$

and the total mean delay that a voice user incurs in making a reservation is

$$DR_V = RT \sum_{i=0}^{\infty} iPs_V\left(1-Ps_V\right)^{i-1} = \frac{RT}{Ps_V}. \tag{2.17}$$

For file users the offered load GR_{FD} is solved for from eq. 2.18

$$SR_{FD} = \frac{M_{FD} RQ_{FD}}{NR_{FD}} = GR_{FD} e^{-GR_{FD}} \tag{2.18}$$

where

$$RQ_{FD} = \frac{\lambda_{FD} \text{ (calls/min)}}{\frac{60}{F_P} \text{ (frames/min)}} = 9.22\text{x}10^{-6} \text{ (calls/frame)}.$$

The success probability is then denoted by

$$Ps_{FD} = \frac{SR_{FD}}{GR_{FD}} \tag{2.19}$$

and is used to find the file reservation delay in eq. 2.20

$$DR_{FD} = RT \sum_{i=0}^{\infty} iPs_{FD}\left(1-Ps_{FD}\right)^{i-1} = \frac{RT}{Ps_{FD}}. \tag{2.20}$$

2.4.2. Interactive Access Delay

Interactive data users are connectionless users that compete using Slotted Aloha in the *NI* slots set aside for them in the frame as well as in any unused connection user slots. In this way there is a moving boundary policy with the interactive data being able to take advantage of the extra capacity when the frame is lightly loaded with connection users. It is assumed that interactive data generates at 1 Kbps so that one slot in a frame would be filled per second. Since there are $1/F_p$ frames per second and there are *NI* slots dedicated per frame there are equivalently $\frac{NI}{F_p}$ dedicated slots available for interactive data users per second. To determine the number of connection slots that are also available the distribution of the number of equivalent interactive slots is found using the same mapping method as before, where each of the connection traffics in turn is converted into equivalent interactive users. The distribution of the number of equivalent interactive users that are on the channel per second is given by

$$FID_k = \frac{\binom{M_{ID}}{k}\left(\frac{\lambda_{ID}}{\mu_{ID}}\right)^k}{\sum_{j=0}^{NC_{ID}} \binom{M_{ID}}{j}\left(\frac{\lambda_{ID}}{\mu_{ID}}\right)^j} \qquad k=0,1,2,\ldots,NC_{ID} \tag{2.21}$$

where

$$M_{ID} = M_W WI + M_V VI + M_{FD} FI$$

$$WI = \left(\frac{\lambda_W}{\lambda_{ID}}\right) \cdot \left(\frac{Rb_W}{Rb_{ID}}\right) \cdot \left(\frac{\mu_W}{\mu_{ID}}\right)$$

$$VI = \left(\frac{\lambda_V}{\lambda_{ID}}\right) \cdot \left(\frac{Rb_V}{Rb_{ID}}\right) \cdot \left(\frac{\mu_V}{\mu_{ID}}\right)$$

$$FI = \left(\frac{\lambda_{FD}}{\lambda_{ID}}\right) \cdot \left(\frac{Rb_{FD}}{Rb_{ID}}\right) \cdot \left(\frac{\mu_{FD}}{\mu_{ID}}\right)$$

and

$$NC_{ID} = \left[\frac{Rb_{FD}}{Rb_{ID}} NC\right] = \left[\frac{64 \text{ Kbps}}{1 \text{ Kbps}} 2243\right] = 1.435 \times 10^5 .$$

The iterative approach given in eq. 2.4 is used to calculate the probability distribution. The mean number of slots in use in terms of equivalent interactive users can now be calculated from

$$TM = \sum_{k=0}^{NC_{ID}} k \, FID_k . \tag{2.22}$$

This is the mean number of equivalent connection slots that are not available for interactive data in one second. The total mean number of slots that are available is then

$$TE = \frac{(NF)}{F_p} - TM . \tag{2.23}$$

The mean number of slots available for interactive data is the total number of slots in a

one second interval minus the number of connection slots that are occupied in this interval. Now that the mean number of available slot has been found, the Slotted Aloha delay can be calculated. It is assumed that the Slotted Aloha accessing scheme for interactive data is operating under steady-state conditions. The throughput can then be written as

$$S_{ID} = \frac{M_{ID}\rho_{ID}}{TE} = G_{ID}e^{-G_{ID}} \tag{2.24}$$

where M_{ID} is the number of interactive users on the ground and the mean expected number of users that are active at any time is $M_{ID}\rho_{ID}$ where

$$\rho_{ID} = \left(\frac{\lambda_{ID}}{\mu_{ID}}\right) = \frac{2.08\text{x}10^{-3}}{3.33\text{x}10^{-2}} = 6.24\text{x}10^{-2}.$$

Solving for G_{ID} numerically using the bisection method the interactive success probability can then be found as

$$Ps_{ID} = \frac{S_{ID}}{G_{ID}}. \tag{2.25}$$

The delay for interactive users can be approximated similarly to the reservation delay for connection users and is given by

$$D_{ID} = RT\sum_{i=0}^{\infty} iPs_{ID}\left(1-Ps_{ID}\right)^{i-1} = \frac{RT}{Ps_{ID}}. \tag{2.26}$$

2.5. Uplink Results

2.5.1. Call Blocking: Potential User Populations

The probability that a video call is blocked due to the frame being occupied is shown in Fig. 2.11 for three different traffic levels for both voice and file data. As can be seen the video call blocking probability is fairly independent of both the other two connection traffics. This is due to the large holding time and data rate of a video user as compared to both a voice and a file data user. On average there has to be a large population of voice and file data users to have an impact on the call blocking probability for video since they take up much less of the capacity of the frame. As shown, the blocking increases as the number of voice and file data users are increased. There is a relatively sharp increase for the a low number of video users but less of an increase as the number of video users increases. For example, with twenty video users the call blocking probability without any other types of users is around $6x10^{-3}$ while the blocking with $5x10^{3}$ voice and file data users respectively is $7x10^{-2}$, which increases to $1.5x10^{-2}$ for $1x10^{4}$ voice and file data users respectively. However, with 100 video users the blocking probability between the cases only changes from $8x10^{-1}$ to $3x10^{-3}$ due to the much greater traffic that a video user represents. In the case of the call blocking probability for voice users, as shown in Fig. 2.12, the blocking however is highly dependent on the number of video users. In the case with no video or file users up to $1.6x10^{4}$ voice users may be served with a blocking probability of $1x10^{-7}$. As the number of video users increases however from 10 to 20 the serviceable voice population drops to $8x10^{3}$ and $5x10^{3}$ users respectively. In addition to video both of these cases assume

a minimum population of $1x10^4$ file data users. The voice populations dependence on the number of video users is due to the large amount of traffic that a video user represents. Due to the long holding time and data rate of video users a minimum video population can drastically change the serviceable number of voice users. To satisfy a minimum of 20 video users and $7.5x10^3$ voice users however only $2x10^3$ file users can be accommodated as is shown in Fig. 2.13. As can be seen the call blocking probability for

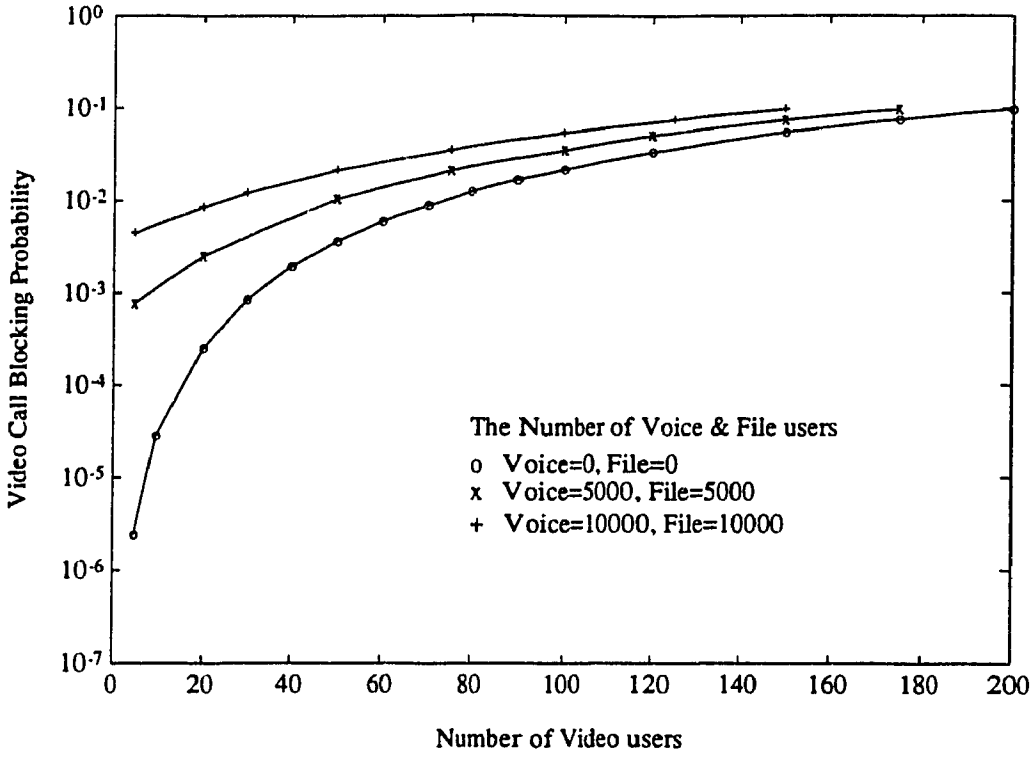


Fig. 2.11. Video call blocking.

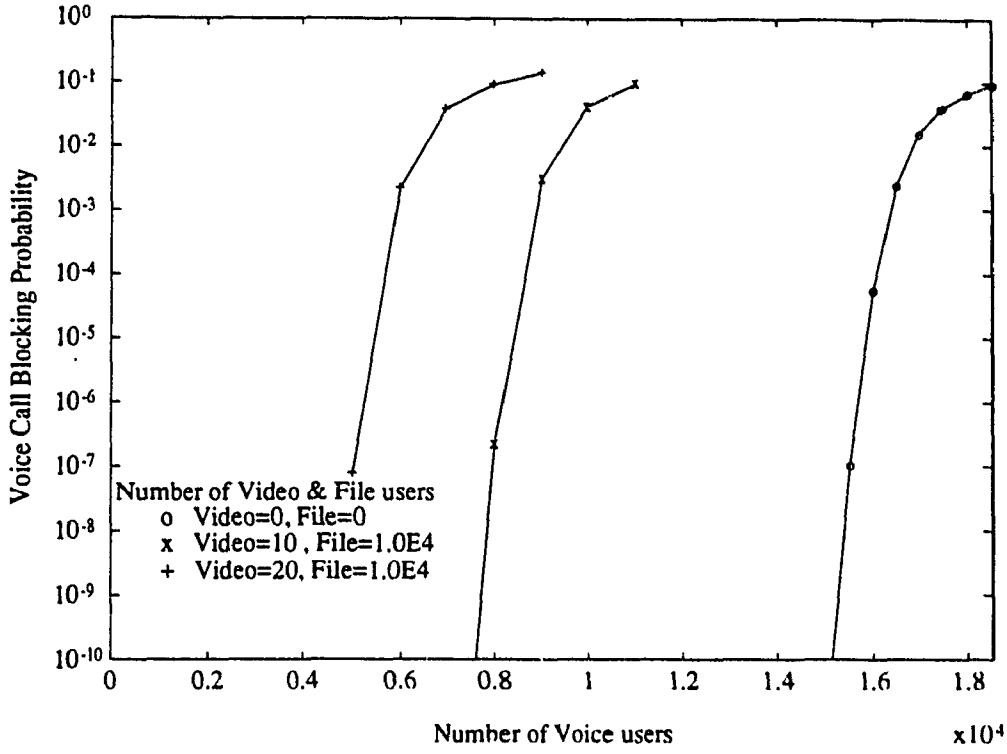


Fig. 2.12. Voice call blocking.

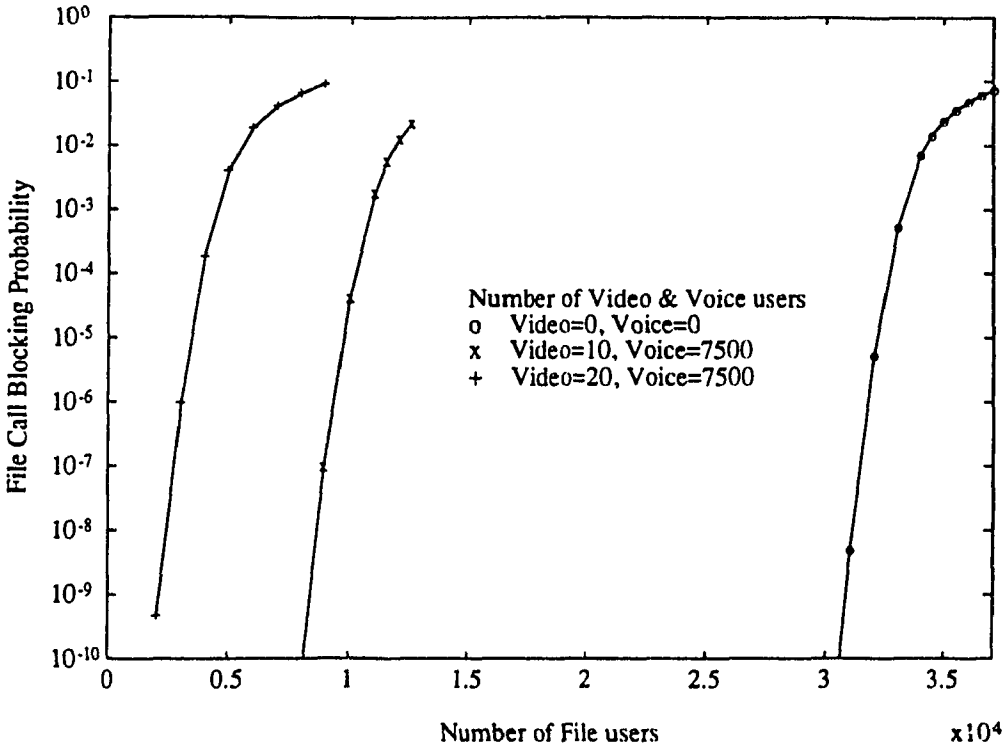


Fig. 2.13. File call blocking.

file users is even more dependent on the number of video and voice users. This dependence is mainly due to the high 20 minute service time that is assumed for file data. Even though an average file data user does not generate calls that frequently, when one does it uses a 64 Kbps channel for a relatively long time and hence it is susceptible to higher blocking from video and voice.

2.5.2. Effect of the Moving Boundary Scheme: Potential Interactive User Populations

To examine the effect of the moving boundary policy on the number of permissible interactive users, the Slotted Aloha delays for interactive users are plotted in Fig. 2.14 for different traffic conditions versus the number of users. Under heavy traffic loads, a minimum of $1x10^4$ interactive users can be served by the 100 slots which are dedicated to them per frame. This minimum population however can be substantially increased to $5x10^5$ users, for a delay of just 0.35 sec., if there is no other traffic on the frame. This large number is due to the low bit rate of interactive users, which is assumed to be 1 Kbps and the very high 150 Mbps rate of the uplink. Even though the frame will rarely be empty, this example shows the very large number of interactive users that can be accommodated with a moving frame policy. In this way the capacity that would be otherwise wasted is either used to support a larger interactive user population or to decrease the delays for the fixed $1x10^4$ user population.

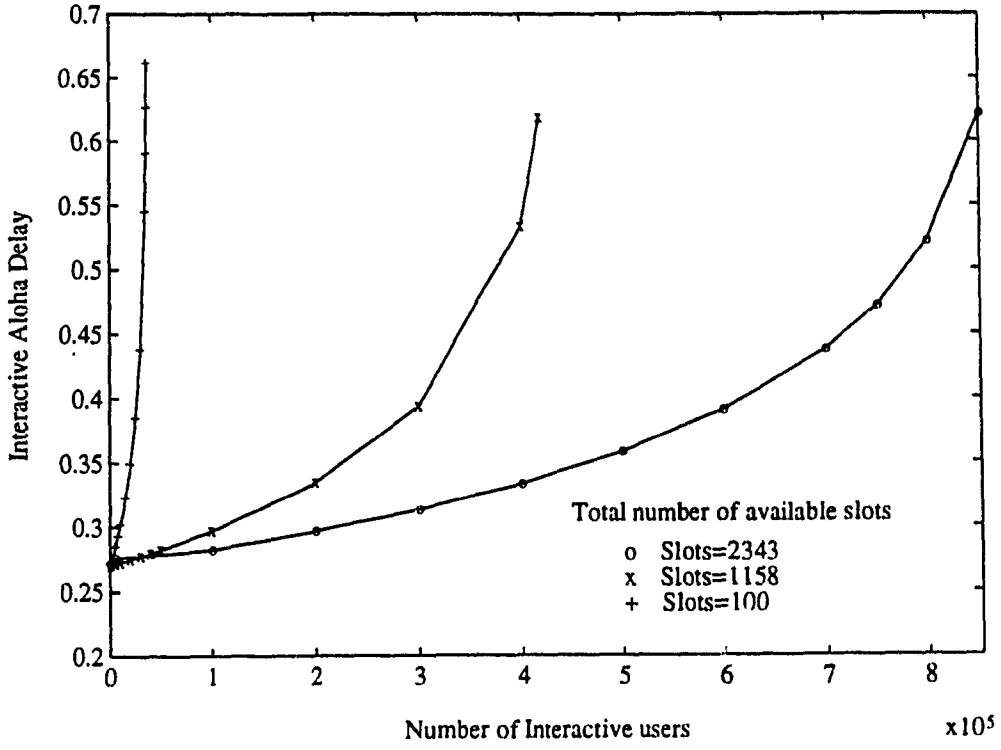


Fig. 2.14. Effect of the movable boundary scheme on interactive access.

2.5.3. Required Number of Call Reservation Minislots

In addition to the call blocking probability the number of reservation minislots for each of the connection services has to be determined in order to minimize the delay for each of the user populations. For the video case Fig. 2.15 shows that even one video reservation minislot per frame is sufficient for a population of 100 or more video users which are assumed to generate three calls per day on average. As was shown in Fig. 2.10, this number represents about the maximum permissible population of video users for the given frame capacity. In contrast, voice users which make on average three calls per hour

require a minimum of 25 and more likely 50 reservation minislots per frame, as shown in Fig. 2.16, for a population of $1x10^4$ users. With 50 minislots per frame the average delay approaches 0.3 sec for a connection with $1x10^4$ users. Using the same criteria only 5 minislots however are needed per frame for $1x10^4$ file data users, as shown in Fig. 2.17, to provide a delay of 0.3 seconds. This is due to the relatively low arrival rate of 5 calls per day for file data users.

Thus, by properly designing the frame along with a channel rate of 150 Mbps and by assuming a throughput of 0.368 for Slotted Aloha, it is seen that it is possible to accommodate up to 20 video users at 45 Mbps, $5x10^3$ voice users at 64 Kbps, $1x10^4$ file data users at 64 Kbps and $1x10^4$ interactive data users at 1 Kbps. Alternatively, the

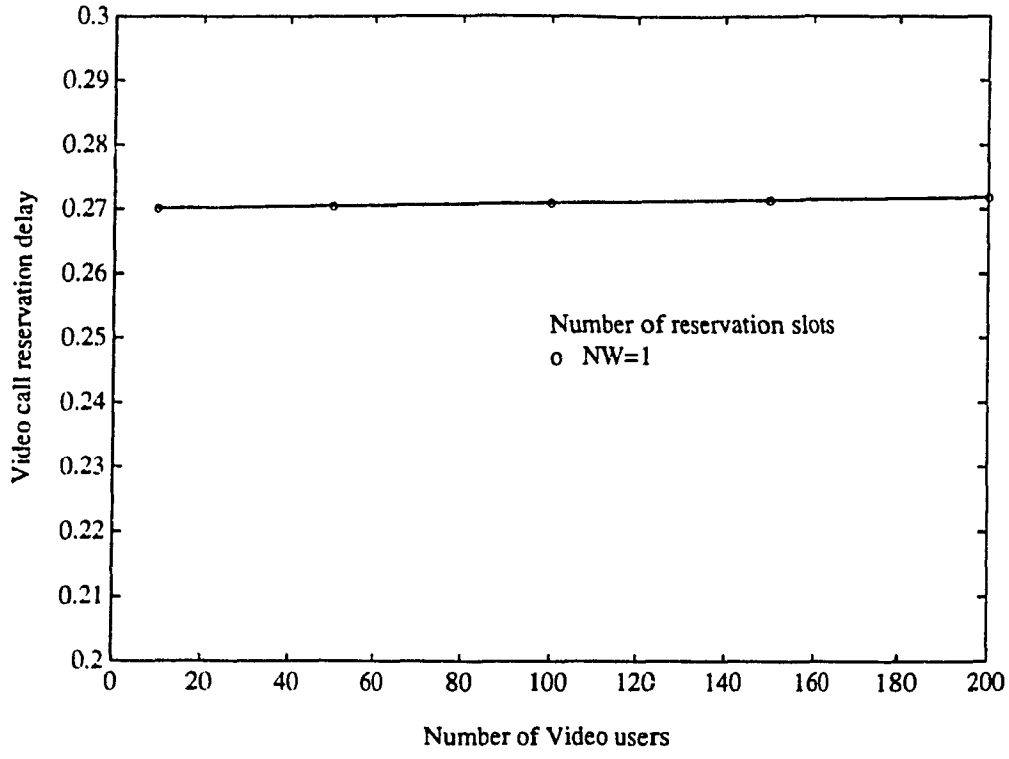


Fig. 2.15. Video reservation delay.

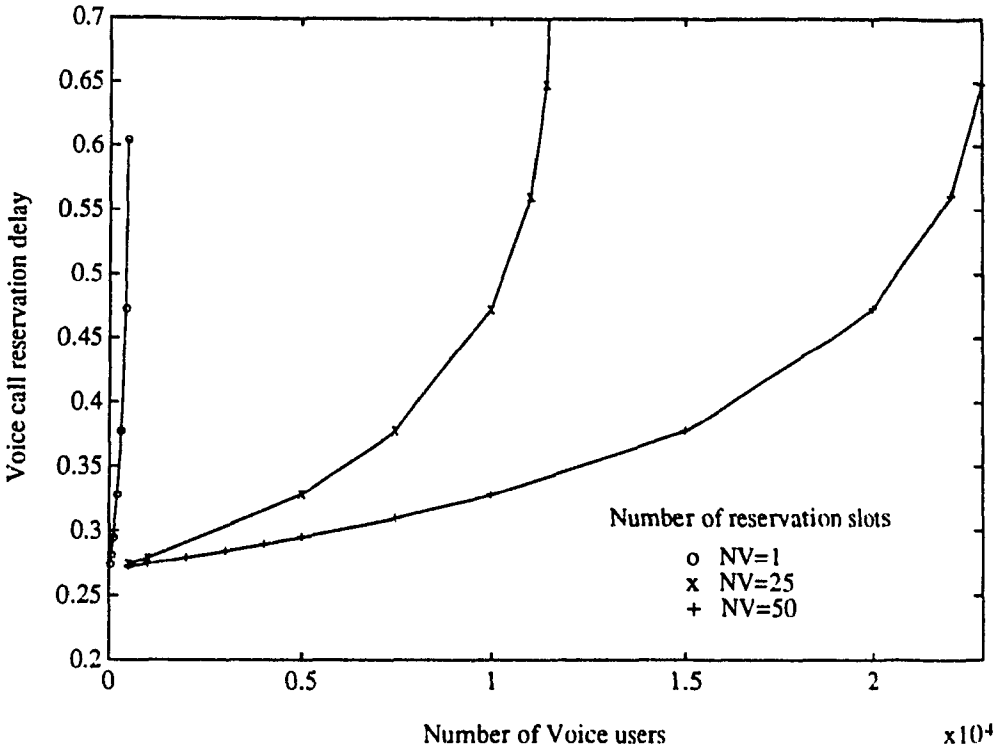


Fig. 2.16. Voice reservation delay.

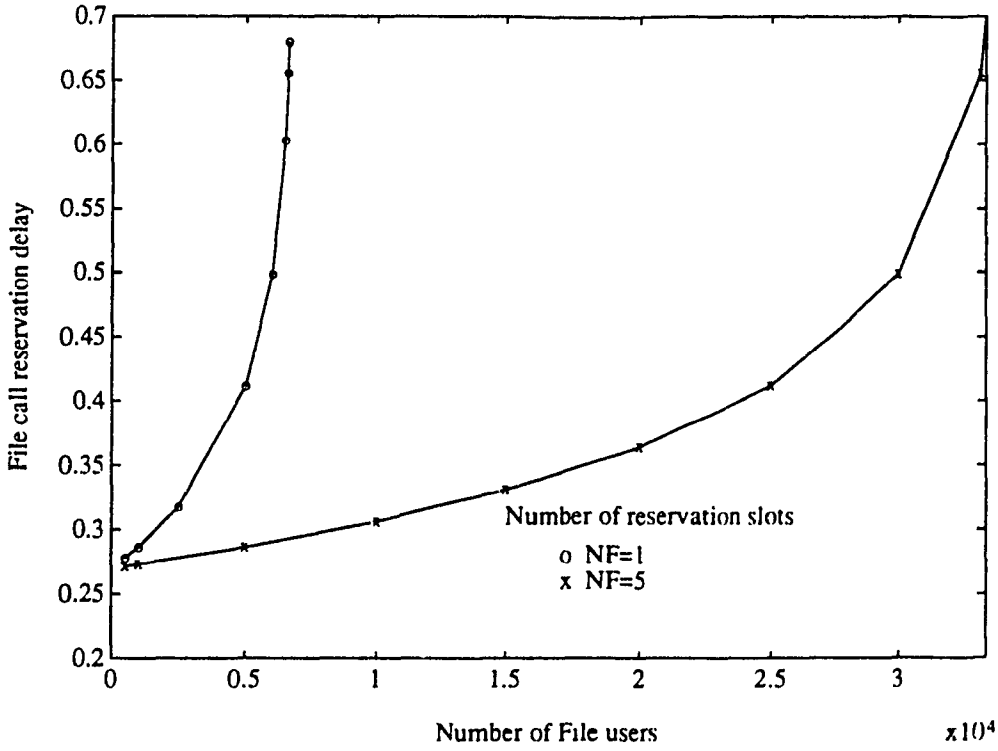


Fig. 2.17. File reservation delay

number of voice and file data users can be rearranged with the frame supporting $7.5x10^3$ voice users and $2x10^3$ file data users. By reducing the number of video users to 10 the extra capacity can be used to accommodate a population of up to $8x10^3$ voice users and $1x10^4$ file and interactive data users respectively. In addition, in times of low capacity the interactive users can use the moving boundary scheme to their advantage to reduce their blocking probability and hence their delay. At 50% capacity on the frame the moving boundary policy will allow a user population of up to $1x10^5$ with a mean delay of 0.3 seconds per packet. The overhead incurred in the frame to implement a reservation scheme is also seen to be minimal with only one minislot being required for video reservations at a delay of just 0.271 seconds, while 50 and 5 minislots will accommodate $1x10^4$ voice and file data users respectively at a mean connection delay of 0.3 seconds.

Chapter 3

On-Board Prioritized Baseband Switch with Analysis Under Uniform Traffic

In a future integrated services satellite system many different types of services such as video, voice, data transfers, and interactive services will have to be accommodated each with their own delay and rate requirements. In order to meet these requirements a future *NxN* on-board packet switch will have to be able to implement a priority structure, perform well under different traffic conditions and above all be fault-tolerant. To this end, the performance of internally blocking and nonblocking switches are examined in this chapter. A design based on a fault-tolerant Knockout switch is then presented as a candidate on-board switch. The proposed modified Knockout switch incorporates a priority scheme, as well as a head of line resolution algorithm to improve the service of the lowest priority users. Finally, the switch is analyzed with both input and output buffering under uniform traffic, in order to determine the appropriate system parameters to meet certain loss and delay criteria.

3.1. Packet Switches

Most packet switching architectures fall into one of two categories, those that are built around a Banyan switch and those that are hybrids of a fully connected switch. Banyan switches minimize the switch architecture, increasing in complexity only on the order of $\frac{N}{2}\log_2 N$, but at the same time maximize the blocking probability since they are

internally blocking and are able to switch at most one input to one output. Fully connected switches on the other hand minimize the blocking probability but maximize the complexity. They are internally non-blocking and have the potential to switch up to N packets to an output port but at the cost of maximizing the complexity which is on the order of N^2.

3.1.1. Banyan Switch

The throughput of Banyan switches is limited due to the fact that they exhibit quite severe blocking both in the internal fabric of the switch as well as at the output ports. A typical blocking scenario for a Banyan switch is shown in Fig. 3.1 where the switching fabric is made up of 2x2 switching modules. At each of the internal switches the packet is directed through the network depending on its output address and blocking occurs whenever two packets arrive simultaneously at an internal 2x2 switch and are destined to the same output port of that switch. It has been shown by Patel [27] and Kumar and Jump [28] that the performance of the Banyan switch under uniform traffic decreases rapidly as the switch size increases. As shown in Fig. 3.2 the decrease is most prominent in the first few stages and with 10 stages ($N=2^{10}=1024$) the throughput has already decreased to 0.25. Increasing the switch size further to $N=2^{25}=3.355\text{x}10^7$ the throughput as shown decreases to around 0.15. To reduce the packet loss internal buffering at each stage was proposed and analyzed by Jenq [29] for a buffer size of one. As shown in Fig. 3.3 the throughput converged to around 0.40 as N increased to 1024. Other analyses by Dias and Jump [30], and Theimer, Rathgeb and Huber [31,32,33] have

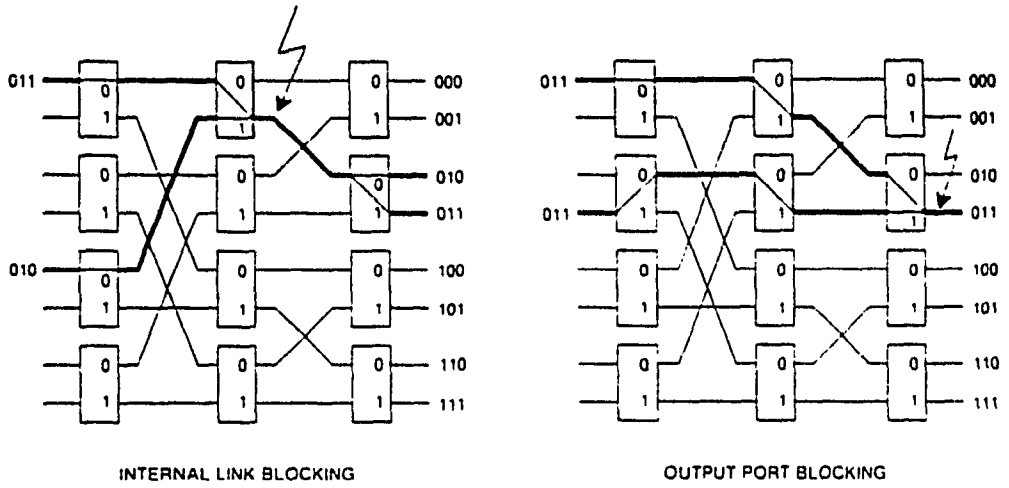


Fig. 3.1. Blocking scenario of a Banyan switch [26].

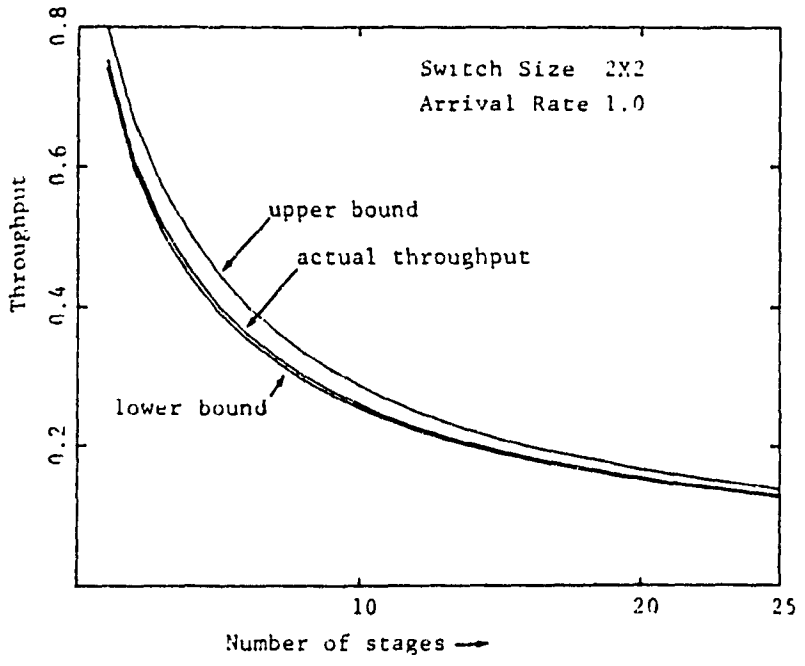


Fig. 3.2. Performance of a Banyan switch under uniform traffic [28].

examined larger buffers for switch sizes up to 64. It was shown that as the buffers are enlarged the throughput may be increased depending on the switch size and the number of buffers, as shown in Fig. 3.4 and Fig. 3.5. However, in all cases the main increase in throughput occurred between buffer sizes of one and two and tapered off as the buffer size increased so that for buffers larger than six there was almost no further improvement. In all the analyses independence was assumed between stages to reduce the complexity and iterative techniques were followed. Exact analysis of a buffered Banyan network is not feasible even in a uniform environment because it leads to a multidimensional state space with one dimension for every buffer of the network, which cannot be easily solved for as N increases. While the stages are correlated Kruskal, Snir and Weiss [34,35] have shown that the correlation decreases exponentially between the stages so that independence is a good approximation except at high loads near saturation.

3.1.1.1. Batcher-Banyan Switch

As an alternative to internal buffering it has been shown that a Batcher sorting network [36] may be used to sort incoming packets so that internal blocking in the Banyan switch may be eliminated altogether, as long as each input is destined to a unique output. The increase in complexity for a sorting network is on the order of $\frac{N}{4}\log_2 N\left(1+\log_2 N\right)$. Fig. 3.6a shows a Batcher-Banyan switch which presorts the packets in ascending order according to their address. The arrows in Fig. 3.6a represent the switch directions for the input with the higher address. A shuffle exchange is used at the input to the Banyan switch to select the packets so that no conflicts will occur within the

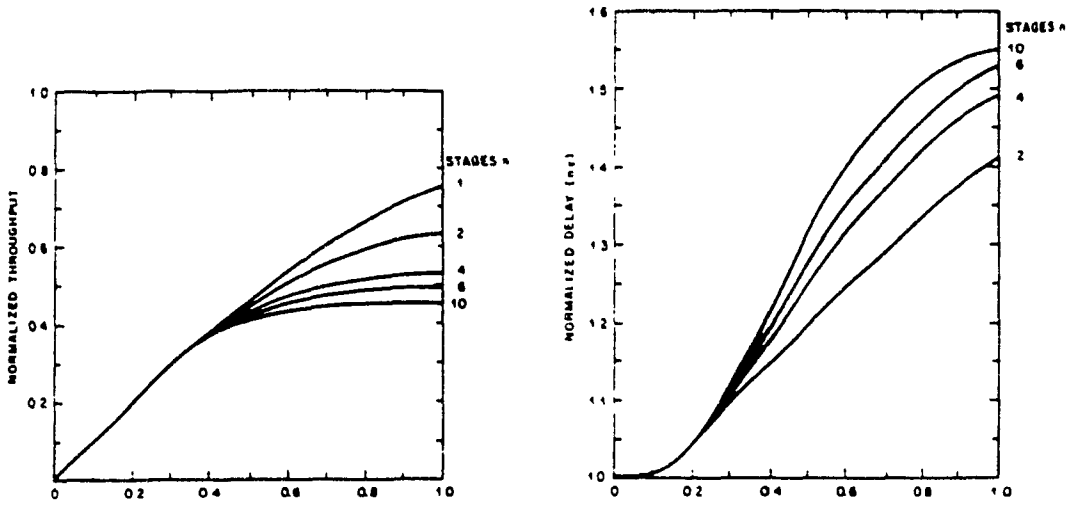


Fig. 3.3. Performance of a single buffered Banyan switch [29].

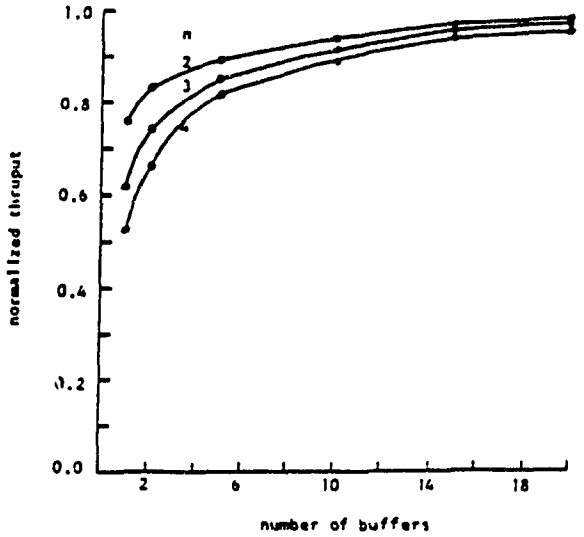


Fig. 3.4. Performance versus buffer size for a Banyan switch [30].

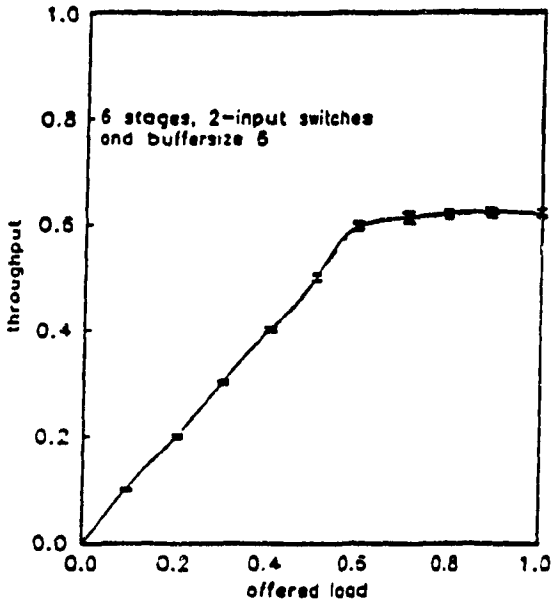
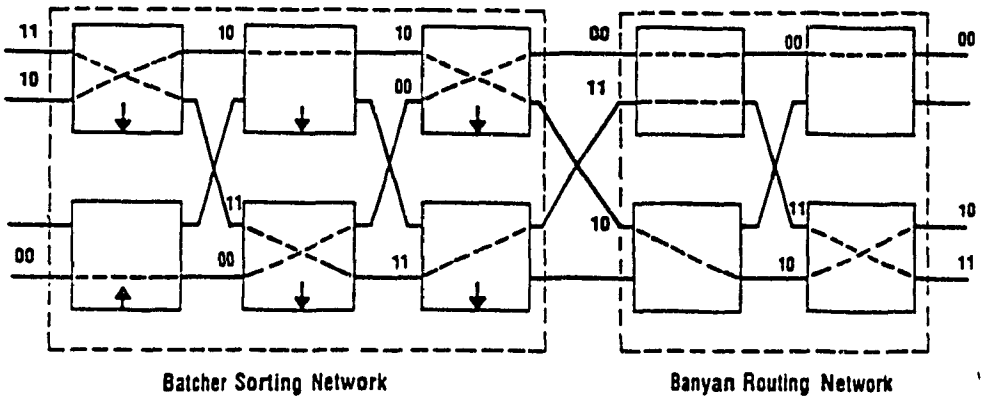


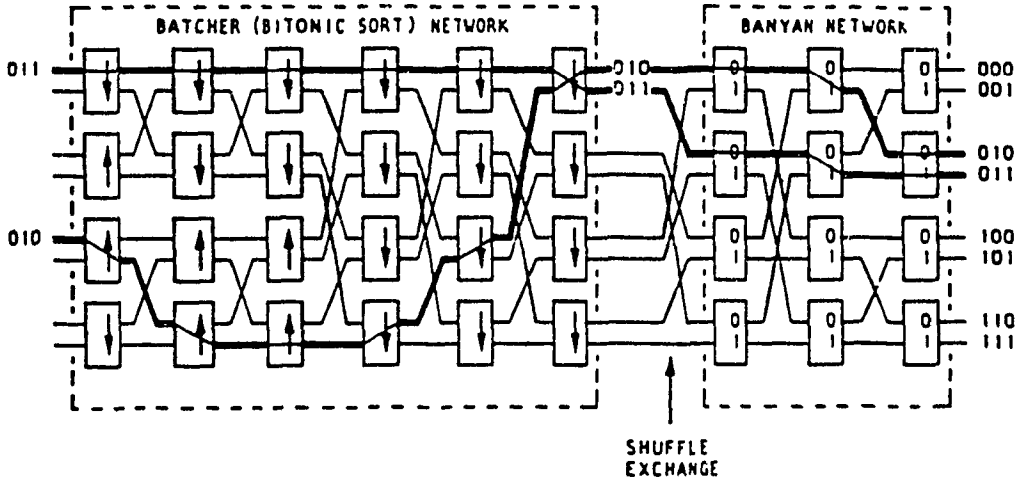
Fig. 3.5. Throughput for a buffer size of six with N=64 [33].

switch. The Batcher sorting network itself is nonblocking since packets with the same address are switched randomly at any node. The Batcher-Banyan switch for the previous blocking example is shown in Fig. 3.6b. Assuming that all the other inputs to the sorter have addresses which are higher than 010 or 011 the sorter sorts 010 as the lowest address followed by 011. This sorting now ensures that these two packets do not collide.

The Batcher-Banyan switch was first examined as a possible candidate for a future packet switch by Huang and Knauer [37] and later analyzed by Hui and Arthurs [38]. Hui and Arthurs analyzed the switch under uniform traffic with input queuing in order to limit the loss. In their design packets with the same destination that were blocked were



(a)



(b)

Fig. 3.6a,b. Batcher-Banyan switch [26].

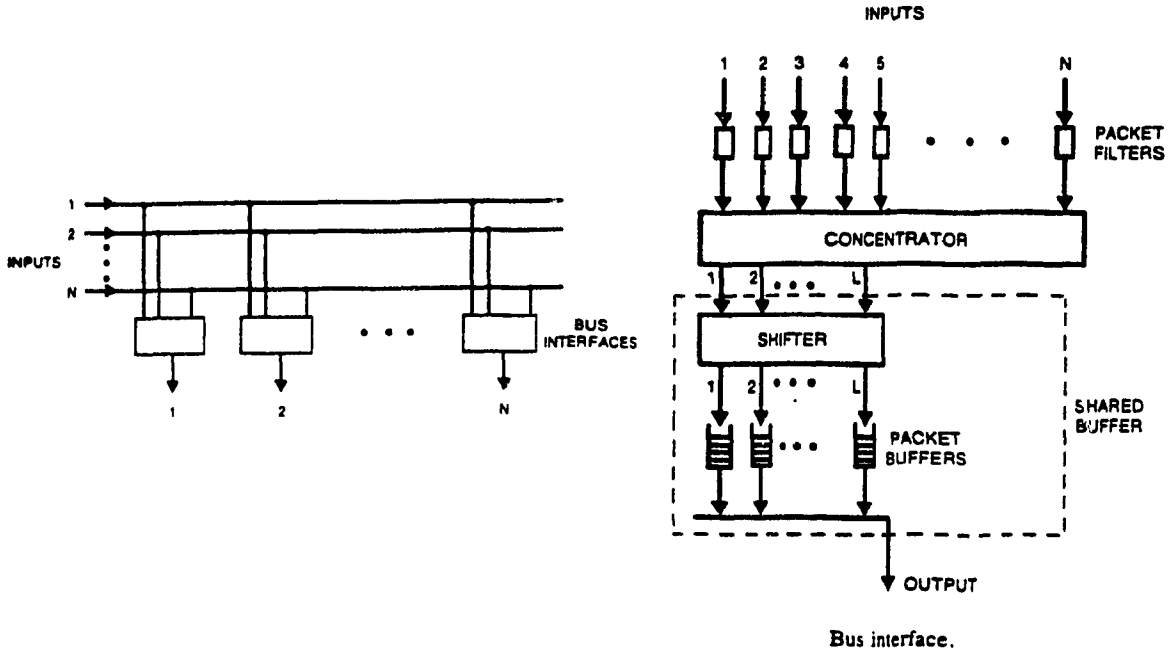
queued at the input. It was shown that with an infinite input queue as the switch size goes to infinity the throughput of the nonblocking Batcher-Banyan switch could be increased to 0.58.

While a single Batcher-Banyan switch may provide enough throughput for certain applications under uniform traffic conditions the blocking severely limits the throughput under nonuniform traffic conditions. To reduce this blocking the number of packets switched per output, L, may need to be increased. This can be accomplished by either speeding up the switch fabric L times or by using L parallel Batcher-Banyan planes. Normally however increasing the speed of the switch is not possible since the switch is already operating at its maximum speed, while increasing the number of planes further increases the switch complexity.

Another aspect of the performance of the Batcher-Banyan switch is fault tolerance. Because of the high interconnectivity of the Batcher-Banyan switch should one of the internal 2x2 switches fail more than one output may be affected depending on the location of the fault. With a multitude of possible scenarios the entire switch might have to be replicated in order to ensure that there is sufficient tolerance. In the next section the internally nonblocking Knockout switch is examined as a means of increasing the fault tolerance at the expense of increasing the complexity. However, it is shown that the complexity need not increase as N is increased in order to maintain an acceptable performance.

3.1.2. Knockout Switch

The Knockout switch was first proposed by Yeh, Hluchyj and Acampora [39] in order to increase the fault tolerance and to increase the switch speedup without speeding up the switch fabric. A unique property of the Knockout switch is that it is modular in design and can be configured as both an output blocking and output nonblocking switch in that $L \leq N$ packets may be switched to each of the output ports per time slot. The Knockout achieves this by using a parallel architecture. As seen in Fig. 3.7 each input is multiplexed to each output. Packet filters at each output select packets which are destined to their respective outputs. These packets are then passed through a concentrator circuit which selects a maximum of $L \leq N$ packets which are placed in parallel into L separate output buffers using a shifter circuit. The block diagram of an 8:4 Knockout concentrator is shown in Fig. 3.8. The concentrator implements an L round knockout tournament in which one packet successfully emerges in each round. A detailed description of the operation of the concentrator is differed until later in this chapter. By using a shifter and parallel output buffers no internal speedup of the switch fabric is needed. The separate output buffers for all purposes are equivalent to a single output buffer. The operation of a shifter is shown in Fig. 3.9 where it is assumed that L=8. Here it is seen that the shifter performs a circular shift of the eight concentrator outputs to the output buffers so that each of the output buffers is evenly filled. It is seen that in the first time slot five packets arrive and are put into buffers one through five. On the second time slot four more packets arrive and the shifter puts these into output buffers six through one.



Bus interface.

Fig. 3.7. Knockout switch [39].

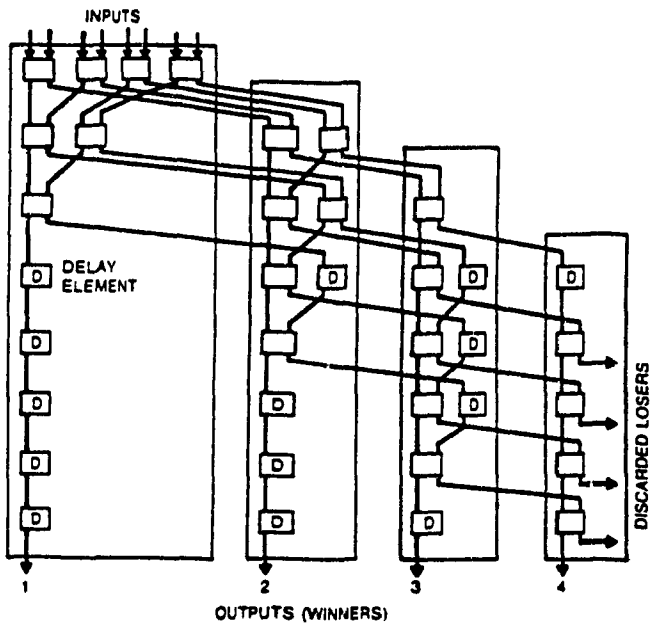


Fig. 3.8. Knockout concentrator [40].

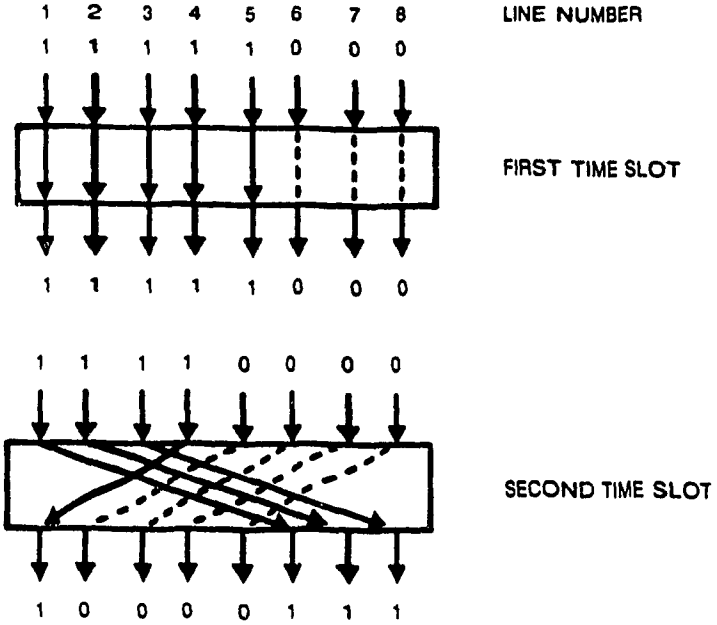


Fig. 3.9 Operation of the parallel shifter [39].

Yeh, Hluchyj and Acampora showed that only a minimal speedup is required to maintain acceptable losses as N goes to infinity. This is due to the rapid decrease in the binomial distribution of the number of packets arriving at an output port per time slot which is given by

$$P_k = \binom{N}{k}\left(\frac{p}{N}\right)^k\left(1-\frac{p}{N}\right)^{N-k} \qquad k=0,1,\ldots,N \tag{3.1}$$

where p is the rate at which packets arrive to a switch input port. The loss probability given a speedup of L can then be found using the ratio of the mean number of packets lost at an output port to the mean number of packets arriving at an output port.

$$\Pr[\text{packet loss}] = \frac{\sum_{k=L+1}^{N} (k-L)\binom{N}{k}\left(\frac{p}{N}\right)^k \left(1-\frac{p}{N}\right)^{N-k}}{p}. \tag{3.2}$$

As $N \to \infty$ this loss probability converges to

$$\Pr[\text{packet loss}] = \left[1-\frac{L}{p}\right]\left[1-\sum_{k=0}^{L}\frac{p^k e^{-p}}{k!}\right]+\frac{p^L e^{-p}}{L!}. \tag{3.3}$$

As seen in Figures 3.10 and 3.11, under uniform traffic conditions for L=8 the probability of blocking without buffering in a Knockout switch can be reduced to $1\text{x}10^{-6}$. It is also seen that under uniform traffic conditions that the blocking probability is independent of N. This is a general property under uniform traffic for any generic internally nonblocking switch. The independence between the loss and N under uniform traffic conditions allows the complexity at each output stage for a certain blo king probability to be minimized irrespective of the switch size. The Knockout switch's ability to switch up to L packets per output vastly improves the performance especially under nonuniform traffic conditions. Under nonuniform traffic conditions however, the number of arrivals to any one output is no longer independent of N and L has to be increased at the nonuniform outputs in order to satisfy the blocking criteria as the switch size increases.

In addition to increasing the throughput, the parallel structure of the Knockout switch also provides a means for easy fault tolerance. Since faults in a Knockout switch only effect one output the tolerance can be increased by adding an additional spare output circuit. If a fault is found in any one of the output circuits the spare circuit can then replace the faulty one by simply switching outputs.

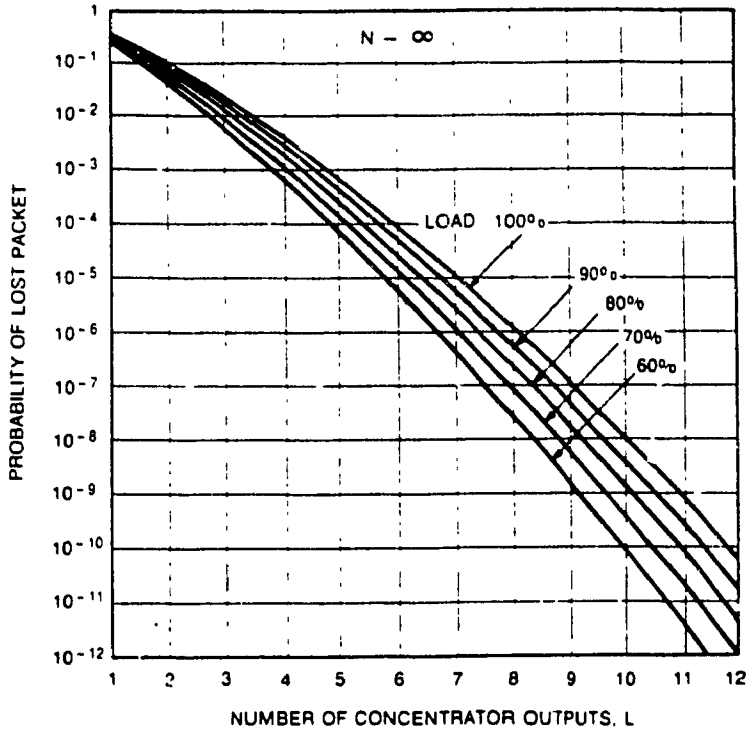


Fig. 3.10. Loss performance of the concentrator for different loads [39].

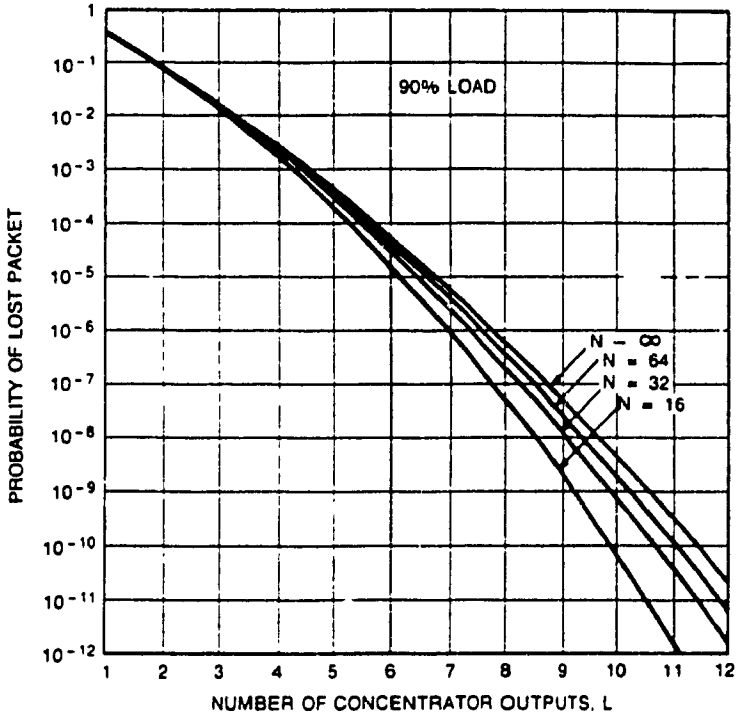


Fig. 3.11. Loss performance of the concentrator for increasing N [39].

The major disadvantage of the Knockout switch is the larger total number of basic switching elements needed in comparison to a Batcher-Banyan switch. For each output port the Knockout switch requires L input packet filters, which can be implemented using a minimum of five gates each. In addition, each of the L stages of a concentrator requires about N switching elements (2x2) for a total of NL switches per output. The shifter which can be built using an omega network also requires a minimum of $\left(\frac{L}{2}\log_2 L\right)$, 2x2 switching elements. Assuming that each 2x2 switch element can be implemented using a minimum of 16 gates the total number of gates then required for a Knockout switch with a speedup of L is $N\left[10N + 16\left(NL + \frac{L}{2}\log_2 L\right)\right]$ [39]. In comparison the total number of gates required for a Batcher-Banyan switch is given by $16\left(\frac{N}{2}\log_2 N + \frac{N}{4}\log_2 N\left(1 + \log_2 N\right)\right)$. Table 3.1. compares the total number of gates

Table. 3.1.
Number of Total Gates Required for an NxN Switch with L=1

N	Batcher-Banyan	Knockout
8	576	10048
32	5120	142336
128	35840	$2.203\text{x}10^6$
512	221184	$3.496\text{x}10^7$
1024	532480	$1.398\text{x}10^8$

required for each configuration with L=1. It is seen that the Knockout switch requires substantially more gates especially as N is increased. However, the implementation of a large Knockout switch in not limited by any means. Because of the modular nature of the design of the concentrator larger concentrators may be built out of smaller elements. Additional outputs can be accommodated by adding more plug-in modules into a daisy chain configuration, as shown in Fig. 3.12. In addition, the L+N pin requirement of the concentrator may be satisfied by building larger concentrators from smaller ones, as shown in Fig. 3.13. With current VLSI technology limiting the number of pinouts to around 256 and the number of gates to approximately 150 000 per chip a concentrator for N=128 can still be easily built [39]. It is assumed that such a switch based on gallium arsenide will be able to operate around 150 Mbps. It also has been shown by Eng [41], that an all optical Knockout switch can be implemented which would allow the rate to be increased to 2-4 Gbps.

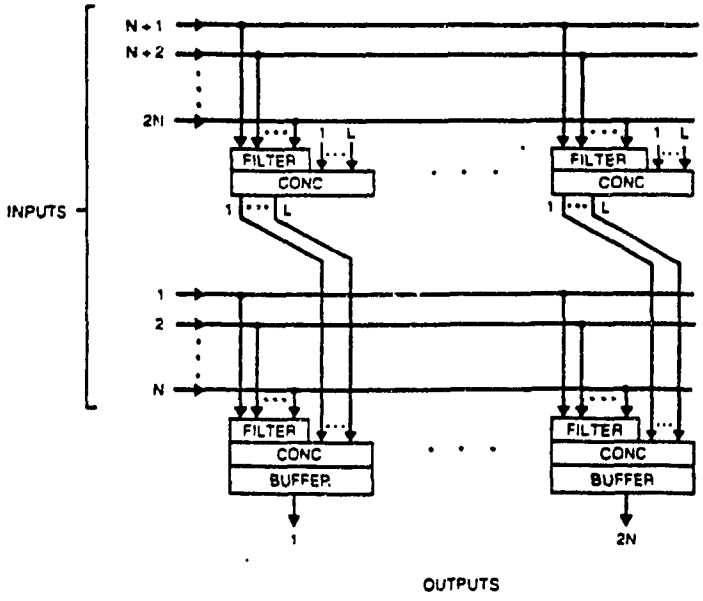


Fig. 3.12. Modular growth for a large Knockout switch [39].

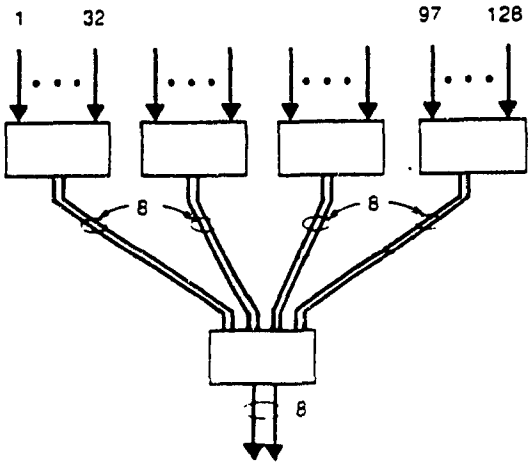


Fig. 3.13 Construction of larger concentrators [39].

3.2. Input and Output Buffering

3.2.1. Input or Output Buffering

Buffering at either the input or the output, as shown in Fig. 3.14, was first examined by Karol, Hluchyj and Morgan [42] as a means of increasing the performance of internally nonblocking switches with speedup. By introducing buffering at the input of a generic internally nonblocking switch it was shown that the output blocking probability may be reduced at the expense of the throughput. Since the traffic is uniformly distributed each of the input buffers are independent and can be modeled by equivalent queues. Modelling each of the input buffers by a Geom/G/1 queue it was shown that as N goes to infinity the throughput is limited to 0.586 due to head of line blocking, as shown in Fig. 3.15. Under head of line (HOL) blocking a blocked packet at the head of the input

queue blocks all the other packets inside the queue until it is successful. This blocking is due to output contention in the switch and limits the throughput of the input buffer. It is possible to analytically prove that the throughput is limited to 0.586 and the proof is repeated here.

Saturation Analysis:

Assuming that the input queues are saturated so that there are always packets waiting in each queue B_m^i can be defined as the number of packets at the heads of the input queues destined for an output i in the mth time slot which are not successful in the switch and A_m^i as the number of packets moving to the head of the input queues during the mth time slot destined to output i. B_m^i can be expressed as

$$B_m^i = \max\left(0, B_{m-1}^i + A_m^i - 1\right). \tag{3.4}$$

Although B_m^i does not represent the occupancy of any physical queue it has the form of an M/D/1 queue system. Under uniform traffic each input may go to any one output with probability $1/N$ so that

$$\Pr\left[A_m^i = k\right] = \binom{F_{m-1}}{k}\left(\frac{1}{N}\right)^k\left(1-\frac{1}{N}\right)^{F_{m-1}-k} \qquad k=0,1,\ldots,F_{m-1} \tag{3.5}$$

where

$$F_{m-1} \triangleq N - \sum_{i=1}^{N} B_{m-1}^i . \tag{3.6}$$

F_{m-1} represents the total number of packets transmitted through the switch during the

$(m\text{-}1)$st time slot which is the total number of queues with new packets at their heads.

$$F_{m-1} = \sum_{i=1}^{N} A_{m-1}^{i}. \tag{3.7}$$

Now the output utilization can be defined as p_o which is equal to $\frac{\overline{F}}{N}$. Intuitively it is seen and as well has been proven [42] that as N tends to infinity A^i, the steady state number of packets addressed to output i, becomes poisson at a rate p_o. This implies that B_m^i is equivalent to an M/D/1 queue. The mean steady state queue size for the M/D/1 queue can be written as

$$\overline{B^i} = \frac{p_o^2}{2(1-p_o)} \tag{3.8}$$

which is also equivalent to $1\text{-}p_o$ from the definition of $F_{m\text{-}1}$ as N goes to infinity. Equating the two it follows that $p_o = (2-\sqrt{2}) = 0.586$.

The input buffers are modelled as Geom/G/1 queues since only one packet is assumed to arrive with probability p at an input buffer per time slot. Furthermore, the arrivals are assumed to be independent from slot to slot so that the process is geometrically distributed. To find the service rate of the input buffers it is noticed that since the steady state arrivals at the heads of the input queues destined to output i become poisson as N goes to infinity, the service distribution is equivalent to the delay distribution of an M/D/1 queue. Packets that are destined to output i thus form a virtual M/D/1 queue. Using the first two moments of the delay distribution as the moments of the service distribution the mean delay of the Geom/G/1 queue can be found. A detailed analysis of

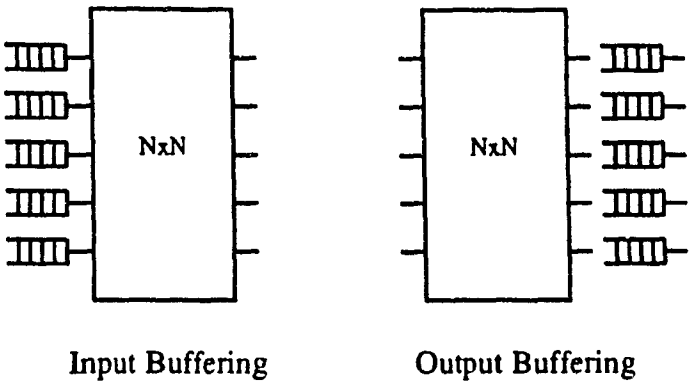


Fig. 3.14. Input and output buffering [42].

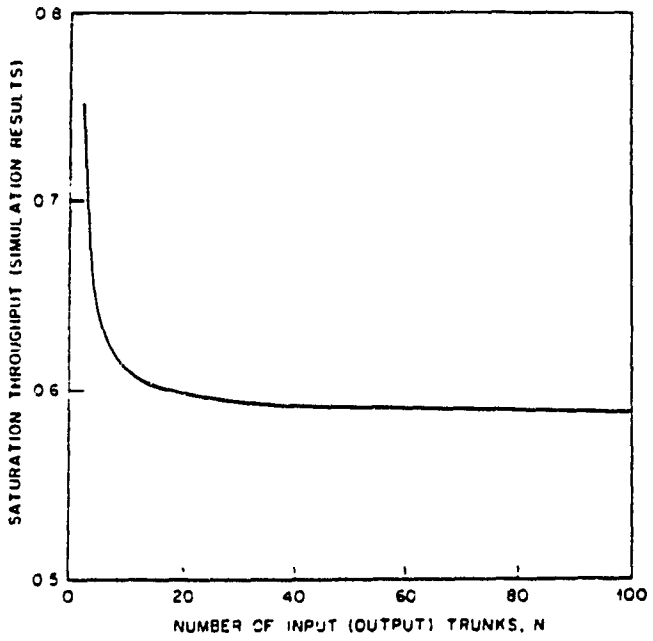


Fig. 3.15. Throughput convergence as N increases [43].

the input buffered case is examined later as a special case of the solution with an infinite input buffer and a finite output buffer b, when b=0.

Output buffering in contrast to input buffering was shown to provide the highest throughput but demands that L be large enough to ensure a low loss since the loss is only dependent on the switch. A constant service was assumed at the output buffer which was modeled by an M/D/1 queue system assuming that $N = \infty$. The process at the output port of the switch is a poisson process with mean p since the traffic is uniformly distributed. The difference in throughput between input (L=1) and output (L=N) buffering is shown in Fig. 3.16.

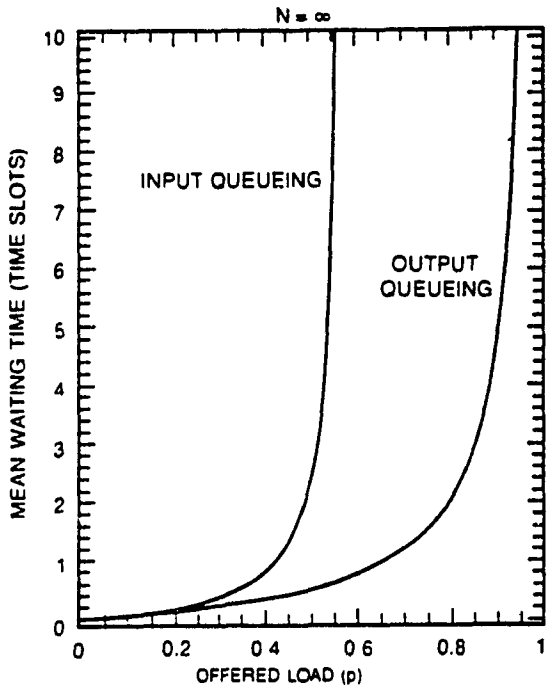


Fig. 3.16. Mean waiting time for input (L=1) and output (L=N) buffering [42].

3.2.2. Lossless Input and Output Buffering

Input and output buffering represent two extremes. In order to increase the throughput and at the same time maintain a feasible speedup both input and output buffering are needed. The performance for $1<L<N$ with both input and output buffering has recently been considered by Chen and Stern [44], Iliadis and Denzel [45] and Oie *et al.* [46]. Both Iliadis and Denzel and Oie *et al.* showed that for infinite input and output buffers the system delay is that of an M/D/1 queue Iliadis and Denzel also examined the case with infinite input buffers and finite output buffers of size b with a back pressure mechanism. With backpressure packets that overflow the output buffer are queued at the input. Iliadis and Denzel used the same Geom/G/1 model as in [42] but extended the service distribution to include the case with a finite lossless output buffer. The moments of the waiting time for a finite queue of size b with backpressure were found using the M/D/1 model shown in Fig. 3.17, where W_b is the delay of an arbitrary packet from the time that it appears at the head of its input port to the time that it is transmitted to its destined output port. The delay $D2$ includes W_b plus the delay in the queue until the packet is served. The mean of W_b can be found by using Little's formula and is the mean head of line queue length divided by the throughput.

$$\overline{W_b} = \frac{\overline{Q_b}}{p}. \tag{3.9}$$

The mean head of line queue length is given by

$$\overline{Q_b} = \sum_{i=1}^{\infty} i q_{b+i} = \sum_{i=b+1}^{\infty} (i-b) q_i = \overline{Q} - b(1-q_0) + \sum_{i=1}^{b-1} (b-i) q_i \tag{3.10}$$

where $\overline{Q} = \sum_{i=1}^{\infty} i\, q_i$ and q_i is the distribution of the discrete time M/D/1 queue (Fig. 3.17) given in Appendix A. Similarly the second order moment can be shown to be equal to

$$\overline{W_b^2} = \frac{\overline{Q_b^2}}{p} = \frac{\overline{Q^2} - 2b\overline{Q} + b^2(1-q_0) - \sum_{i=1}^{b-1}(b-i)^2 q_i}{p}. \tag{3.11}$$

The first and second moments of the M/D/1 waiting time can then be used as the moments of the service time to find the waiting time of the input Geom/G/1 queue. The waiting time of the Geom/G/1 queue has been shown in [24] to be equal to

$$\overline{W_i} = \frac{p\left(\overline{T^2} - \overline{T}\right)}{2\left(1 - p\overline{T}\right)} \tag{3.12}$$

where $\overline{T} = \overline{W_b} + 1$ and $\overline{T^2} = \overline{W_b^2} + 2\overline{W_b} + 1$ are the two moments of the service distribution. The service time of a packet consists of the waiting time W_b until it is selected for transfer plus one slot for transmission. The total delay, D, of the system

$$D = W_i + W_s \tag{3.13}$$

is then the waiting time in the input queue, W_i, plus W_s, which is the waiting time at the head of line and the time spent waiting in the output queue. Since no packets are lost at the output the mean waiting time found using Little's formula is the mean queue size of the equivalent output system, Fig. 3.17, divided by the throughput.

$$\overline{W_s} = \frac{\overline{Q}}{p}. \tag{3.14}$$

The results are shown in Fig. 3.18, where it is seen that for $b=\infty$ that the system is

equivalent to an M/D/1 queue as $N \rightarrow \infty$ since for this case $L=N$. Substituting $b=\infty$ into eq. (3.9) it is seen that $\overline{W_b}=0$ so that $\overline{D}=\overline{W_s}=\frac{\overline{Q}}{p}$. The case with $b=0$ on the other hand is equivalent to input buffering only. Substituting $b=0$ it is seen that $\overline{W_b}=\overline{W_s}=\frac{\overline{Q}}{p}$ and the delay is found from eq. (3.12) and eq. (3.13). It is seen in Fig. 3.18 that the throughput is limited to 0.586 for this case.

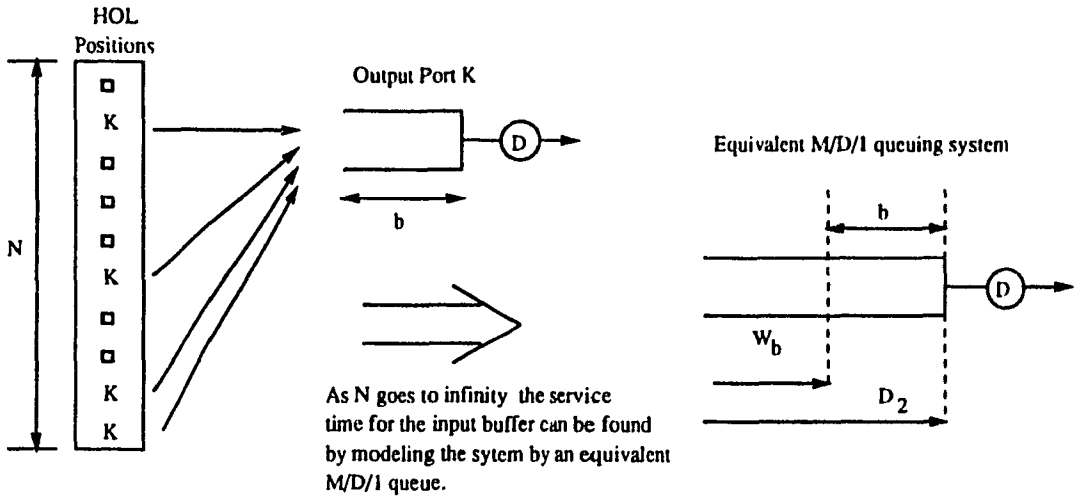


Fig. 3.17. Equivalent M/D/1 queuing system [45].

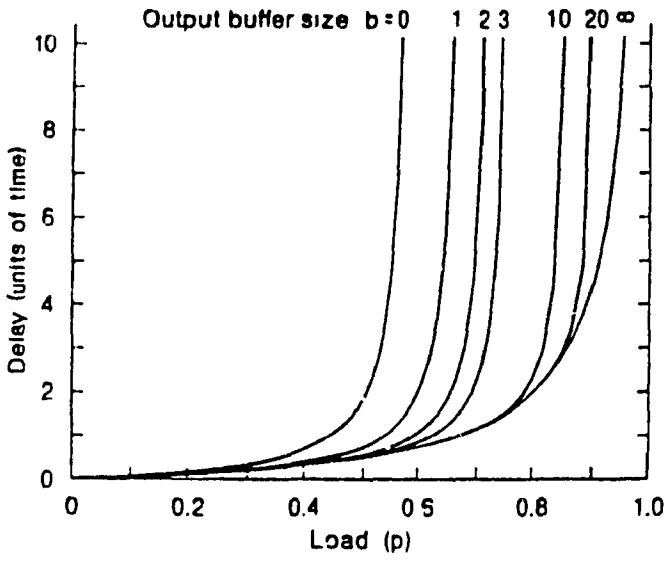


Fig. 3.18. Infinite input and finite output (b) buffering [45].

3.2.3. Finite Input and Output Buffering

Chen and Stern examined both infinite and finite input and output buffering. In their paper they showed that the improvement in the input buffer performance increases very little as the speedup is increased above three due to the decreasing blocking probability, as shown in Fig. 3.19. They also showed that the loss performance is highly dependent on the correct input and output buffer sizes and that the optimum choice under uniform traffic occurs when the input and output buffers are roughly the same size. As seen in Fig. 3.20a for L=4 and a load of 0.4 only moderate input (K_i) and output buffer (K_o) sizes are needed to ensure an acceptable loss. Correct buffer sizes are especially critical as the load increases as shown in Fig. 3.20b, where $K=K_i+K_o$. Chen and Stern in

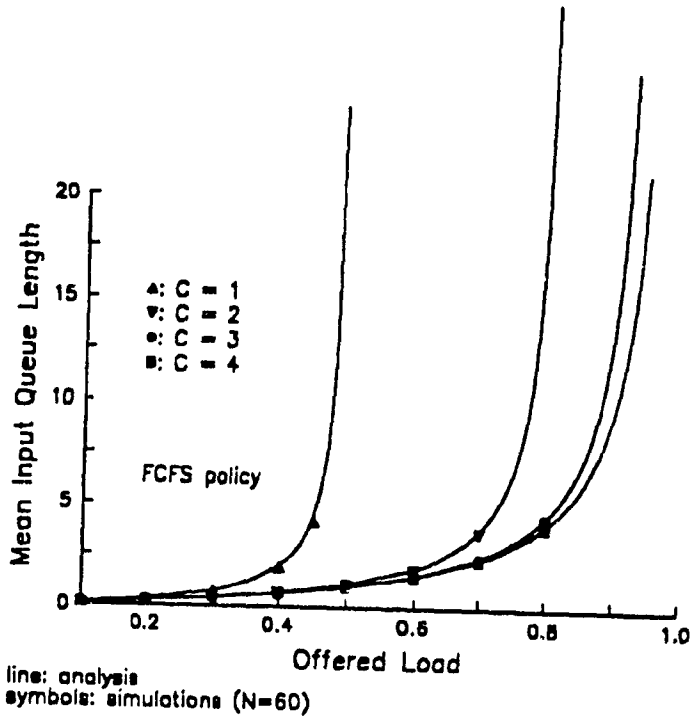
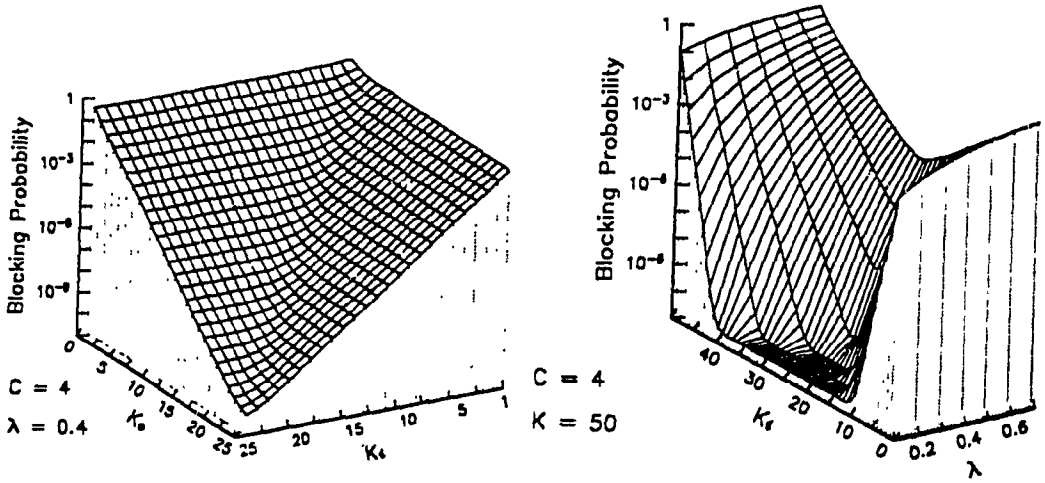


Fig. 3.19. Input buffer length as the speedup L=C increases [44].



(a) (b)

Fig. 3.20a,b. Performance with input and output buffering [44].

their analysis modelled the input buffers as M/G/1 queues while the output buffers were modelled as M/M/1 queues. Unlike Iliadis and Denzel's paper independence was assumed between the input and the output. As long as the output buffer loss is low however this is a valid approximation. In addition, because of the independence assumption the service distribution for the M/G/1 queue was found from the waiting time of an M/M/L virtual queue. Since there is no backpressure mechanism an M/M/L model is justified since up to L packets may be switched regardless of the output buffer state. The first two moments of an M/M/L queue are given by

$$\begin{aligned} \overline{W_s} &= \frac{u(\lambda/\mu)^L}{(L-1)!\,(L\mu-\lambda)^2} q_o + \frac{1}{\mu} \\ \overline{W_s^2} &= \frac{2(\lambda/\mu)^L}{(L-1)!\,(L\mu-\lambda)^2}\left(\frac{\mu}{L\mu-\lambda}+1\right) q_o + \frac{2}{\mu^2} \end{aligned} \tag{3.15}$$

where

$$q_o = \left[\sum_{n=0}^{L-1} \frac{(\lambda/\mu)^n}{n!} + \frac{L(\lambda/\mu)^L}{L!(L-\lambda/\mu)}\right]^{-1}$$

is the steady state probability of no packets in the system given by Erlang's second formula, and where λ=arrival rate and μ=mean packet length. The mean waiting time of the M/G/1 queue is then

$$\overline{W_t} = \overline{W_s} + \frac{\lambda \overline{W_s^2}}{2\left(1-\lambda \overline{W_s}\right)}. \tag{3.16}$$

For the finite buffer case Chen and Stern numerically approximated the input

buffer blocking probability, P_{bi}, from the tail of the distribution of the queue length N_i, ie. $P_{bi} = \text{Prob}(N_i > K_i)$. The blocking probability was found using an existing algorithm [47] and the z-transform of the M/G/1 queue length which is given in [24].

At the output the finite output buffers were modeled as M/M/1/K queues with an arrival rate of $\lambda(1-P_{bi})$ and a buffer size of K_o. The output blocking probability is then

$$P_{bo} = \frac{(1-p_o)p_o^{K_o}}{1-p_o^{K_o+1}} \tag{3.17}$$

where

$$p_o = \frac{\lambda(1-P_{bi})}{u}. \tag{3.18}$$

The blocking probability in Fig. 3.20 is the probability that a packet is blocked somewhere in the system and is given by

$$P_B = 1-(1-P_{bi})(1-P_{bo}). \tag{3.19}$$

3.3. The Modified Knockout Switch: Priority and HLR

The combination of performance, fault tolerance and modular design makes the Knockout switch a good candidate for a first generation on-board VSAT satellite switch which will most likely require moderate switch sizes of $1024 \geq N \geq 32$ to start with depending on the system bandwidth. However before the Knockout switch can be used in an integrated environment it has to be duly modified to implement a priority structure. As well, the switch will have to be analyzed with both input and output buffering in order

to minimize the loss probability which is extremely important in a satellite application since retransmission due to the large round trip propagation delay is virtually impossible. In addition selecting these buffer sizes with a priority structure is crucial to any future design. In the following sections it will be shown that with only minor changes to the original Knockout switch a priority structure can be easily implemented. Also, a head of line resolution (HLR) algorithm will be examined for use with the priority Knockout switch in order to increase the throughput of the lowest priority users.

3.3.1. Priority Scheme

A priority scheme can be implemented in a Knockout switch by using the knockout tournament principle of the concentrator. The structure that is implemented is shown in Fig. 3.21 where video has the highest priority followed by voice, file data and finally by interactive data. Since the Knockout switch concentrator can switch a maximum of L packets to any one output during a time slot if more than L packets are destined to an output some packets will be blocked. Blocked packets are assumed to be queued at the input. The priority algorithm shown in Fig. 3.21 proceeds through the priority structure assuming there are A video packets, B voice packets, C file packets and D interactive packets destined to the same output. If $L<A$ then $(A-L)$ video packets will be blocked along with all the other services since video is the highest priority. Next, if $A<L$ and $A+B>L$ then $A+B-L$ voice packet will be blocked. On the other hand if $L \leq A+B$ then neither video nor voice will be blocked. Continuing through the algorithm the conditions under which file and interactive data are blocked are similarly shown.

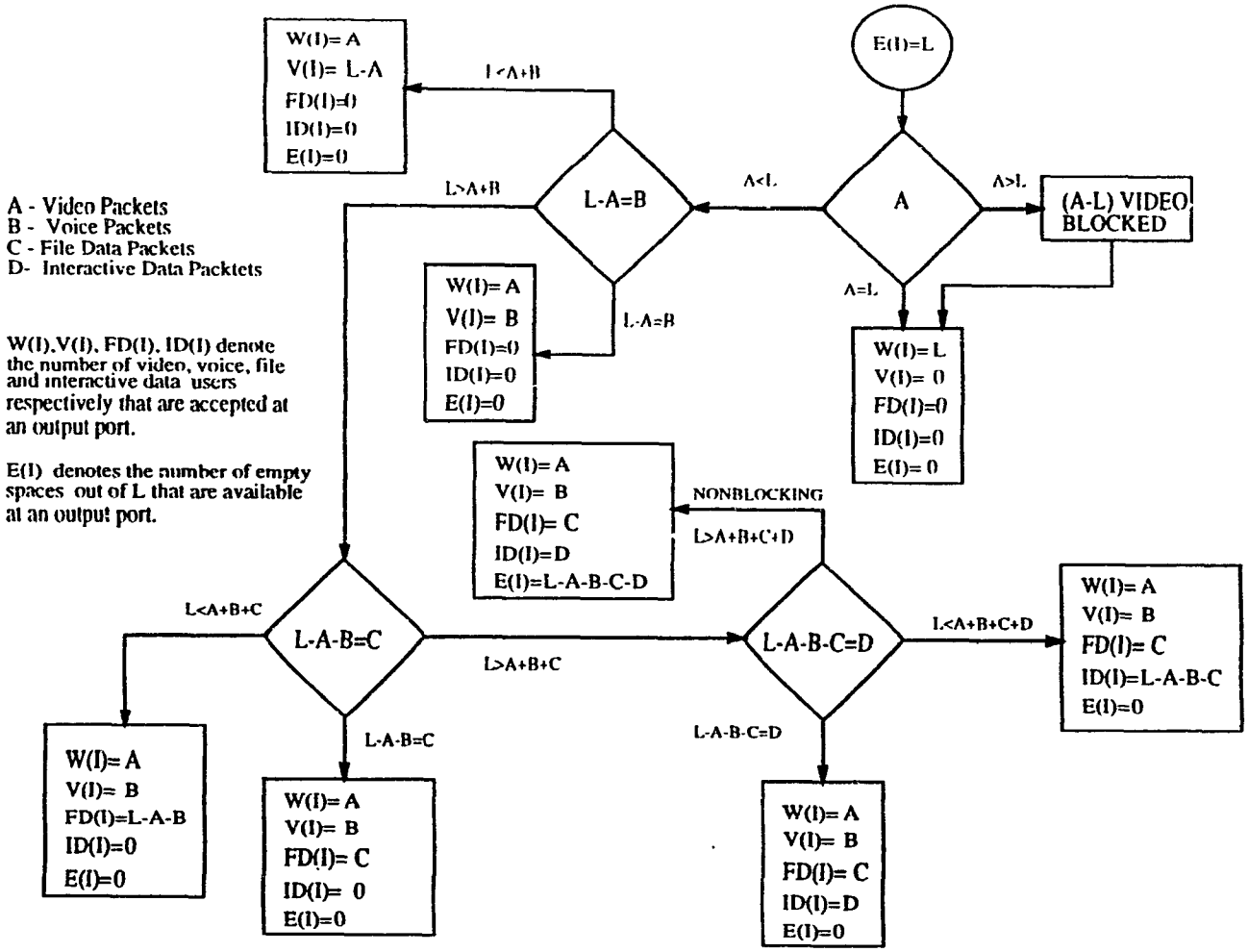


Fig. 3.21. Priority algorithm.

3.3.2. Head of Line Resolution

Under head of line blocking, packets in the input buffers may experience excessive delays because the packet in the server is continuously being blocked. HOL blocking is especially severe for the low priority packets under heavy loads and under nonuniform traffic conditions. In these cases it is necessary to alleviate the HOL problem in order to maintain an acceptable loss or delay at the input buffer. One method of coping with HOL blocking is to increase the switch speed, namely the speedup L of the Knockout switch. By increasing L the switch throughput could be further increased under both uniform and nonuniform traffics. However, increasing L increases the switch complexity especially as N is increased. Speedup could be further complemented by using an incrementing priority scheme to ensure a low variance in the delay. In such a scheme the priority of the packets which are continuously delayed would be incremented in order to ensure that they are successful.

Another means of increasing the switch throughput is through the use of input port expansion [48]. With input port expansion, as shown in Fig. 3.22, the number of inputs to the switch is increased so that more than one packet may compete from each input per time slot. This method however also increases the complexity of the switch itself in order to increase the throughput. The increase in complexity is a major concern with this method especially as N and the input port expansion, S, are increased. Besides the added complexity to the switch, under nonuniform traffic it is still possible that both packets at an input will be blocked for a considerable time. Therefore, while this method reduces the probability of HOL blocking it is still a distinct possibility.

Instead of altering the switch to improve the throughput another alternative, called look-ahead contention resolution [38, 42], proposes that w cycles of contention resolution be implemented in each time slot. This method proposes that upon blocking of the initial packet another packet in the buffer would compete in its place in a second cycle and then upon blocking of this packet still another packet would be chosen to compete in a third cycle etc., up until a possible wth cycle. In this scheme, as for speedup, only one packet at maximum is cleared from any one input. However, unlike speedup and input port expansion this scheme provides information as to which inputs were blocked and allows a decision to be taken in selecting the packets for the next cycle. By choosing the next cycle packets correctly the blocking probability may be minimized and provide a means to reduce the HOL blocking probability under both uniform and nonuniform traffic conditions. The price that is paid in implementing this scheme in most switches is that the switching speed has to be physically increased by w to accommodate the arbitration cycles. As will be shown however, a look-ahead HLR scheme can be successfully implemented in a Knockout switch architecture without speeding up the operation of the switch.

To ensure a low HOL blocking both speedup and a two round look-ahead contention resolution algorithm are incorporated in the modified Knockout switch. In this way the blocking is kept low to start with and if a packet should be blocked the look-ahead algorithm will increase the chance that another packet will be successful. The look-ahead HLR algorithm is assumed to be only implemented for the two lowest priority services, namely file and interactive data, which suffer at the expense of video and voice.

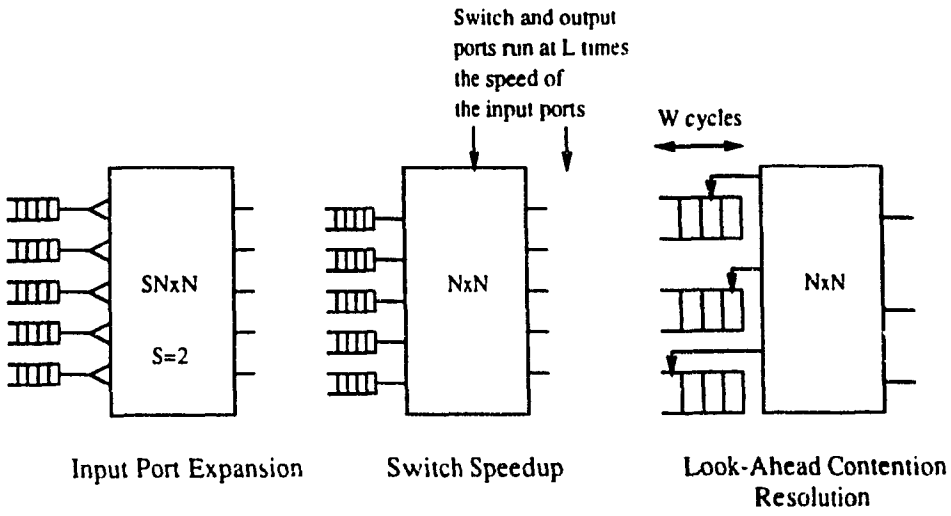


Fig. 3.22. Head of line resolution schemes [48].

In this way the look-ahead HLR algorithm is used to guarantee a minimum service for these users. Upon blocking of a file or an interactive packet the HLR algorithm allows another file or interactive packet (termed the sister packet), from the same input buffer, to compete for a second time in order to try to increase the throughput. The sister packets are assumed to be chosen randomly in the analysis so that there is no processing time associated in selecting the packet. In addition, the same priority structure is maintained and the file and interactive sister packets compete for any empty spaces at the switch outputs, which were not used in the first round of the competition. If a sister packet is also blocked then blocking is called and no further action is taken. If however the sister packet is successful then the throughput is increased.

3.3.3. Implementing Priority and HLR in a Knockout Switch

The modified Knockout system is shown in Fig. 3.23. In order to be able to implement the look-ahead HLR technique in a Knockout switch the server controller has to be informed of an unsuccessful packet within a very small fraction of the slot period. If the overhead of the acknowledgment becomes too large the delays incurred would cut into the frame period with the effect of reducing the amount of information that the frame can carry. This would be unacceptable since the time it takes to inform the server controller that a collision has taken place could be better spent switching another packet. To achieve a very small first round acknowledgement time the server controller only sends the control part of the packet in the first round to compete at the various outputs. The control part of the packet, as shown in Fig. 3.24, contains the address of the output port destination, information about the class of the service packet and the round number in which it is competing. Since the control information in the packet is only a few bits in length the processing time of the concentrator to determine which packets are unsuccessful in the first round of competition is very small compared to the length of a slot. This allows enough time in a slot for the second round packets to compete without speeding up the operation of the concentrator. The results of the first round of competition from the concentrator are then fed back to the server controller using the bidirectional bus interface, as shown in Fig. 3.23. If a file or interactive data packet is blocked the server controller then lets the sister packet compete in the second round. To distinguish second round packets from first round packets the round number bit in the control part of the frame is set to one. In the second round the complete packets from both the successful

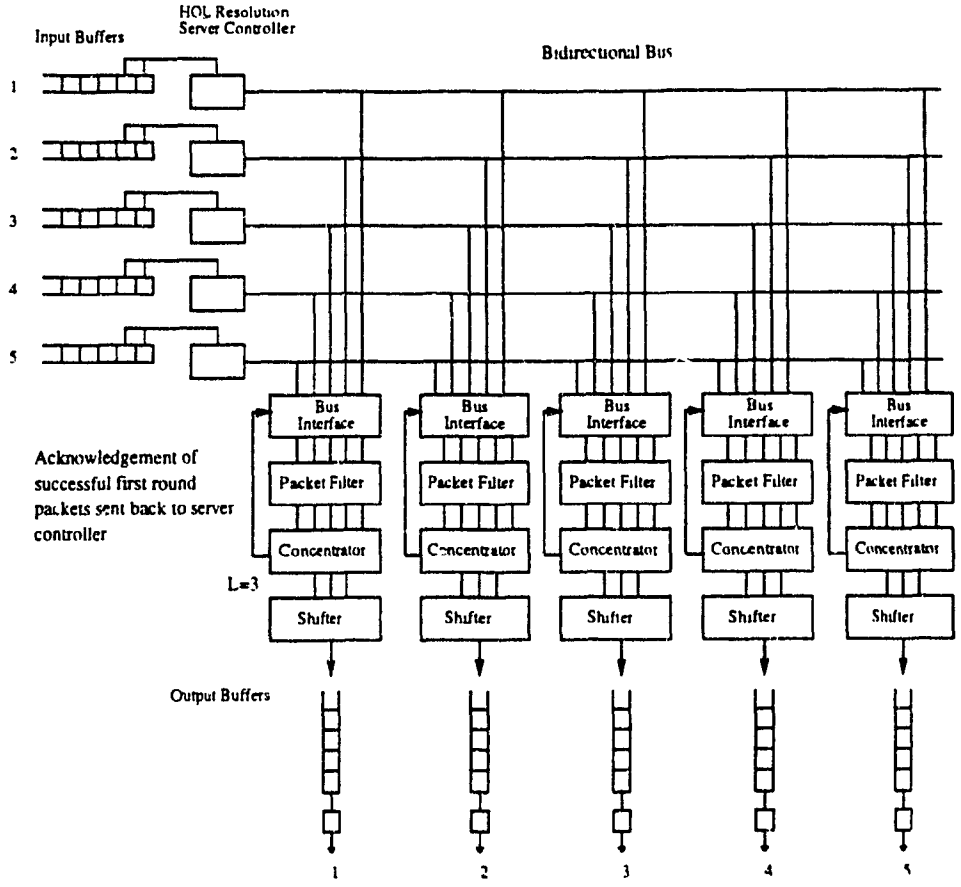


Fig. 3.23. Modified Knockout switch.

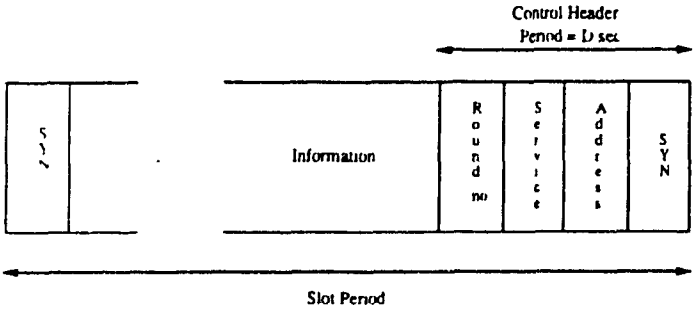


Fig. 3.24. Modified control packet header.

first round and the second round sister packets are sent to their respective outputs. At the concentrator first round packets have priority over second round packets along with the normal priority structure. In this way first round packets are assured of a higher success and the second round packets compete amongst themselves for the remaining available positions.

The operation of the concentrator at one of the outputs is shown in Fig. 3.25 for N=8, L=5 and where W, V, FD, and ID denote the video, voice, file and interactive data packets respectively. Both first and second round packets are represented with the superscript denoting the round number and the subscript denoting the input port number of the packet. In addition, input ports that do not have a packet to send to the output port are denoted by 0. The structure of the concentrator follows that of a knockout tournament in which all the packets compete to produce one winner in the first tournament and with the losers competing amongst themselves in a second, third tournament etc. up until the Lth tournament. The concentrator itself is built around 2x2 switches using a parallel structure. As seen, packets in the next (K+1)th tournament compete before the Kth tournament is finished. At each stage of the Kth tournament the losers are fed into the next stage of the (K+1)th tournament. In each stage of a tournament where there are an odd number packets competing one of the control portions of the frame has to be delayed until the next stage, since each round is built using 2x2 switches. The delay blocks ,marked D, represent the delay of just the control part of the frame which is used to set the switch direction. By using a parallel structure for the concentrator the number of stages is reduced, thereby reducing the delays for the collision acknowledgement for HLR.

The number of stages needed for a given L can be written as

$$\begin{cases} \log_2 N & K = 1 \\ \log_2 N + 2(K-1) & K \le \log_2 N \\ \log_2 N + 2\left(\log_2 N - 1\right) + \left(K - \log_2 N\right) & L \ge K > \log_2 N . \end{cases} \tag{3.20}$$

The number of stages can be seen to increase by two in each tournament for $K \le \log_2 N$, after which they increase by one stage for the rest of the tournaments. This is due to the binary switches which decrease in number in the first stage of each tournament by a factor of two for $K \le \log_2 N$, after which one stage has to be delayed from the previous tournament to provide two competing packets to the first stage of the next tournament.

In the example in Fig. 3.25 seven packets are destined to the output but only the five with the highest priority are switched, while the rest are blocked and remain buffered at the input. One expects that either W_2^1 or W_3^1 should get the first and second place since they have the highest priority. In this case W_2^1 gets the first place position but either video packets have the same probability of being chosen and the switch simply chooses one at random. Similarly, V_6^1 is seen to be chosen over V_1^1 followed by the file packet F_7^1. There are also two second round interactive sister packets however these are blocked due to the limited switch speedup and remain queued at the input buffer.

To acknowledge the success/failure of the sister packets the concentrator transmits the results of the competition through the bidirectional bus as soon as the last bit of the information bearing part of the frame clears the bus. In this way no time is wasted since the acknowledgment comes right after the data is transmitted. With the acknowledgment the server controller then either selects a new packet from the buffer or keeps the current

packet in the server for the next switching slot.

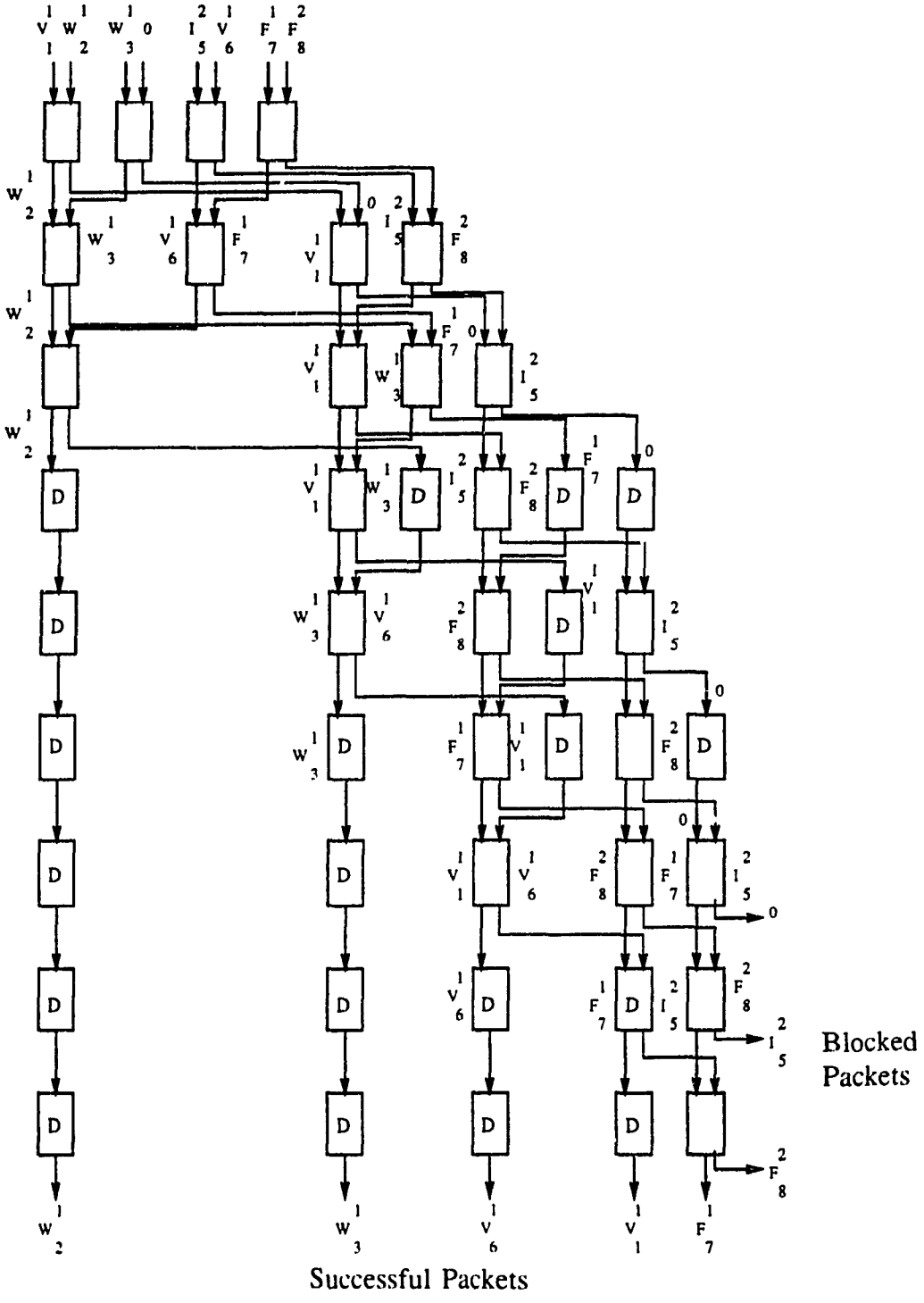


Fig. 3.25 Modified concentrator for priority and HLR, N=8, L=5.

3.4. Uniform Traffic Analysis of the Modified Knockout Switch with Input and Output Buffering

In the following, the performance of the prioritized Knockout switch with input and output buffering speedup and HLR is analyzed. The analysis however is independent of the switch itself and applies equally to any nonblocking switch architecture that can implement priority, speedup and HLR. To analyze the modified Knockout switch discrete-time markov chains are used for both the input and output buffers. The input buffer chain unlike the M/G/1 model used in [44] assumes that only one packet may arrive per time slot which is similar to [45]. However unlike [45], the buffer length is assumed to be finite. Instead of using the tail distribution to approximate a finite queue, the input buffer is solved directly from the balance equations and it is shown that a closed form expression may be found. At the output buffer a finite discrete-time queue is examined with a maximum of L arrivals and with different downlink rates. The model used is similar to [43] except that the finite buffer case is examined and the speedup may be less than N. In the analysis of the on-board switching system independence is assumed between the input and output buffer. That is to say that there is no backpressure mechanism in the system. When the output buffer overflows packets are simply lost. This is a fair assumption since the loss probability of the output buffer is much lower than that of the input buffer system which has to contend with the HOL problem. In addition when the output buffer loss increases, especially under nonuniform traffics, the downlink rate would in practice be increased to handle the load. So that under all conditions a design would ensure that the input buffer is the bottleneck of the system and not the output

buffer. The output buffer performance however is an integral part of the system and is designed so that L packets can be switched to the output per slot with a low loss probability.

At the input of the on-board switching system each of the services is assumed to have its own input buffer, as shown in Fig. 3.26. Here P_W, P_V, P_{FD} and P_{ID} represent the probability that a video, voice, file or interactive data packet arrives during a time slot. Similarly U_W, U_V, U_{FD} and U_{ID} are the probabilities that the respective buffers will be served during one time slot. The markov chain for the video input buffer is shown in Fig. 3.27. The markov chains for the other services have the same form with their corresponding arrival and service probabilities. The service probabilities are assumed to be directly proportional to the traffic ratios of the respective services. In this way there is no priority structure at the input buffer and the loss and buffer delays for each service are independent. If one common buffer had been assumed then some sort of priority structure in the buffer would have had to be implemented to avoid excessive delays to high priority video and voice packets from lower priority file and interactive packets, which get stuck at the head of the line. The price that is paid for examining the fair separate buffer case is that the resulting delays are upper bounded due to the division of the service according to traffic ratios. However, by following this method the relationship between the priorities at the switch between the different services can be clearly seen which would otherwise be obscured.

For the HLR algorithm file and interactive sister packets are assumed to be chosen at random so that no processing time is incurred and the analysis is analytically tractable.

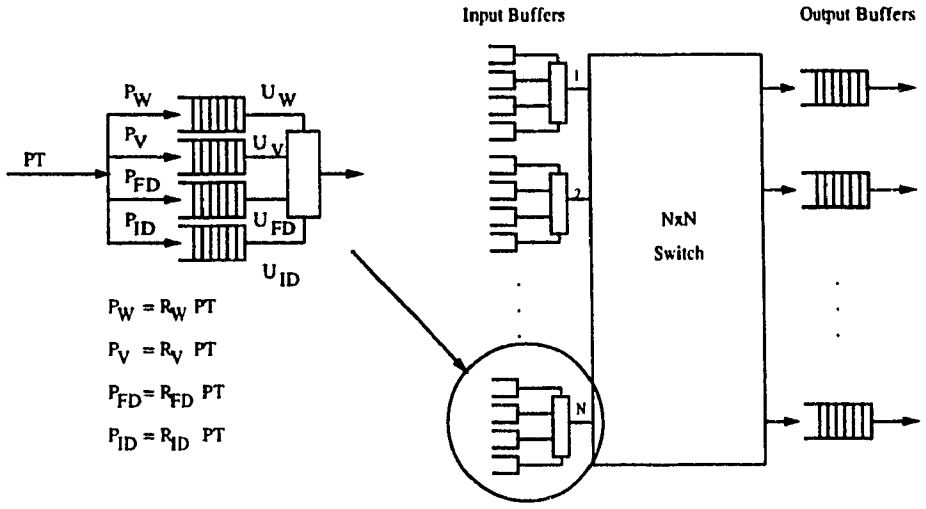
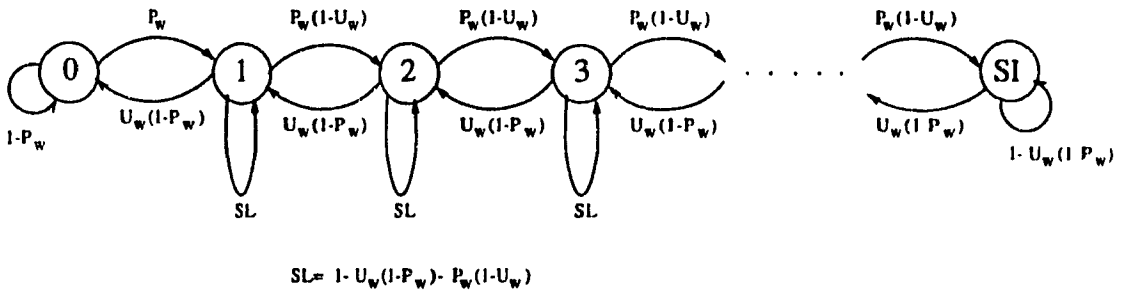


Fig. 3.26. Separate input buffers.



$SL = 1 - U_w(1-P_w) - P_w(1-U_w)$

Fig. 3.27. Discrete-time video input buffer chain.

The random selection assumption is not an unrealistic assumption especially in cases where there is a certain degree of randomness in the system, which is the case under uniform traffic and under most nonuniform traffic cases where some degree of nonuniformity is superimposed on a much larger random traffic base.

To analyze the performance of the switch system both with and without HLR an iterative scheme is used, in which the switch blocking probability for each service is found based on the current input buffer system probability distribution. The resulting blocking probabilities are then used to calculate the service probability of the input buffers. This service probability in turn is then used to find the new probability distribution for the input buffer system. This iterative process is continued until the input buffer system converges for a given input rate. Once convergence is reached the switch blocking probability, input buffer overflow and mean delay are found for each service. Also at convergence the output buffer system is solved by finding the distribution of the number of packets arriving at an output per time slot.

3.4.1. Service Probability With and Without HLR

In order to calculate the switch blocking probabilities for the different types of services the distribution of these services at an output has to be found. Under uniform traffic each input has an equally likely probability of going to any one output so the probability distribution of the number of packets being switched per time slot to any output is binomially distributed. The probability distributions for each service at an output port are given in equations (3.21)-(3.24), where W_i, V_i, FD_i and ID_i denote the probability

distributions for video, voice, file data and interactive data respectively. In addition R_W, R_V, R_{FD} and R_{ID} denote the traffic ratios for the respective services while Φ_{W_0}, Φ_{V_0}, Φ_{FD_0} and Φ_{ID_0} denote the probability that the buffer system is empty for each service respectively. In each of these equations the traffic at a switch input is divided by N to reflect the traffic to one output. The traffic at an input for a service type is given by the probability that there is something in the buffer server times the probability that the buffer is being selected. The product, $\left(1-\Phi_{W_0}\right)R_W$, for example represents the probability that there is something in the video server when the video buffer is being served. An input buffer is assumed to be served in proportion to the traffic ratio since no priority is assumed at the input buffer. However, there is priority in the switch which reflects itself in the resulting service rate for each input buffer.

$$W_i = \binom{N}{i}\left[\frac{\left(1-\Phi_{W_0}\right)R_W}{N}\right]^i\left[1-\frac{\left(1-\Phi_{W_0}\right)R_W}{N}\right]^{N-i} \tag{3.21}$$

$$V_i = \binom{N}{i}\left[\frac{\left(1-\Phi_{V_0}\right)R_V}{N}\right]^i\left[1-\frac{\left(1-\Phi_{V_0}\right)R_V}{N}\right]^{N-i} \tag{3.22}$$

$$FD_i = \binom{N}{i}\left[\frac{\left(1-\Phi_{FD_0}\right)R_{FD}}{N}\right]^i\left[1-\frac{\left(1-\Phi_{FD_0}\right)R_{FD}}{N}\right]^{N-i} \tag{3.23}$$

$$ID_i = \binom{N}{i}\left[\frac{\left(1-\Phi_{ID_0}\right)R_{ID}}{N}\right]^i\left[1-\frac{\left(1-\Phi_{ID_0}\right)R_{ID}}{N}\right]^{N-i} \tag{3.24}$$

The switch blocking probabilities for the different services, equations (3.25)-(3.28), follow the priority scheme outlined in Fig. 3.21 where video has the highest priority and interactive data the lowest. These blocking probabilities are the ratios of the mean number of packets blocked at an output to the mean number of packets destined to an output. For video users as seen in eq. (3.25), the blocking probability is only dependent on video traffic since video is the highest priority and the probability that a tagged packet will be blocked is just the probability that it is one of the discarded packets when more than L video packets arrive at an output. For voice users, eq. (3.26), the blocking probability is dependent on the priority structure between video and voice. The first summation of the numerator takes into account the probability that there are L or more video at an output, in which case all arriving voice to the output will be blocked. The second summation in the numerator is the case when L-1 or fewer video arrive at an output in which case none or a fraction of the arriving voice packets are blocked. Again the blocking probability is found by dividing the mean number of blocked voice packets by the mean number that arrive. The switch blocking probabilities for file and interactive data, equations (3.27) and (3.28) respectively, follow in the same manner except that three types of services are competing for the L available switching slots for file while all four services are competing in the interactive data case. In all the switch blocking probability equations independence is assumed between each of the service types, so that up to N packets of each traffic may go to one output. This assumption introduces more traffic than is actually present and thus

$$Pb_W = \frac{\sum_{a=L+1}^{N} (L-a)\, W_a}{\sum_{a=0}^{N} a\, W_a} \tag{3.25}$$

$$Pb_V = \frac{\sum_{a=L}^{N} W_a \sum_{b=0}^{N} b\, V_b + \sum_{a=0}^{L-1} W_a \sum_{b=L-a+1}^{N} (b-(L-a))\, V_b}{\sum_{b=0}^{N} b\, V_b} \tag{3.26}$$

$$Pb_{FD} = \frac{\left[\begin{array}{l} \sum_{a=L+1}^{N} W_a \sum_{b=0}^{N} b\, V_b \sum_{c=0}^{N} c\, FD_c + \sum_{a=0}^{L} W_a \sum_{b=L-a}^{N} V_b \sum_{c=0}^{N} c\, FD_c \\ \quad + \sum_{a=0}^{L-1} W_a \sum_{b=0}^{L-a-1} V_b \sum_{c=L-a-b+1}^{N} (c-(L-a-b))\, FD_c \end{array}\right]}{\sum_{c=0}^{N} c\, FD_c} \tag{3.27}$$

gives an upper bound to the loss probabilities, however the error is minimal since the binomial probabilities quickly decay.

The service probability for the input buffers for video and voice can now be determined and are given in equations (3.29) and (3.30) respectively. The service

$$Pb_{ID} = \frac{\left[\begin{array}{l} \sum_{a=L+1}^{N} W_a \sum_{b=0}^{N} b\, V_b \sum_{c=0}^{N} c\, FD_c \sum_{d=0}^{N} d\, ID_d \\ + \sum_{a=0}^{L} W_a \sum_{b=L-a}^{N} V_b \sum_{\substack{c=L-a-b \\ c>0}}^{N} FD_c \sum_{d=0}^{N} d\, ID_d \\ + \sum_{a=0}^{L-1} W_a \sum_{b=0}^{L-a-1} V_b \sum_{c=0}^{L-a-b-1} FD_c \sum_{d=L-a-b-c+1}^{N} (d-(L-a-b-c))\, ID_d \end{array}\right]}{\sum_{d=0}^{N} d\, ID_d} \tag{3.28}$$

probabilities for video and voice are the probabilities that a packet is not blocked at the switch upon service.

$$U_W = \left(1-Pb_W\right) R_W \tag{3.29}$$

$$U_V = \left(1-Pb_V\right) R_V \tag{3.30}$$

The service for the case without HLR for file and interactive data follow in the same manner. However for the HLR case before we can calculate the input buffer service probability the sister packet blocking probability has to be found, in order to take care of the possibility that the sister packet may also be blocked at the switch. In the HLR algorithm the same priority scheme is maintained with file data sister packets having priority over interactive data sister packets. To find the sister packet blocking probabilities for file and interactive data we need to find the probability distribution of the number of empty spaces at an output before the sister packets compete. The blocking threshold for file sister packets is now a distribution dependent on how many first round packets arrived at an output. Similarly, the blocking threshold for interactive sister packets is

dependent on how many first round packets as well as how many file sister packets arrived at an output. The distribution for the number of empty spaces left for file sister packets after the first round of competition is given by

$$E_{FD_i} = \binom{N}{L-i}\left(\frac{\Psi}{N}\right)^{L-i}\left(1-\frac{\Psi}{N}\right)^{N-L+i} \qquad i=1,2,3,\ldots,L$$
$$E_{FD_0} = 1-\sum_{i=1}^{L} E_{FD_i} \tag{3.31}$$

where $\Psi = \left(1-\Phi_{W_0}\right)R_W+\left(1-\Phi_{V_0}\right)R_W+\left(1-\Phi_{FD_0}\right)R_{FD}+\left(1-\Phi_{ID_0}\right)R_{ID}$ represents the total traffic from all the services seen at an input port of the switch. Here E_{FD_0} is used to normalize the distribution for computational purposes but E_{FD_0} could also be found directly from the binomial. The file sister packet blocking probability, Psb_{FD}, can now be found and is given by

$$Psb_{FD} = \frac{\sum_{k=0}^{L} E_{FD_k} \sum_{i=k}^{N} (i-k)\binom{N}{i}\left(\frac{Pb_{FD}\,Ps_{FD}}{N}\right)^{i}\left(1-\frac{Pb_{FD}\,Ps_{FD}}{N}\right)^{N-i}}{\sum_{i=0}^{N} i\binom{N}{i}\left(\frac{Pb_{FD}\,Ps_{FD}}{N}\right)^{i}\left(1-\frac{Pb_{FD}\,Ps_{FD}}{N}\right)^{N-i}} \tag{3.32}$$

where $Ps_{FD} = \left(1-\Phi_{FD_0}-\Phi_{FD_1}\right)$ is the probability that there is another packet (a sister packet) in the buffer besides the one being served. The same principle of the ratio of means is used to find the blocking probability. However, now the blocking threshold is a distribution given by E_{FD_k} and in order to find the probability that a sister packet is blocked we have to average over the entire distribution. Here file sister packets are treated as new traffic and compete for the empty spaces left over from the first round of

competition. The probability of having a file sister packet at an input is $Pb_{FD}Ps_{FD}$, which is the probability that the initial packet in the first round of competition was blocked and that there is another packet in the buffer to serve as the sister packet. Having found Psb_{FD}, the mean service probability for the file input buffer with HLR can be written as

$$U_{FD} = \left[\left(1-Pb_{FD}\right)+Pb_{FD}\ Ps_{FD}\left(1-Psb_{FD}\right)\right] R_{FD}. \tag{3.33}$$

The first term inside the brackets, $\left(1-Pb_{FD}\right)$, is the probability that the first round file packet is not blocked while the second term, $Pb_{FD}\,Ps_{FD}\left(1-Psb_{FD}\right)$, represents the improvement due to HLR which is the probability that a file packet is initially blocked, a sister packet is available and finally the sister packet is successful.

The derivation of the HLR service probability for the interactive input buffer follows in much the same manner. However, the probability distribution of the number of empty spaces seen by the interactive sister packets now depends on the success of the file sister packets. The expression for the number of empty spaces left for interactive sister packets at an output is given by

$$E_{ID_i} = \sum_{k=i}^{L} E_{FD_k} \binom{N}{k-i} \left(\frac{Pb_{FD}\ Ps_{FD}}{N}\right)^{k-i} \left(1-\frac{Pb_{FD}\ Ps_{FD}}{N}\right)^{N-k+i}$$
$$i=1,2,3,\ldots,L \tag{3.34}$$

$$E_{ID_0}=1-\sum_{K=1}^{L} E_{ID_k}.$$

E_{ID_i} it is seen is determined by again treating the file sister packets as new traffic. The probability of having i empty spaces after file sister packets compete is then the

probability that there were initially k empty spaces after which k-i file sister packets went to the output port. Since the probability of having k initially empty spaces has the distribution E_{FD_k}, we have to average over this distribution. The interactive sister packet blocking probability follows in eq. (3.35), where $Ps_{ID} = \left(1-\Phi_{ID_0}-\Phi_{ID_1}\right)$ is the probability that there is a sister packet in the interactive buffer. Equation (3.35) averages the sister blocking probability for a given number of empty spaces over the distribution of the number of empty spaces given by E_{ID_i}.

$$Psb_{ID} = \frac{\sum_{k=0}^{L} E_{ID_k} \sum_{i=k}^{N} (i-k) \binom{N}{i} \left(\frac{Pb_{ID}\, Ps_{ID}}{N}\right)^i \left(1-\frac{Pb_{ID}\, Ps_{ID}}{N}\right)^{N-i}}{\sum_{i=0}^{N} i \binom{N}{i} \left(\frac{Pb_{ID}\, Ps_{ID})}{N}\right)^i \left(1-\frac{Pb_{ID}\, Ps_{ID}}{N}\right)^{N-i}}. \tag{3.35}$$

The service probability for the interactive input buffer can then be written as

$$U_{ID}=\left((1-Pb_{ID})+Pb_{ID}\, Ps_{ID}\, (1-Psb_{ID})\right) R_{ID}. \tag{3.36}$$

3.4.2. Input Buffer System Analysis

Since all the buffer systems are assumed to be the same except for the respective arrival and service probabilities the video case will only be examined. The corresponding loss and delay expressions for the other services can then be found by simply substituting the respective arrival and service probabilities. The markov chain of the finite length input buffer system for video was shown in Fig. 3.27. The closed form expression for the probability distribution of the input buffer system chain is derived in Appendix B and can

be expressed as

$$\Phi_{W_i} = \left(\frac{P_W{}^i \left(1-U_W\right)^{i-1}}{\left(\left(1-P_W\right)U_W\right)^i} \right) \Phi_{W_0} \qquad i=1,2,...,SI$$
$$\Phi_{W_0} = \frac{\left(1-XK\right)}{1-X-XK-X^{SI+1}K^{SI}} \tag{3.37}$$

where $X = \dfrac{P_W}{\left(1-P_W\right)U_W}$, $K = \left(1-U_W\right)$ and where SI is the length of the input buffer systems. From Fig. 3.27 it can be seen that the loss probability for the video case can be written as

$$IBL_W = \Phi_{W_{SI}}. \tag{3.38}$$

Due to the single forward transition that the loss probability is equal to the probability of being in state SI. The mean buffer delay for video can then be found by using Little's formula and is given in eq. (3.39), where the numerator is the mean queue size and denominator is the throughput which is the probability that there is something in the system when there is service. It follows that the input loss and mean queuing delay are similarly found for each of the other types of service.

$$IBD_W = \frac{\sum_{i=2}^{SI} (i-1)\,\Phi_{W_i}}{\sum_{j=1}^{SI} U_W \Phi_{W_j}} \tag{3.39}$$

3.4.3. Output Buffer System Analysis

The output buffers are modeled as discrete time markov chains with the possibility of up to L arrivals per time slot from the switch and with various constant downlink service rates, which are multiples of the uplink rate. All the services are queued into a single common output buffer at each switch output with no priority given to any one service. The mean delay at the output buffer and loss is thus the same for all services. To solve the output buffer system the arrival distribution, Pob_k, at an output first has to be found. The reciprocal probability distribution of Pob_k denoted by $E_i^{'}$ is just the number of empty spaces that are left after the interactive sister packets compete and can be found along the same lines as in eq. (3.34). The final distribution of the number of empty spaces left at an output can then be written as

$$E_i^{'} = \sum_{k=i}^{L} E_{ID_k} \binom{N}{K-i} \left(\frac{Pb_{ID}\, Ps_{ID}}{N} \right)^{k-i} \left(1 - \frac{Pb_{FD}\, Ps_{FD}}{N} \right)^{N-k+i}$$

$$i=1,2,3,\ldots,L \qquad (3.40)$$

$$E_0^{'} = 1 - \sum_{k=1}^{L} E_k^{'}$$

Again the independence assumption is invoked and E_{ID_k} is now averaged over all the possible ways of starting with k empty spaces at an output for interactive sister packets and finally ending up with i empty spaces after the interactive sister packets have competed. The distribution of the number of arriving packets at an output follows by reversing the indices of the distribution for the final number of empty spaces left out of L and is given by

$$Pob_k = E'_{L-k} \qquad k=0,1,2,\ldots,L. \tag{3.41}$$

With the arriving distribution in hand the transpose of the discrete time markov transition matrix

$$T = \begin{bmatrix} t_{0,0} & t_{0,1} & t_{0,2} & \cdots & t_{0,S} \\ t_{1,0} & t_{1,1} & t_{1,2} & \cdots & t_{1,S} \\ t_{2,0} & t_{2,1} & t_{2,3} & \cdots & t_{2,S} \\ \vdots & \vdots & \vdots & & \vdots \\ t_{S,0} & t_{S,1} & t_{S,2} & \cdots & t_{S,S} \end{bmatrix}$$

is solved for the different output buffer systems with varying time division multiplexed (TDM) downlink rates. Three cases are examined for DLR=1, 2 and 3 where DLR is defined as the ratio of the downlink rate to the uplink rate. The output buffer system is solved for these ratios in order to examine the effect on the overflow and mean output buffer delays especially under high loads.

The discrete time chain for the output system with DLR=1 is shown in Fig. 3.28

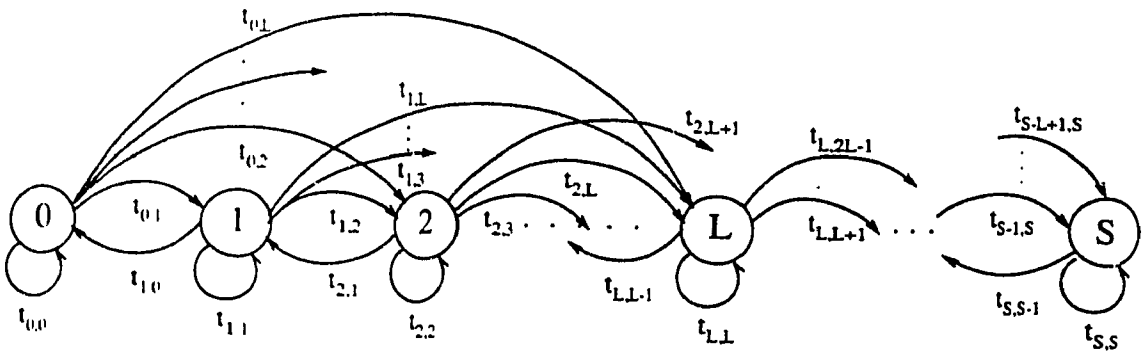


Fig. 3.28. Discrete time output buffer system chain for DLR=1.

where S is the output buffer system length. The values of the transition matrix can be described by eq. (3.42).

$$
\begin{aligned}
t_{i,j} &= Pob_j && i=0, \quad 0 \le j \le L \\
t_{i,j} &= Pob_0 && 0 < i \le S, \quad j=i-1 \\
t_{i,j} &= Pob_{j-i+1} && 0 < i \le S-L+1, \quad i < j < i+(L-1) \\
t_{i,j} &= Pob_1 && S > j = i > 0 \\
t_{i,j} &= 1-Pob_0 && i=j=S \\
t_{i,j} &= \sum_{k=j-i+1}^{L} Pob_k && S > i > S-L+1, \quad j=S \\
t_{i,j} &= Pob_{j-i+1} && S > i > S-L+1, \quad i < j \le S-1
\end{aligned}
\tag{3.42}
$$

Solving the chain, the distribution of the output buffer is found and is denoted by Θ_i. The buffer overflow probability is calculated by averaging the probability of being discarded on arrival given the system is a certain state, over the probability distribution Θ_i of the system as shown in eq. (3.43).

$$
OBL_{DLR=1} = \sum_{i=S-L+2}^{S} \Theta_i \left[\frac{\left(\sum_{j=0}^{L} (i+j-S-1) Pob_j \right)}{\sum_{j=0}^{L} j Pob_j} \right] \tag{3.43}
$$

Here the ratio of the means is used to determine the loss probability associated with a given state. The numerator of the term inside the brackets is the mean number lost and the denominator is the mean number to arrive. The output buffer delay, shown in eq. (3.44), is found using Little's formula with the mean queue size divided by the throughput. In this case the service is constant so the throughput is just the probability

that there is something in the server.

$$OBD = \frac{\sum_{i=2}^{S}(i-1)\Theta_i}{\sum_{j=1}^{S}\Theta_k} \tag{3.44}$$

For the case where the downlink rate in twice that of the uplink rate (DLR=2) the buffer system markov chain can be modeled as shown in Fig. 3.29. The transition probabilities can be described by the following

$$\begin{aligned}
t_{i,j} &= Pob_j && i=0,\ 0\le j\le L \\
t_{i,j} &= Pob_{j-i+2} && \{0<i\le S-L+2,\ i<j<i+L-2\} \\
& && \vee\ \{S>i>S-L+2,\ i<j\le S-1\} \\
t_{i,j} &= \sum_{k=j-i+2}^{L} Pob_k && S>i>S-L+2,\ j=S \\
t_{i,j} &= Pob_2 && S>j=i>0 \\
t_{i,j} &= 1-Pob_0-Pob_1 && i=j=S \\
t_{i,j} &= Pob_1 && S\ge i>1,\ j=i-1 \\
t_{i,j} &= Pob_1+Pob_0 && i=1,\ j=0 \\
t_{i,j} &= Pob_0 && S\ge i\ge 2,\ j=i-2\,.
\end{aligned} \tag{3.45}$$

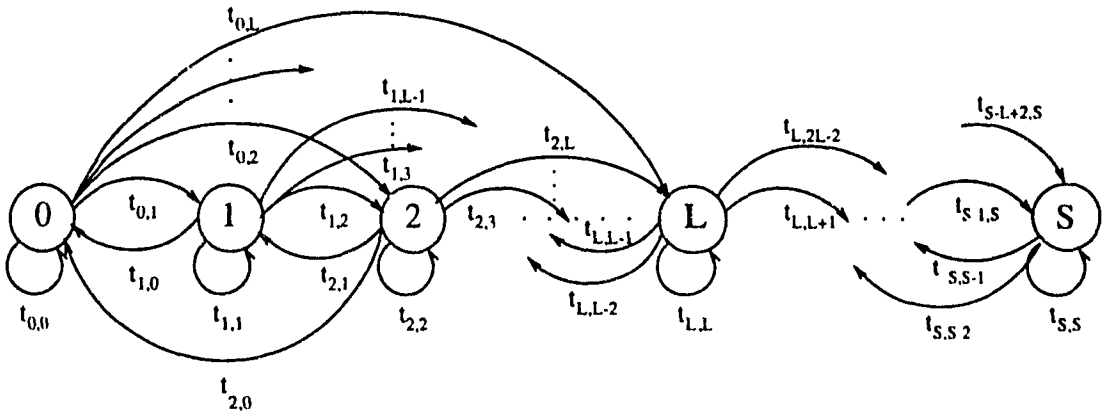


Fig. 3.29. Output buffer chain for *DLR*=2.

Similarly the discrete markov chain for the case when *DLR*=3 is shown in Fig. 3.30 and the transition probabilities can be described by

$$
\begin{aligned}
t_{i,j} &= Pob_j && i=0,\ 0\le j\le L \\
t_{i,j} &= Pob_{j-i+3} && \{0<i\le S-L+3,\ i<j<i+L-3\} \\
& && \vee\ \{S>i>S-L+3,\ i<j\le S-1\} \\
t_{i,j} &= \sum_{k=j-i+2}^{L} Pob_k && S>i>S-L+3,\ j=S \\
t_{i,j} &= Pob_3 && S>j=i>0 \\
t_{i,j} &= 1-Pob_0-Pob_1-Pob_2 && i=j=S \\
t_{i,j} &= Pob_2 && S\ge i\ge 2,\ j=i-1 \\
t_{i,j} &= Pob_0+Pob_1+Pob_2 && i=1,\ j=0 \\
t_{i,j} &= Pob_0+Pob_1 && i=2,\ j=0 \\
t_{i,j} &= Pob_1 && S\ge i\ge 3,\ j=i-2 \\
t_{i,j} &= Pob_0 && S\ge i\ge 3,\ j=i-3.
\end{aligned}
\qquad (3.46)
$$

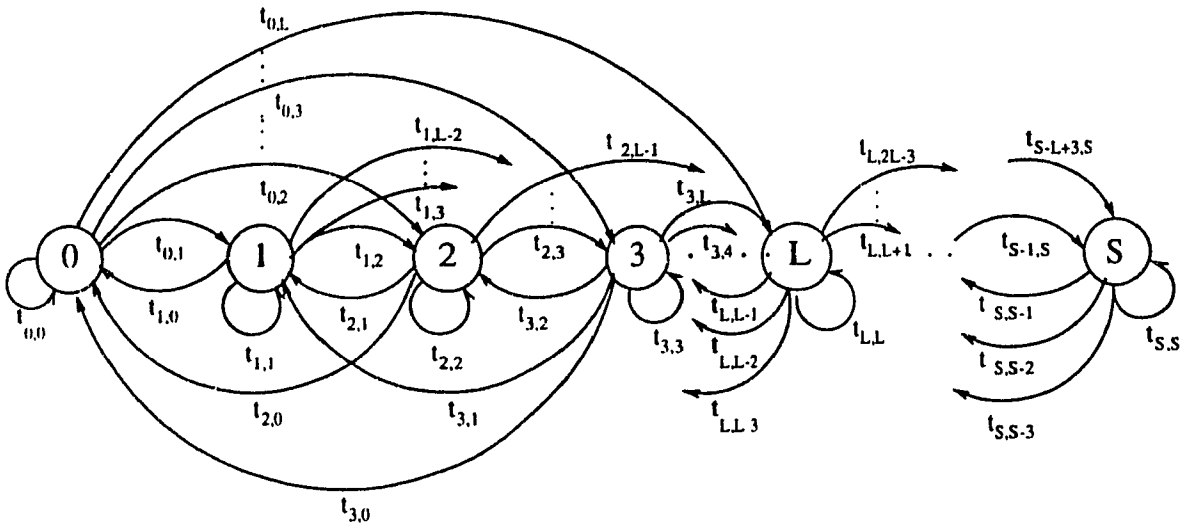


Fig. 3.30. Output buffer chain for *DLR*=3.

The loss probability and the mean buffer delays for *DLR*=2 and *DLR*=3 can be found similarly to those for the *DLR*=1 case and the general formulas dependent on *DLR* are given in equations (3.47) and (3.48).

$$OBL_{DLR} = \sum_{i=S-L+(DLR+1)}^{S} \Theta_i \left[\frac{\left(\sum_{j=0}^{L} (i+j-S-DLR) Pob_j \right)}{\sum_{j=0}^{L} j\, Pob_j} \right] \qquad (3.47)$$

$$OBD_{DLR} = \frac{\sum_{i=DLR+1}^{S} (i-DLR)\Theta_i}{\sum_{j=1}^{DLR} j\Theta_j + DLR \sum_{j=DLR+1}^{S} \Theta_j} \tag{3.48}$$

The ratio of means is again used for find the buffer loss in eq. (3.47). As an example, for DLR=2 if L=4 and S=6 and it is assumed that j=4 packets arrive then i=S-L+(DLR+1)=5. Being in state five with four arrivals, two packets are lost, one is queued and one is lost. Thus the system has to be in state five for at least one packet to be lost (i+j-S-DLR=1). Similarly, if one is in state six then two packets are lost with four arrivals and one packet is lost with three arrivals.

The output buffer delay given in eq. (3.48) takes into account the multiple DLR servers so that to find the mean queue size the distribution is summed from DLR+1 to S. In the denominator the throughput is calculated by increasing the service by one for each state up to DLR after which the service remains constant at DLR.

3.5. Performance Results Under Uniform Traffic Conditions

3.5.1. Effect of Speedup and HLR on Input Buffer Delays

The input buffer delays for the prioritized Knockout switch with and without HLR are shown in Figures 3.31-3.33. In each figure the delay versus the load for different switch speedups is plotted with the buffer length fixed at 30. In addition the traffic ratios are assumed to be equal, $R_W=R_V=R_{FD}=R_{ID}$, and the load PT is assumed to be divided equally among the four services, i.e., $P_W=P_V=P_{FD}=P_{ID}$, $PT=P_W+P_V+P_{FD}+P_{ID}$. In this way no priority is assumed in the buffers and each receives the same amount of traffic. The priority structure of the Knockout switch can be clearly seen in Fig. 3.31 where L=1. In this plot the highest priority video users have the lowest delay while the lowest priority interactive users have the highest delay. With a limited switching capacity the probability of collision limits the maximum file data load for a delay of 20 without HLR to 0.56 and the load for the lowest priority interactive data users to 0.48. With HLR however the throughput can be substantially increased. The maximum load as seen for file data users may be increased to approximately 0.62 while that of interactive data may be increased to 0.56 for a delay of 20. As the speedup is increased to two, as shown in Fig. 3.32, the throughput is increased due to the diminishing switch blocking probability. As seen, for L=2 the performances of the various services are beginning to converge. This is especially true of the HLR cases where the algorithm takes advantage of the lower blocking probabilities. The probability that a file or interactive data sister packet is blocked on the second round of competition decreases as the speedup is increased, which results in a higher throughput for the lower priority users. The maximum load for file data users as

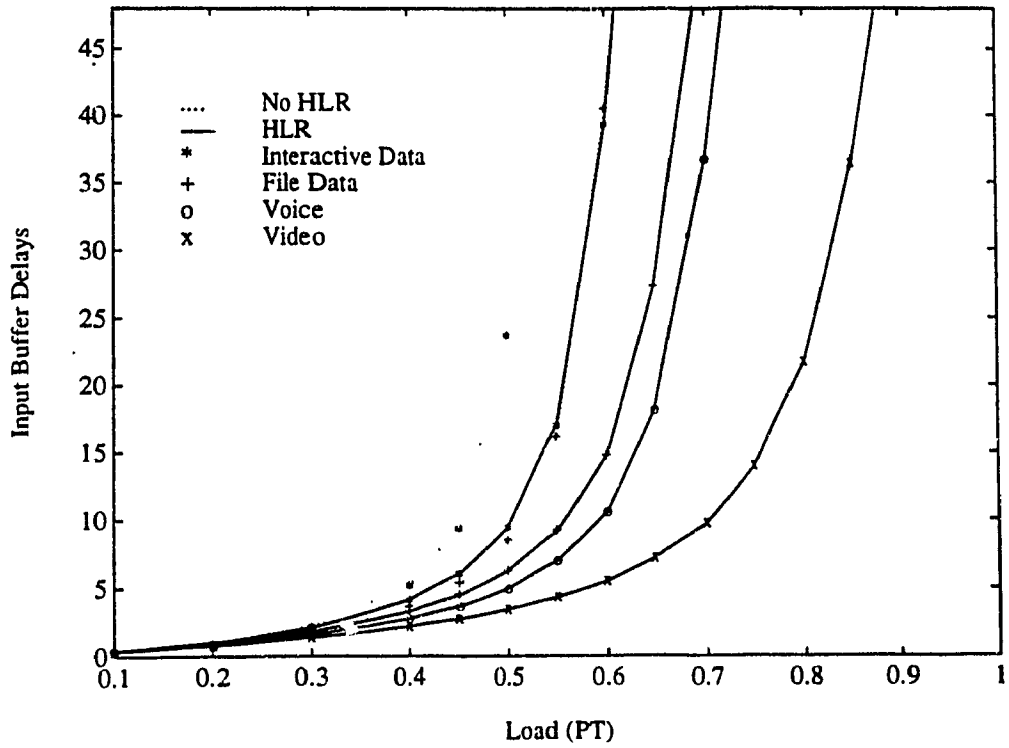


Fig. 3.31. Input buffer delays (L=1).

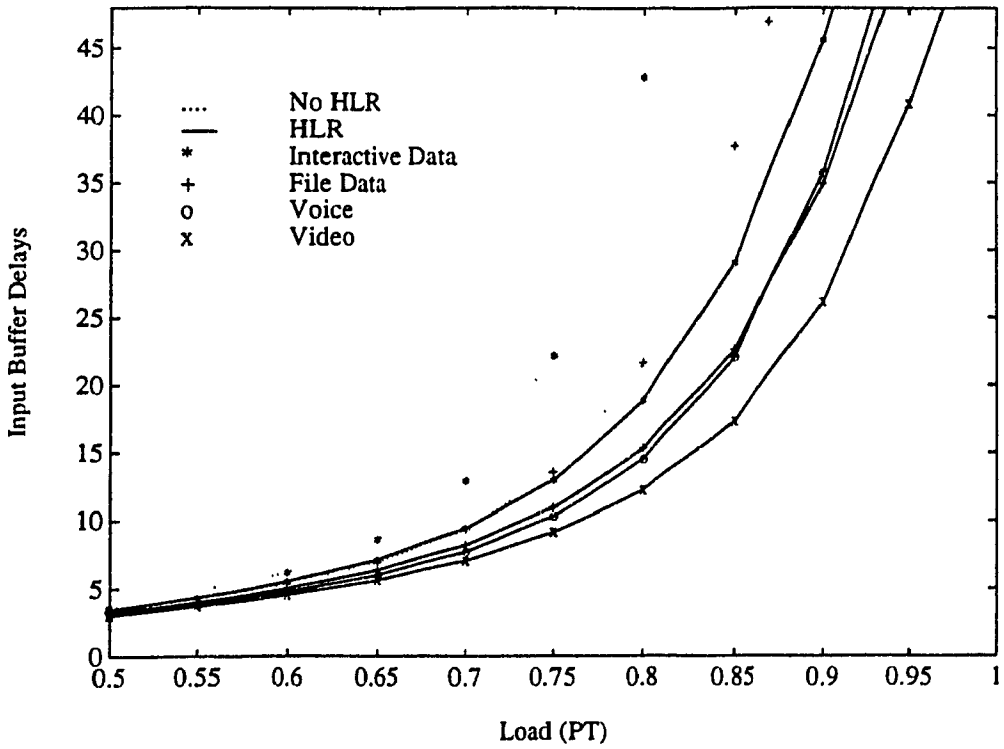


Fig. 3.32. Input buffer delays (L=2).

seen increases from 0.79 without HLR to 0.84 with HLR while that of interactive data increases from 0.74 to 0.81 at a delay of 20 respectively. The HLR algorithm it is seen improves the performance for interactive data users more than it does for file data users. This is due to the lower initial blocking probability of file data which already approaches the maximum buffer load without HLR. With a higher initial blocking probability interactive data users have more to gain from HLR and do so due to the availability of empty spaces at the output. As the sister packet blocking probability increases though the improvement decreases. Increasing the speedup to L=3 as is shown in Fig. 3.33 has the effect of converging all the services even further. As shown, there is only about a 0.02 difference in the maximum load between all the services even at loads above 0.95. This is due to the low switch blocking probability which reflects itself in low delays at the input buffer. Therefore, under uniform traffic conditions the speedup of the switch can be kept at a minimum in a buffered system in which a priority structure is implemented. It is seen that the HLR algorithm substantially improves the throughput at low values of switch speedup when there is a high blocking probability. At higher switch speedups however the need for the HLR algorithm decreases. Thus, the advantage of the HLR algorithm is that it reduces the switch speedup needed to provide a minimum service for the low priority users.

To examine the effect of changing the traffic ratio on the interactive buffer delay, the delay is plotted in Fig. 3.34 for different ratios with and without HLR assuming a buffer length of 30. It is seen that the HLR algorithm provides a substantial improvement

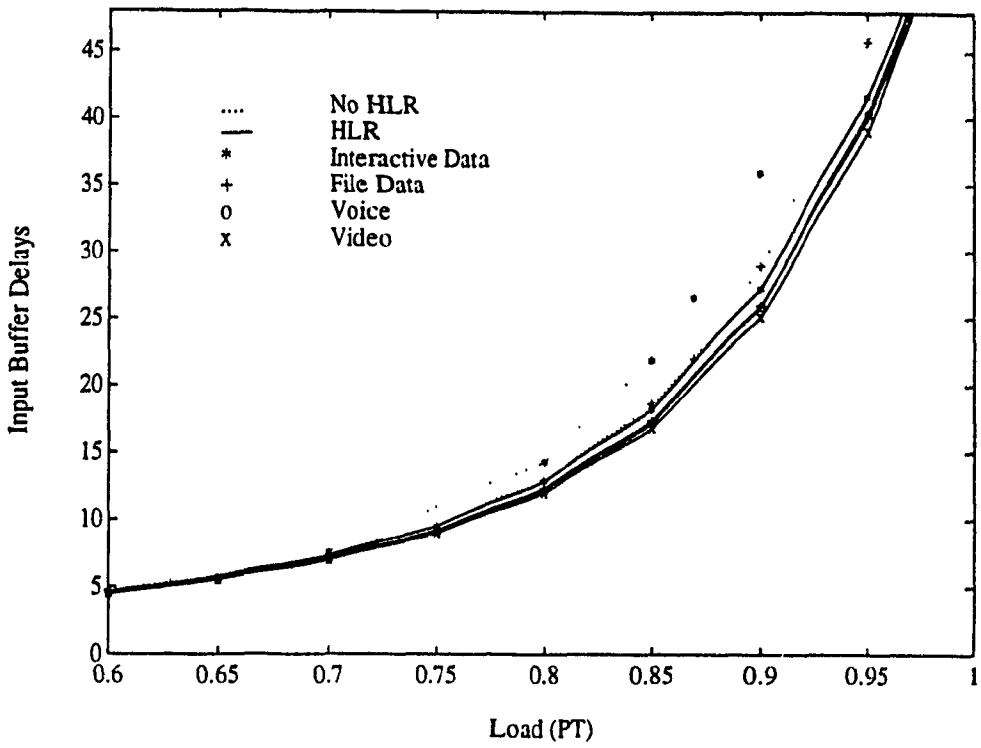


Fig. 3.33. Input buffer delays (L=3).

in the maximum load for all of the cases. With a low ratio the probability of being blocked due to a higher priority is large and limits the load to 0.46 which can be increased to 0.51 with HLR. The improvement of the HLR algorithm is also limited by the high sister packet blocking due to the larger number of higher priority file sister packets. For a ratio of 0.95 the maximum load is around 0.6 without HLR but may be as high as 0.7 with HLR. The large increase is due to most of the traffic being interactive data users. Thus, there is a greater chance that an interactive data user will be successful in the first round as well as in the second since there are few higher priority users.

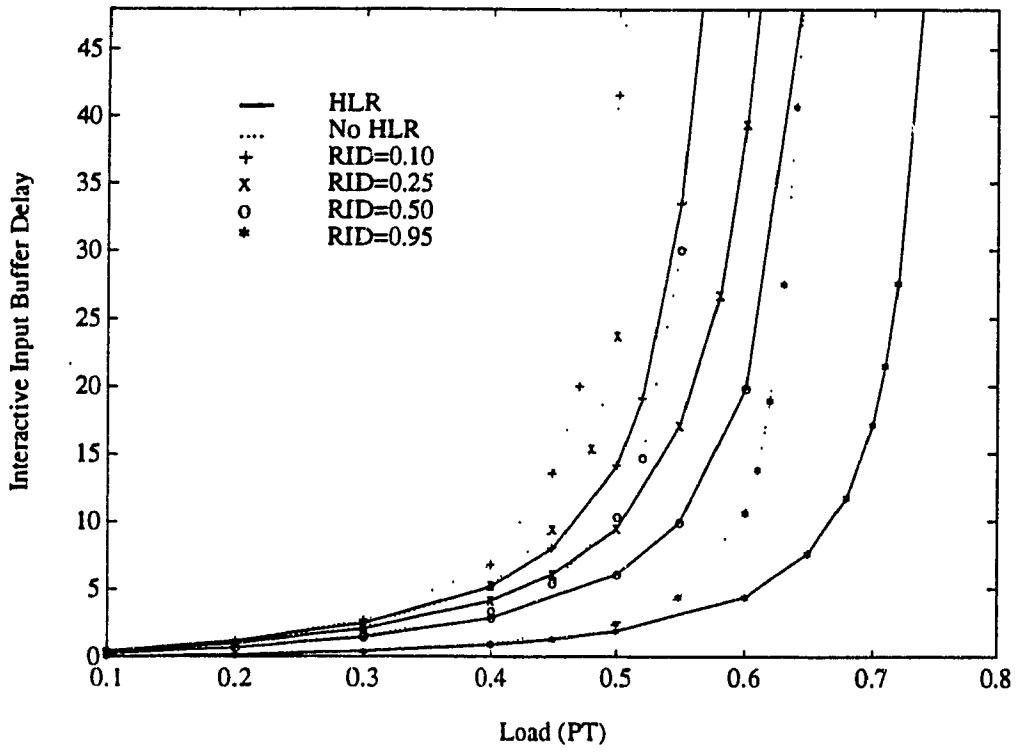


Fig. 3.34. Input buffer delays for different traffic ratios.

3.5.2. Effect of Buffering, Speedup and HLR on the Loss Probability

To examine the effect of input buffering in the prioritized Knockout switch the loss probability without buffering as L is increased is compared with the buffer overflow case for each of the services respectively. Fig. 3.35 compares the loss probability for video service in the discard case with that of the buffered case for different values of L and different input buffer lengths. It is seen that as the input buffer length is increased the loss is substantially reduced at low to medium traffic levels but due to the saturation of the finite buffer experiences overflow losses above $1x10^{-6}$ at loads above 0.8. As shown, an increase in buffer length from 20 to 30 to 40 for L=1 at a load of 0.5 represents a

decrease in loss from $1.23x10^{-7}$ to $5.5x10^{-11}$ to $2.47x10^{-14}$ while at a higher load of 0.7 the loss probability may be reduced from $3.25x10^{-4}$ to $1x10^{-5}$ to $3.39x10^{-7}$ respectively. In addition by using a speedup of two and a moderate buffer length of 30, it is seen that the loss at a load of 0.7 can be reduced below that of the discard case with L=4 while at a load of 0.4 the buffered case is equivalent to the L=8 discard case. Increasing L above two has little effect on the loss probability for the buffered case since video is the highest priority traffic and has a very low blocking probability.

The results of voice traffic, shown in Fig. 3.36, show an even greater improvement in the loss probability for the buffered case versus the discard case due to the decrease in priority. At a speedup of two for a buffer length of 30 and a load of 0.53 the equivalent discard case requires a speedup of eight to meet the same loss. Similarly, the discard case requires a speedup of six at a load of 0.7 to match the L=2 buffered case. It is also seen that for voice users there is not only a significant difference between the buffer lengths but also between the L=1 and L=2 cases and a smaller difference between L=2 and L=3, which becomes larger as the load increases. This is due to the fact that voice suffers at the expense of video and the blocking at the switch increases as the load increases. Due to the higher blocking there is a much greater difference between the L=1 and L=2 cases as compared to video, however the blocking limits the maximum load for the L=1 case to less than 0.6 for a loss of $1x10^{-6}$ even with a buffer length of 40.

For file data users the difference between the discard and the buffered cases with HLR increases as is shown in Fig. 3.37. It is still seen that a moderate speedup of two or three is only needed in order to limit the loss probability to $1x10^{-6}$. Since file data is

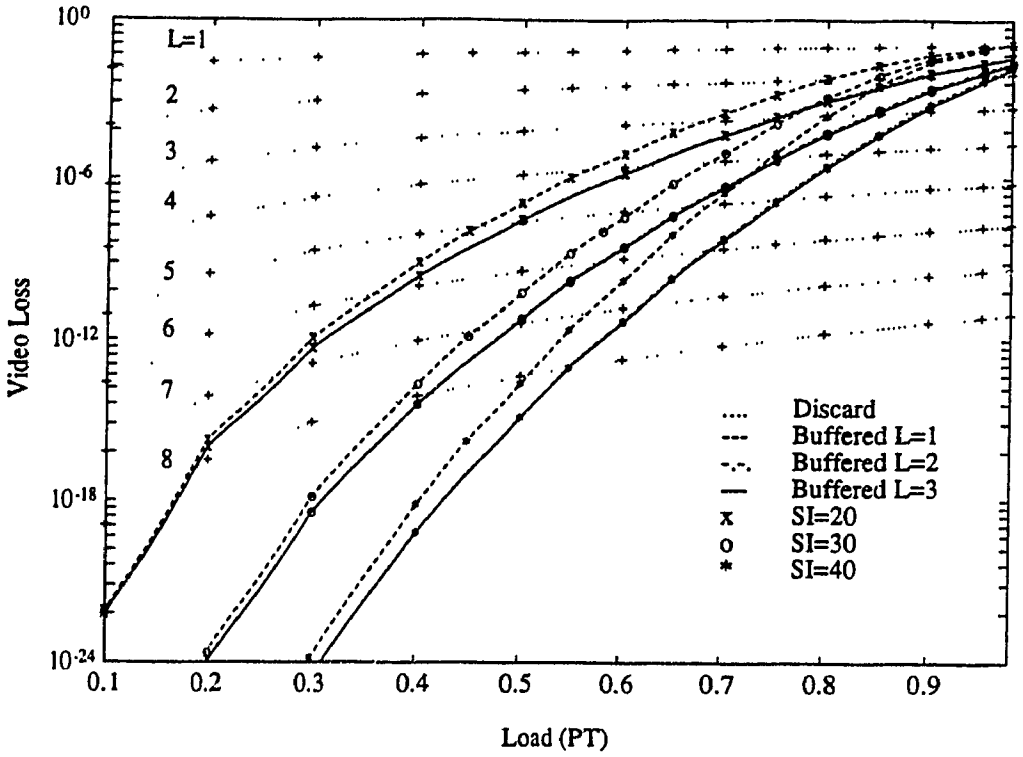


Fig. 3.35. Video loss with and without buffering.

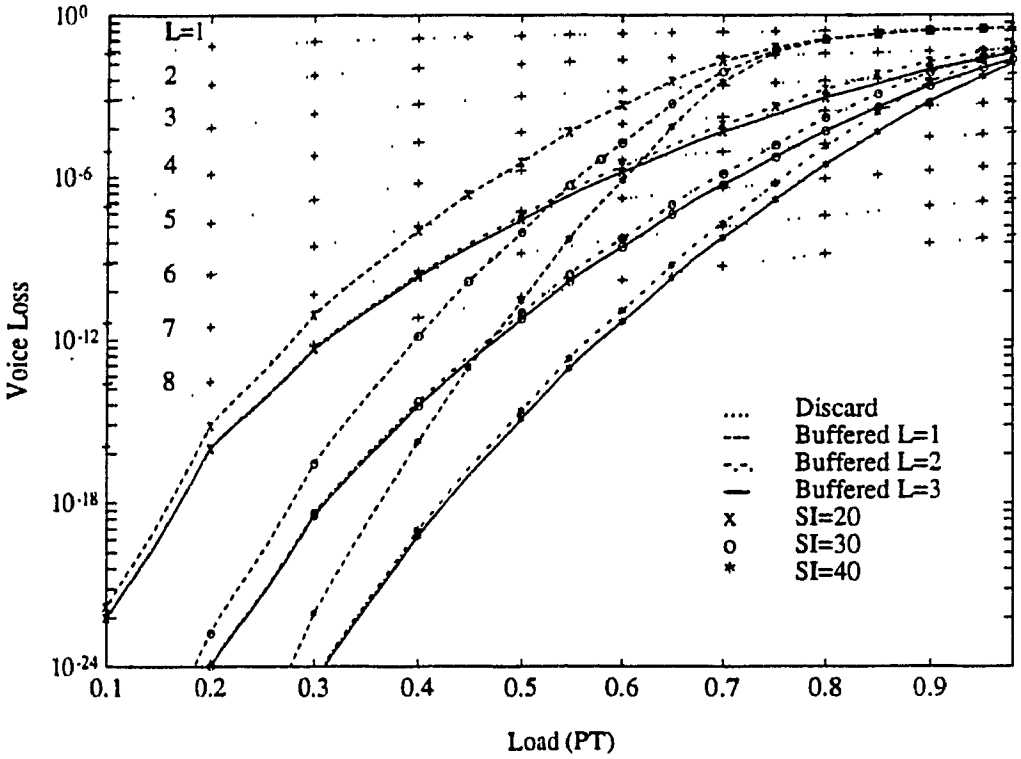


Fig. 3.36. Voice loss with and without buffering.

the third priority service the difference between the buffered case with L=2 and L=3 is becoming more apparent, especially at loads greater than 0.6. In addition, the dominance of the higher speedup is more visible as the buffer length increases. To ensure the lowest loss however, both the buffer length and speedup may have to be increased for high loads. As shown, a loss of less than $1x10^{-6}$ can still be maintained for the L=2 case up to a load of 0.7 with a buffer length of 30.

With HLR the lowest priority service namely interactive data, as shown in Fig. 3.38, also continues to maintain acceptable loss probabilities with L=2 and L=3 for modest input buffering while the performance of the discard case continues to decrease. Since interactive data suffers the highest blocking probability the dependence on L is much more pronounced. The loss probabilities for the L=2 and L=3 cases for a load of 0.8 with a buffer of size 40 for example are $9.5x10^{-5}$ and $5.01x10^{-6}$ respectively. For the discard case this is equivalent to increasing L from five to six. The losses at a load of 0.5 with L=2 and with buffer lengths of 20, 30 and 40 are $1.2x10^{-7}$, $5.39x10^{-11}$ and $2.39x10^{-14}$ respectively and the losses at a load of 0.7 for the same buffer lengths are $2.74x10^{-4}$, $8.02x10^{-6}$ and $2.35x10^{-7}$ respectively. Comparing these results with those for buffered video it is seen that these losses are comparable at these loads with those of buffered video with L=1. In addition, it is seen that a buffer size of 30 is sufficient to reduce the loss for interactive data to $1x10^{-6}$ for a speedup of three at a load of 0.7. The difference therefore between the speedup needed for video and interactive data with HLR is just one, while in the discard case the speedup has to be increased by four, from L=4 to L=8, in order to maintain the same performance.

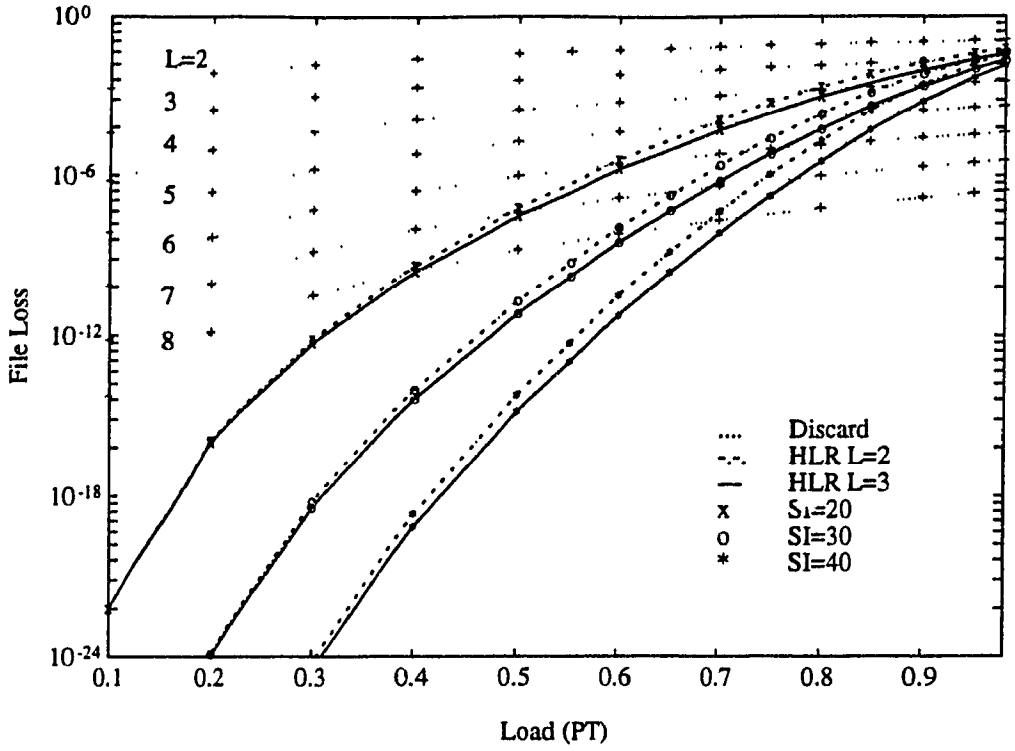


Fig. 3.37. File data loss with and without buffering.

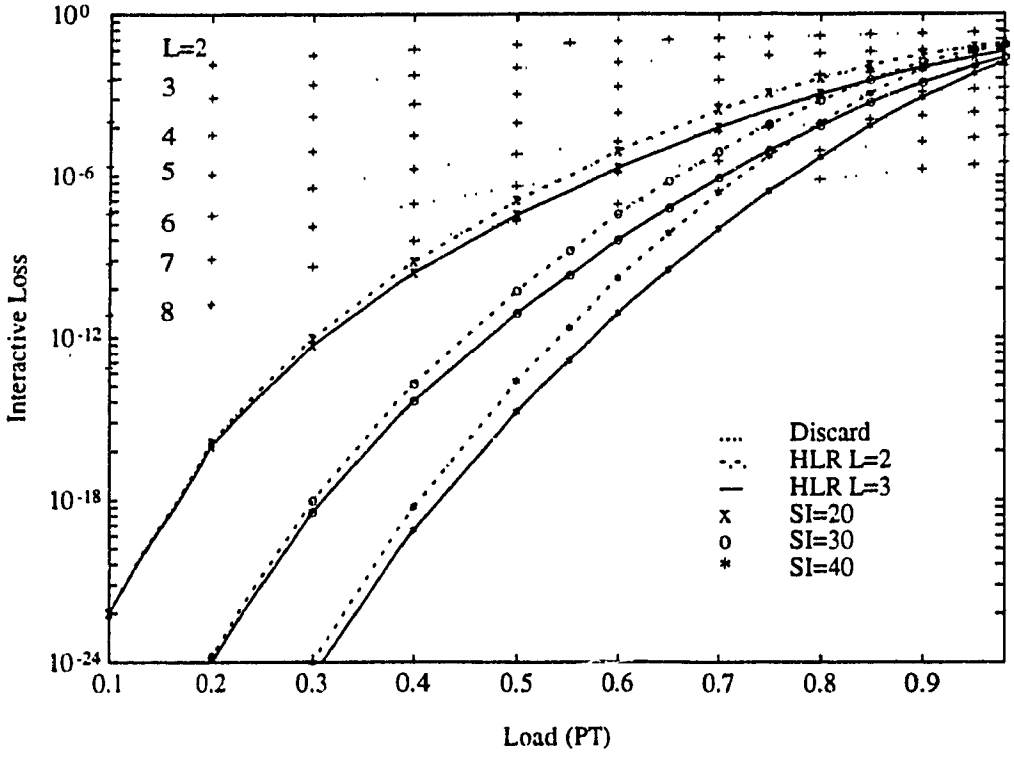


Fig. 3.38. Interactive data loss with and without buffering.

3.5.3. The Effect of the Input Buffer Length on the Performance

In order to find the appropriate input buffer length for the prioritized Knockout switch under uniform traffic conditions the loss probability for each of the services is plotted versus the buffer size, as shown in Figures 3.39-3.46, for different traffic loads and different switch speedups . For video users it is seen in Fig. 3.39 that a speedup of two and a buffer length of 30 is sufficient to maintain a loss below $4x10^{-7}$ for loads less than 0.7. For L=1 an increase in buffer length to 37 is needed to ensure a loss of $1x10^{-6}$ at a load of 0.70. For voice traffic with a speedup of two, as shown in Fig. 3.40, a buffer length of 30 still provides enough buffering for a load of 0.7 to maintain a loss of $1x10^{-6}$. However, at L=1 with a buffer length of 40 a maximum load of only 0.6 can be supported at a loss of $1x10^{-6}$ which is a 0.1 drop in the load compared to video. Figures 3.41-3.44 show the results with and without HLR for file data users as L is increased. As seen in Fig. 3.41 with L=1 and HLR, an overflow of less than $2x10^{-7}$ can be maintained for loads up to 0.50 with a buffer length of 30. Without HLR a buffer of length 39 is required to maintain the same loss. It is also seen that the improvement due to HLR increases as the buffer length increases. This is due to the increased probability of finding a sister packet in the buffer as the buffer length increases. The HLR algorithm also substantially decreases the loss at higher loads. For a load of 0.6 with a buffer length of 30 the loss is reduced by two orders of magnitude to $1.5x10^{-4}$ with HLR from $1x10^{-2}$ without HLR. Increasing the speedup to two, as shown in Fig. 3.42, makes it possible reduce the buffer size to 23 for a load of 0.6 and still provide an overflow probability of $1x10^{-6}$ with HLR. For a load of 0.70 for the same overflow probability the buffer length has to be increased

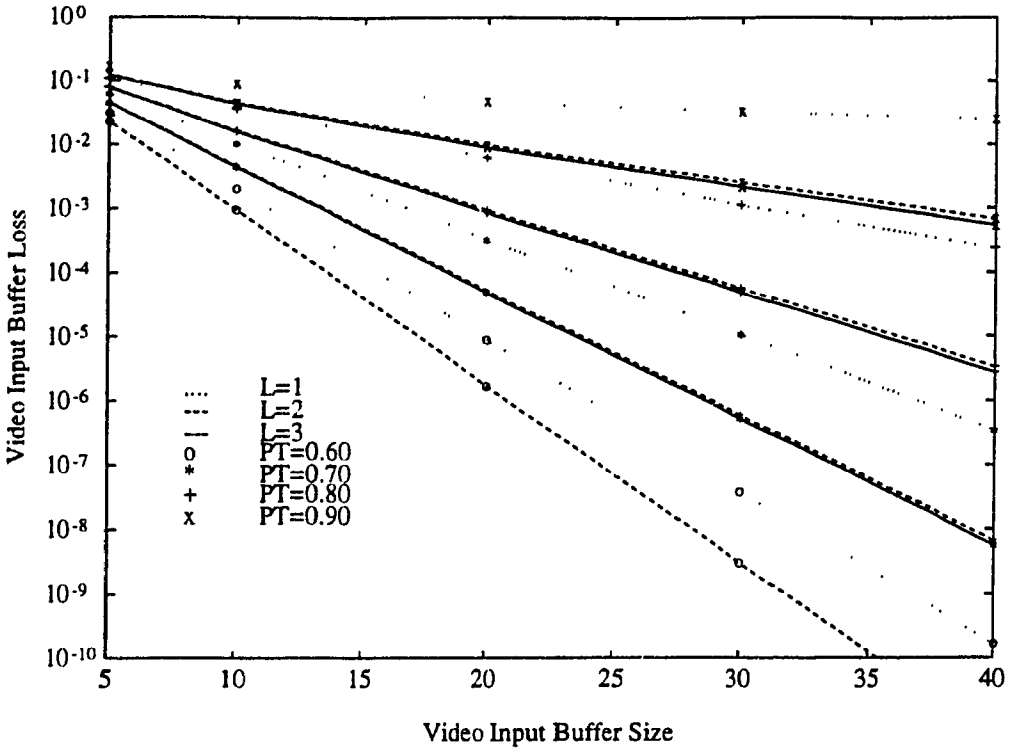


Fig. 3.39. Video input buffer loss for increasing buffer size.

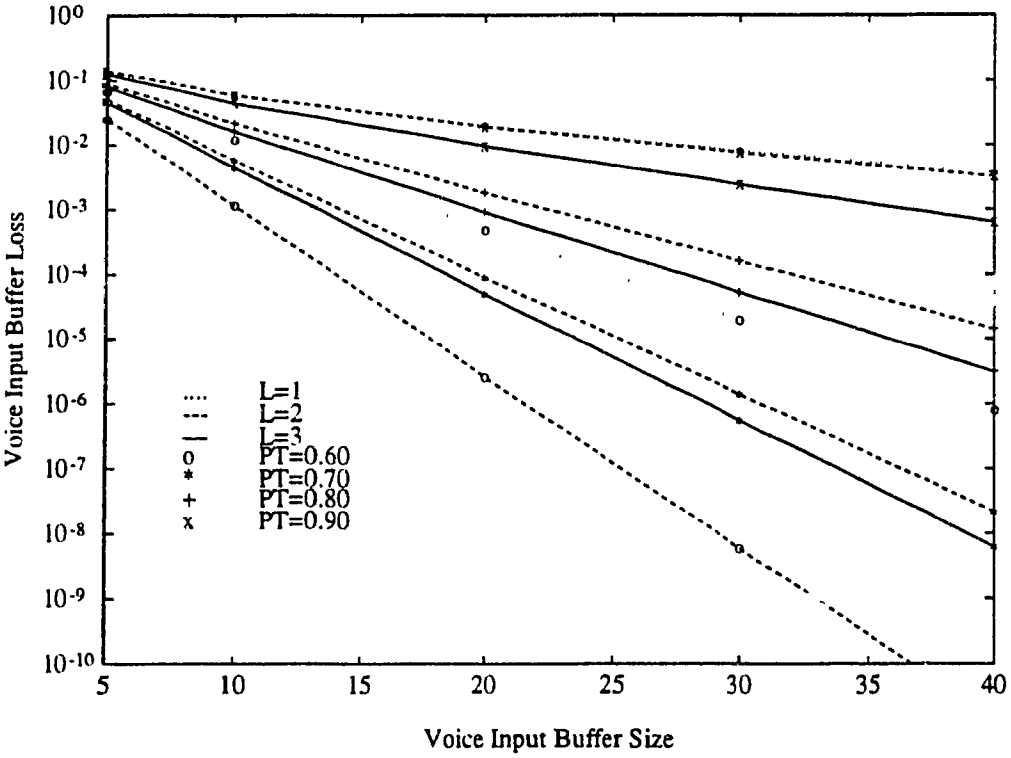


Fig. 3.40. Voice input buffer loss for increasing buffer size.

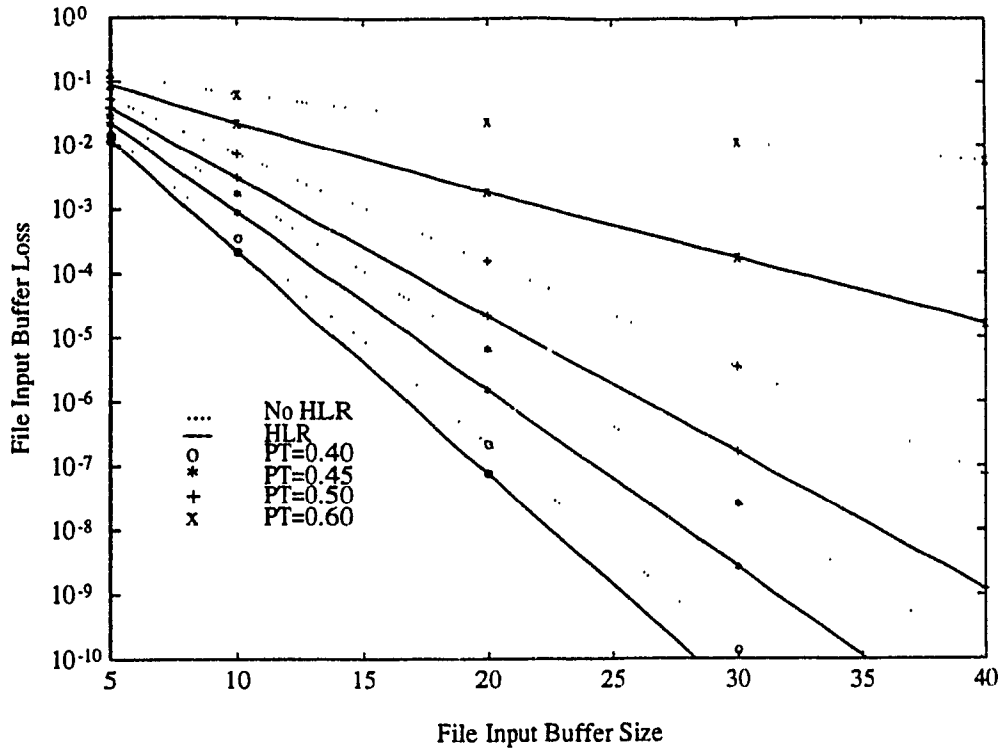


Fig. 3.41. File input buffer loss with increasing buffer size (L=1).

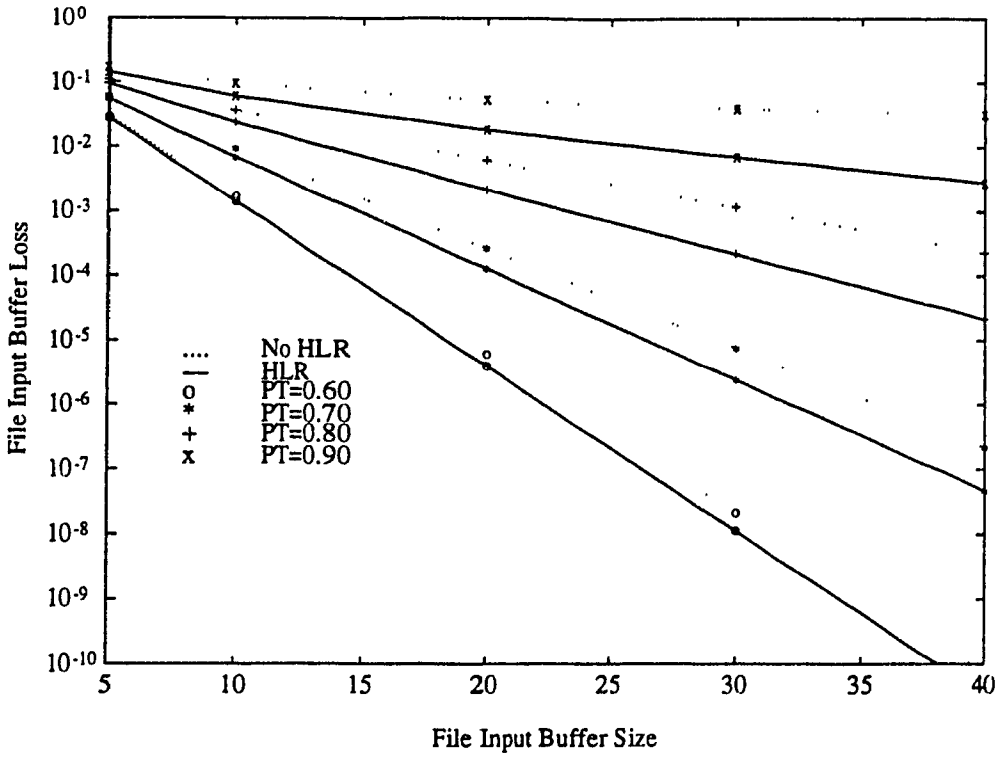


Fig. 3.42. File input buffer loss with increasing buffer size (L=2).

to 33 with HLR and to 37 without HLR. Increasing the switch speedup to three, as shown in Fig. 3.43, decreases the improvement due to the HLR algorithm further. As shown, a buffer of length of 30 is sufficient to reduce the loss to $6x10^{-7}$ for a load of 0.7 for both cases. Even at higher loads of 0.80 and 0.90 the increased speedup has reduced the need for the HLR algorithm. The results for interactive data are shown in Figures 3.44-3.46. As can be seen for L=1 a buffer size of 26 is sufficient for the HLR case to reduce the loss to $1x10^{-6}$ for a load of 0.45 but without HLR a buffer of length 36 is required. Also, it can be seen that HLR provides an even greater improvement in the overflow probability for interactive data than for file data. At a buffer length of 30 and at a load of 0.5 for example the HLR algorithm reduces the overflow probability to $9x10^{-6}$ from $2x10^{-3}$. Increasing the speedup to two, as shown in Fig. 3.45, it is seen that HLR continues to provide a significant improvement. At a load of 0.60 and with a buffer of size 30 the HLR case has a loss of $4x10^{-8}$ while without HLR the loss is $1.5x10^{-7}$. Alternatively, the input buffer has to be increased in length from 28 with HLR to 32 without HLR for a loss of $1x10^{-7}$. At a higher load of 0.7 to maintain a loss of $1x10^{-5}$ the buffer has to be increased in length from 29 with HLR to 37 without HLR. Thus, for interactive data which is the lowest priority traffic and suffers the highest losses the HLR algorithm is necessary to minimize the buffer size for loads greater than 0.6 even with a speedup of two. With L=3, as shown in Fig. 3.46, it is seen that the increased speedup now again become more dominant and the improvement due to HLR decreases. It can be seen that a buffer length of 30 again provides a minimum loss of $1x10^{-6}$ at a load of 0.7 without HLR, due to the small difference between the case with HLR for a speedup of three at

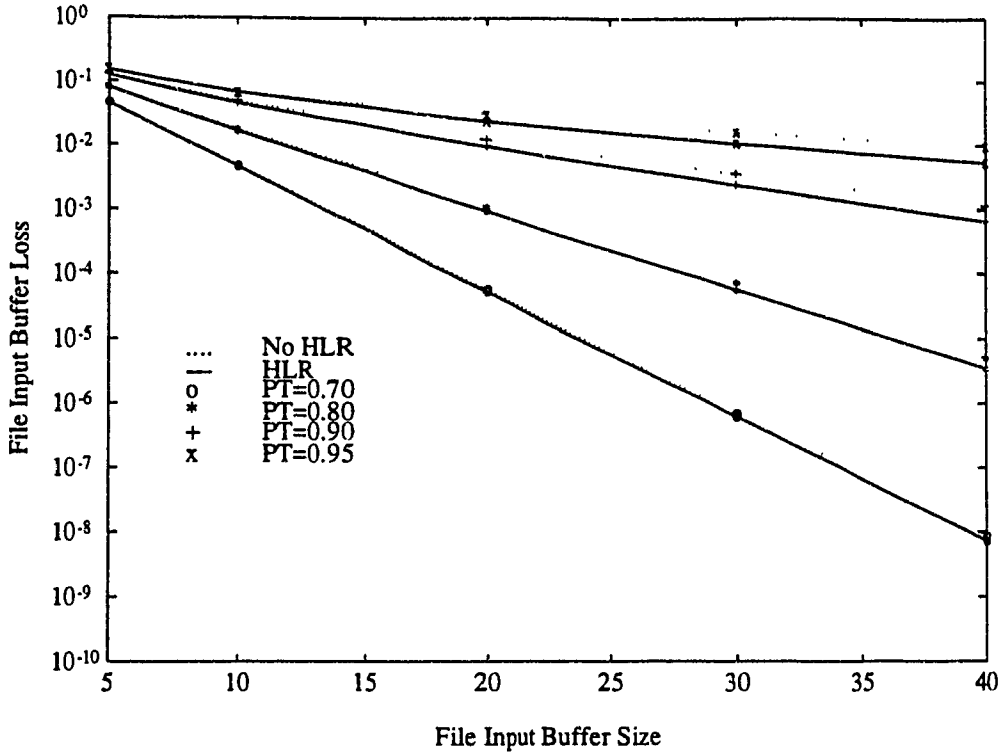


Fig. 3.43. File input buffer loss with increasing buffer size (L=3).

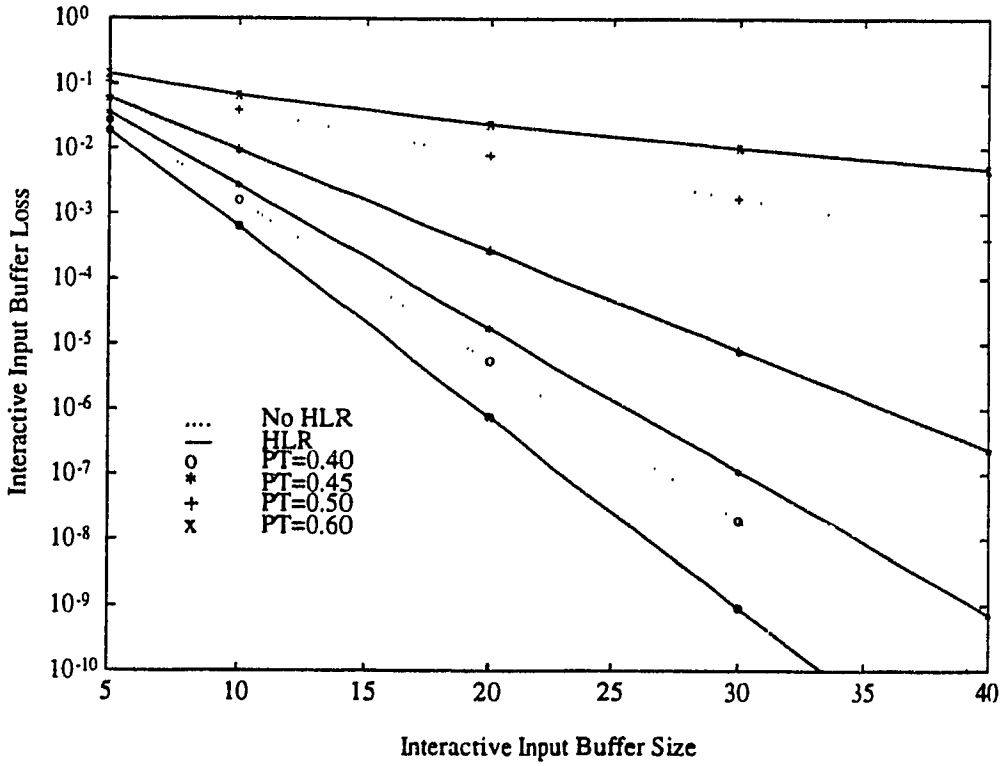


Fig. 3.44. Interactive input buffer loss with increasing buffer (L=1).

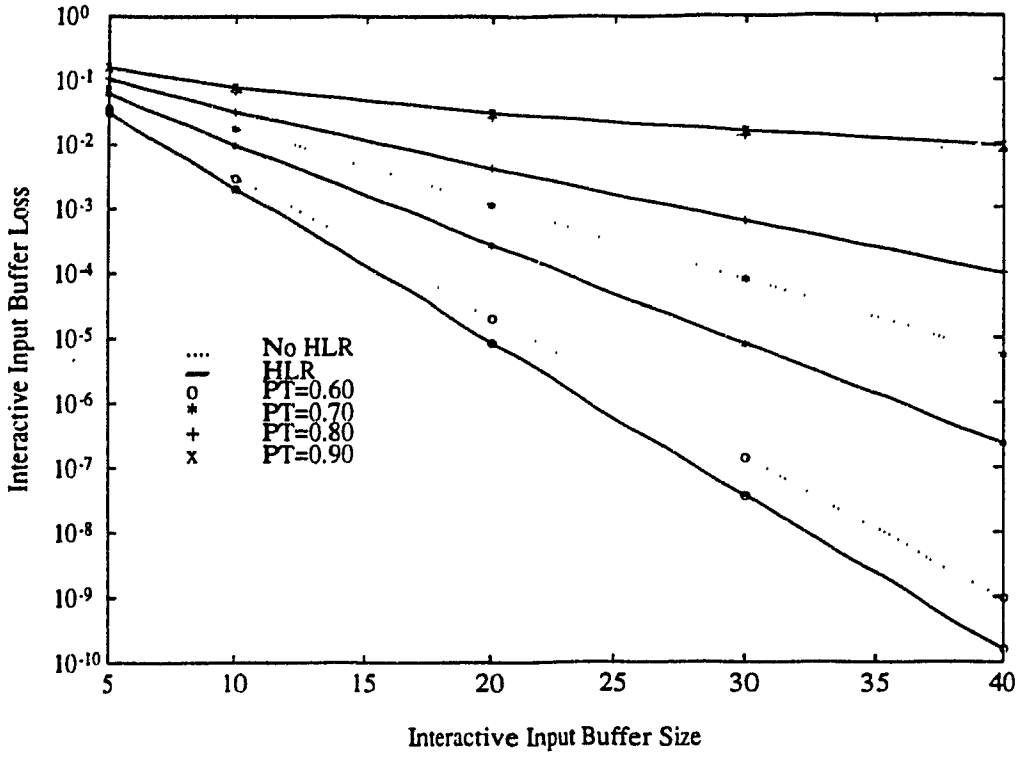


Fig. 3.45. Interactive input buffer loss with increasing buffer size (L=2).

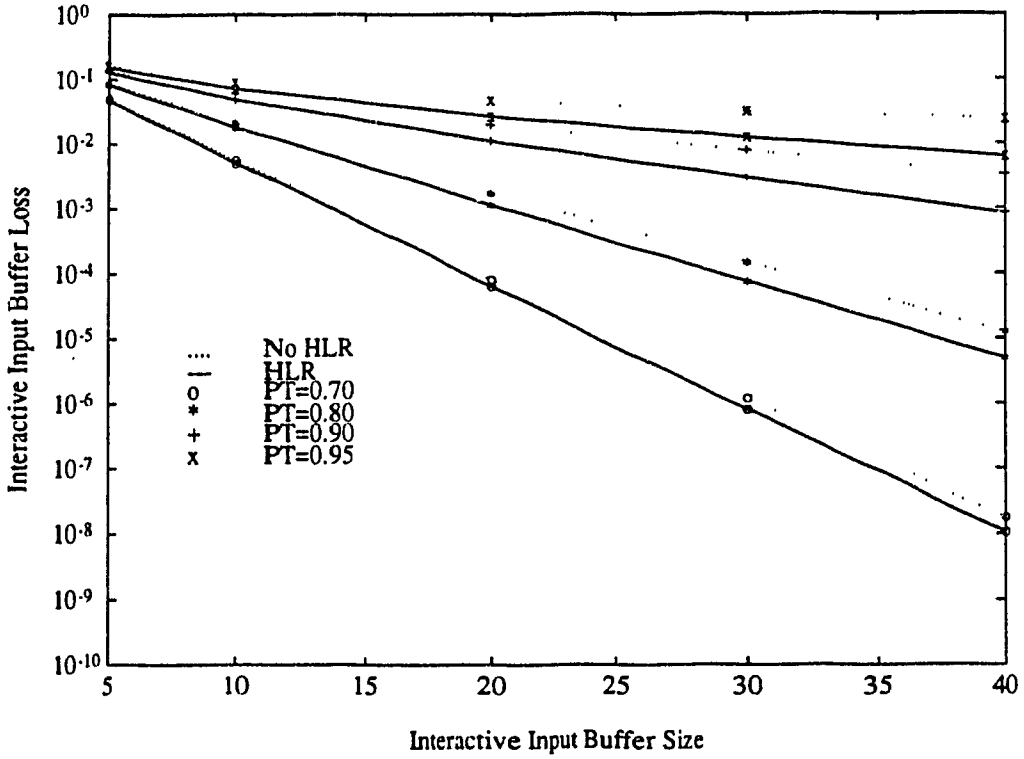


Fig. 3.46. Interactive input buffer loss with increasing buffer size (L=3).

this load. However, at higher loads of 0.8 and 0.9 the buffer has to be increased to 33 and 41 respectively without HLR, from 30 with HLR, in order to maintain the same loss.

3.5.4. Output Buffer Design Under a Variety of Different Traffic Conditions

At the output buffer the minimum buffer size for a certain overflow has to be determined for the ratio of the downlink to the uplink rate, *DLR*, as the traffic load increases. The output buffer loss should be ideally designed to be much less than the loss at the input so that the input remains the bottleneck of the system. It would not make any sense to increase the switch throughput by increasing the speedup or by using HLR and then have to discard packets at the output buffer due to overflow. To this end, the output buffer loss is plotted versus the buffer size for different traffic loads in Figures 3.47-3.49 for L=2,3,4 and for various values of *DLR*. As shown in Fig. 3.47 for the case where L=2, a minimum output buffer length of between 11 and 20 is sufficient for a load of 0.6 to maintain a loss between $1x10^{-6}$ and $1x10^{-11}$ respectively. However, as the load is increased to 0.7 and 0.8 the buffer size has to be drastically increased. The large amount of traffic arriving at the output port is due to input buffer saturation at these loads and realistically the switch system cannot be operated in this region. When the switch speedup is increased the traffic to any one output under uniform conditions increases resulting in higher losses, as shown in Fig. 3.48. However, the losses are drastically reduced by increasing *DLR* to two. With a minimum buffer length of 10 the loss probability can be reduced to at least $1x10^{-9}$ for a load as high as 0.95. The results for the case with L=4 are shown in Fig. 3.49. Here it is seen that a buffer length of 15 is sufficient with *DLR*=2

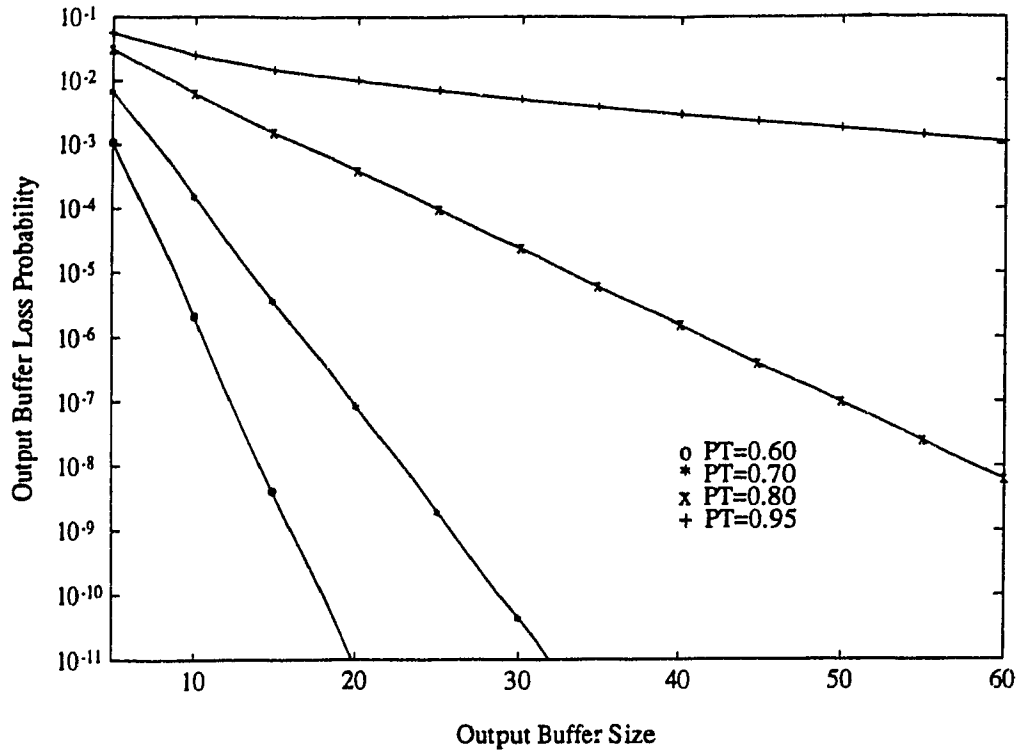


Fig. 3.47. Output buffer loss (L=2).

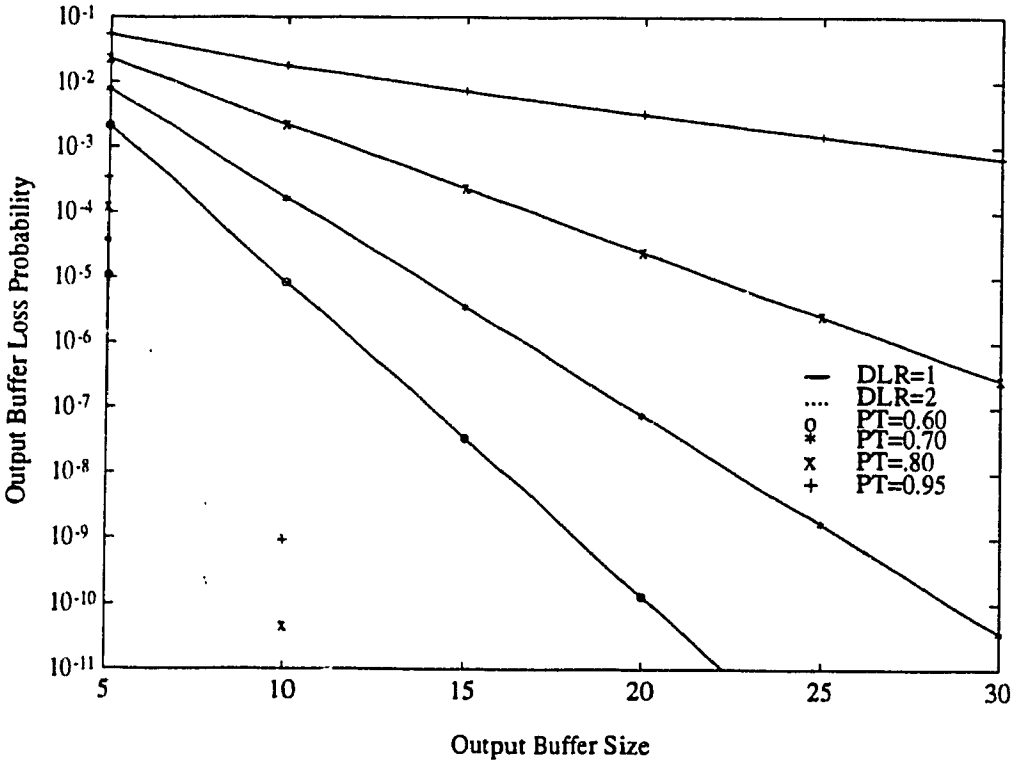


Fig. 3.48. Output buffer loss (L=3).

to reduce the loss to $1x10^{-11}$ for loads up to 0.9, while for *DLR*=3 a buffer length of 9 is needed. From these graphs it is seen that a *DLR* of two drastically reduces the buffering required at each output port, however increasing *DLR* further for higher switch speedups is not necessary under uniform traffic conditions.

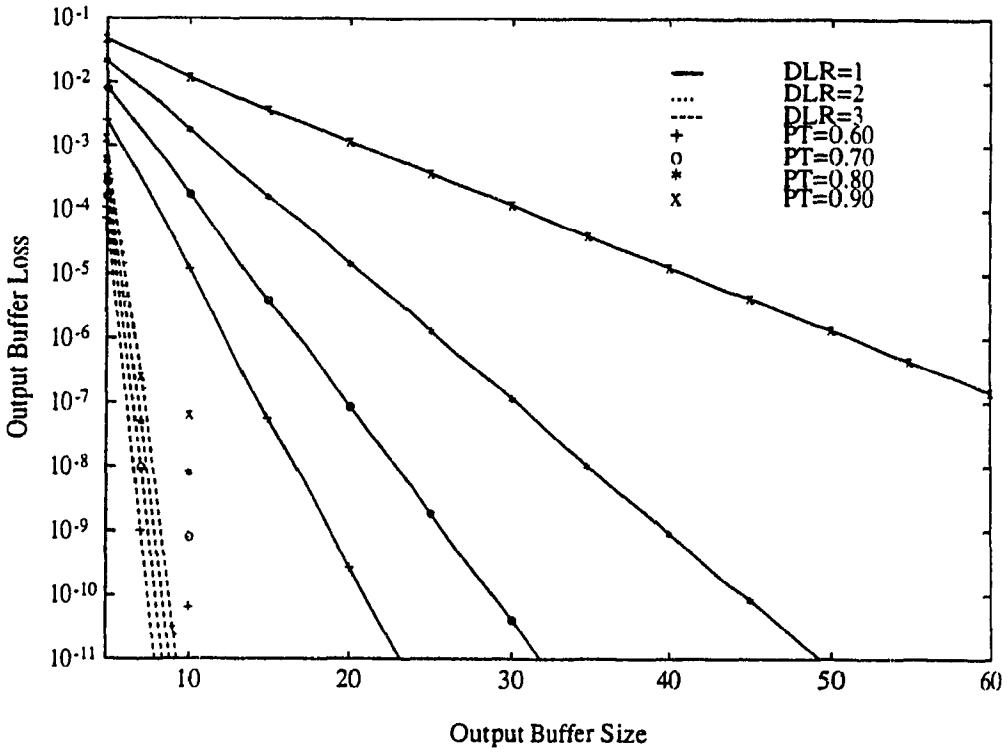


Fig. 3.49. Output buffer loss (L=4).

The output buffer delays for the various *DLR* values and switch speedups are shown in Fig. 3.50. It is assumed the buffer size is 30. Here it is seen that for *DLR*=1 as the switch speedup is increased from one to three, the maximum load decreases from 0.85 to 0.7 respectively due to the increased traffic. However, with *DLR*=2 the delay hardly changes except at very high loads so that there is very little improvement in increasing *DLR* above two under uniform traffic conditions.

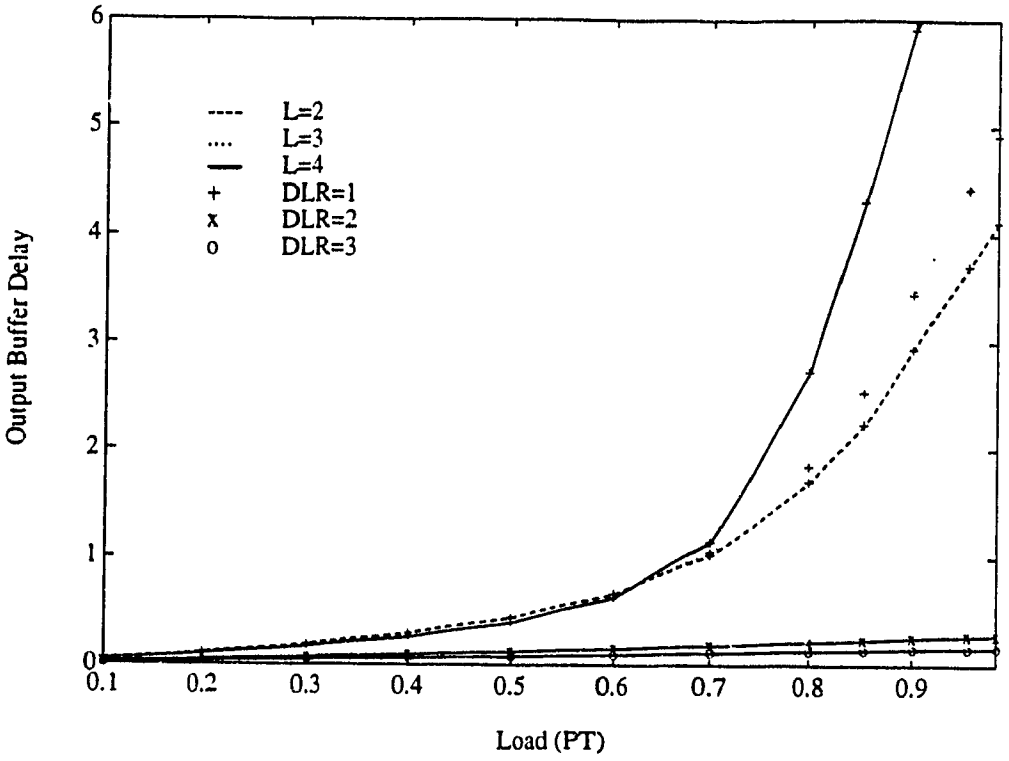


Fig. 3.50. Output buffer delays with L=2,3,4 and *DLR*=1,2,3.

Chapter 4

Performance Under Hotspot Traffic Conditions

4.1. Characterizing Nonuniform Traffic

Nonuniform traffic patterns are expected to arise frequently in an integrated services environment and the on-board switching system has to be designed to ensure a minimum performance for each service. Under the most general form of nonuniform traffic each input i may go to any output j with probability d_{ij}. Therefore there are virtually an infinite number of patterns. Of the many patterns however those of interest are those which arise frequently and which cause severe congestion. One of the most general congestive patterns that is often analyzed is G1 (first general case) traffic. Under G1 traffic, as shown in Fig. 4.1, both the input and the output ports are partitioned into separate groups and each input group is assumed to send a proportion of its traffic to one of the output groups. The N input and output ports are partitioned into two groups (D_1,D_2) and (O_1,O_2) respectively. As seen the D_1 ports transmit to O_1 with probability P_{11} and to O_2 with probability P_{12}, while the D_2 ports are assumed to transmit to O_1 with probability P_{21} and to O_2 with probability P_{22}. One of the most congestive patterns that is subset of G1 is the hotspot pattern. Under a hotspot traffic pattern a proportion of the traffic from each input goes to one output, the hotspot, while the rest of the traffic is uniformly distributed over all the outputs. Hotspot traffic can be characterised by D_1=N, D_2=0, O_1=1, O_2=N-1 and $P_{11}=H+\frac{(1-H)}{N}$, $P_{12}=\frac{(1-H)(N-1)}{N}$, where H is the proportion of the traffic that goes to the hotspot output port. Hotspot traffic is likely to arise in many

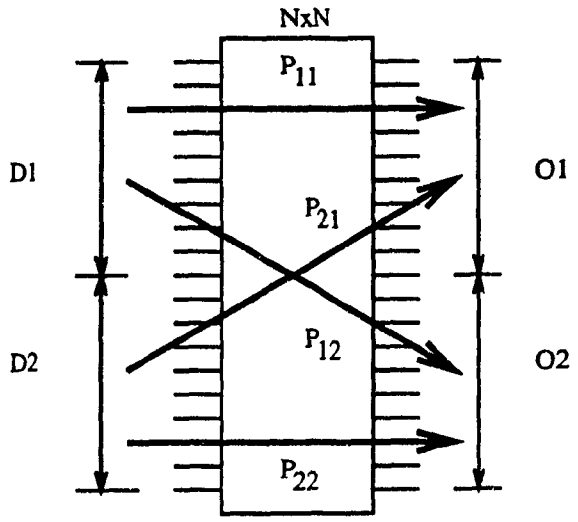


Fig. 4.1. G1 nonuniform traffic pattern.

situations where users share a common resource such as a central computing facility or other information source. Under hotspot traffic the distribution of the total number of packets arriving at the hotspot output due to the hotspot portion as well as from the uniform traffic is given by

$$P_k[\text{hotspot}] = \binom{N}{k}\left(Hp + \frac{(1-H)p}{N}\right)^k \left(1 - Hp - \frac{(1-H)p}{N}\right)^{N-k} \tag{4.1}$$

where p is the probability of a packet arriving at an input port per time slot. As seen the mean number of arrivals is no longer independent of N.

Another subset of the G1 traffic pattern is point-to-point traffic in which one input transmits to one output continuously while all the other inputs are assumed to transmit uniformly to all the outputs. Point-to-point is characteristic of video and voice users which

maintain high rate connections over long durations. Point-to-point traffic represents the case when D_1=1, D_2=N-1, O_1=1, O_2=N-1 and P_{11}=1, P_{12}=0, $P_{21} = \frac{1}{N}$, $P_{22} = \frac{N-1}{N}$. The distribution of the number of packets arriving at the point-to-point output is

$$P_k[\text{P-T-P}] = (1-p)\binom{N-1}{k}\left(\frac{p}{N}\right)^k\left(1-\frac{p}{N}\right)^{N-1-k} + p\binom{N-1}{k-1}\left(\frac{p}{N}\right)^{k-1}\left(1-\frac{p}{N}\right)^{N-k}. \tag{4.2}$$

The first term represents the probability that the point-to-point is not active while the second term represents the probability that the point-to-point is active. In either case, the other N-1 inputs are uniformly distributed to all the outputs. The value of k can only equal N when the point-to-point is active and that is why the binomial chooses k-1 out of N-1 in the second term. In the first term k can be at most N-1. It is seen that since there is only one point-to-point link, the number of arrivals and hence the loss quickly converges as N is increased.

Still another pattern which is often examined and is a subset of G1, is G2 (second general case) traffic. Under balanced G2 traffic D_1=N, $O_1 = O_2 = \frac{N}{2}$ and P_{11}=r, P_{12}=(1-r) where $0.5 \le r \le 1$, while under unbalanced conditions the output groups are assumed to be different sizes. Thus, under G2 traffic all the inputs have a preference for one group of outputs. Such situations may arise in a satellite network due to time zones with heavier traffic going to regions where the work day has not yet ended and less traffic going to regions where it has. The distribution at the group one outputs is given by

$$P_k[\text{G2 (O1)}] = \binom{N}{k}\left(\frac{2rp}{N}\right)^k\left(1-\frac{2rp}{N}\right)^{N-k} \tag{4.3}$$

while at the group two outputs the distribution is given by

$$P_k\,[\mathrm{G2\,(O2)}] = \binom{N}{k}\left(\frac{2(1-r)p}{N}\right)^k\left(1-\frac{2(1-r)p}{N}\right)^{N-k} \qquad (4.4)$$

In both cases, as can be seen, the mean of the distribution is independent of N so that the loss converges for an internally nonblocking switch as N goes to infinity.

4.2. Performance of the Banyan and Knockout Switches Under Nonuniform Traffic

4.2.1. Banyan Switch Under Nonuniform Loads

It has been shown that under nonuniform traffic that the performance of the buffered banyan switch degrades substantially due to both the internal blocking as well as the blocking at the output. Garg and Huang [49] showed that under both balanced and unbalanced G2 traffic conditions with N=16 that the Banyan switch severely degrades in performance as the traffic increases, as seen in Fig. 4.2 and Fig. 4.3 respectively. Under the unbalanced condition it was assumed that O_1=4 and O_2=12. Here the numbers in the brackets refer to P_{11} and P_{12}. As seen the unbalanced load has a lower performance due to the higher blocking and that as all the traffic goes to O_1 the throughput decreases to 0.15. The performance under hotspot traffic with a buffer size of four was also examined by Pfister and Norton [50]. The simulation results for N=16, as seen in Fig. 4.4, show that the performance was very dependent on the traffic and that for even small hotspot ratios of H=0.04 and 0.08 the throughput was limited to only 0.27 and 0.1 respectively. Kim and Leon-Garcia [51] have also recently shown that the throughput under point-to-point traffic for the single buffered Banyan network decreases rapidly as the number of

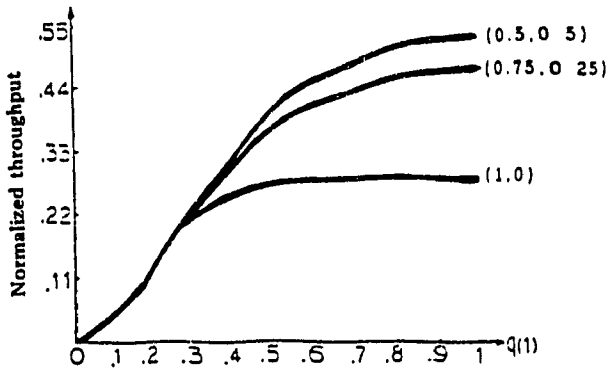


Fig. 4.2. Performance of the Banyan switch under balanced G2 traffic [49].

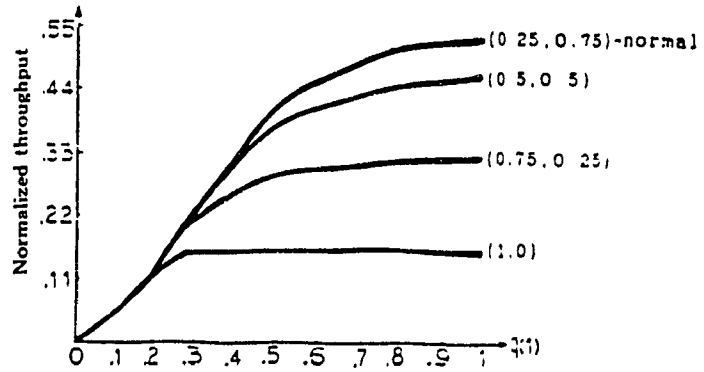


Fig. 4.3. Performance of the Banyan switch under unbalanced G2 traffic [49].

stages increase limiting the switch size. In their paper, two types of point-to-point traffic were examined. In the first type each input i only transmits to output j for $i=j$. The performance of this pattern was examined for the case when there is a maximum number of internal conflicts. As an example, the maximum conflict pattern for N=16 is shown in Fig. 4.5. As can be seen there is a conflict at each node and packets have to be queued at each stage. The throughput performance under this type of conflict pattern degrades rapidly. As the switch size is increased from N=8 to 32 to 64 the throughput decreases from 0.49 to 0.25 to 0.12 respectively. In the second pattern that was examined one input-output pair formed a point-to-point link while the other inputs were uniformly distributed to the other N-1 outputs. As seen in Fig. 4.6, as the single source single destination (point-to-point) traffic increases the uniform background throughput decreases rapidly. Again, the performance is heavily dependent on N. The corresponding delay graphs for the uniform background and the point-to point traffics are shown in Fig. 4.7.

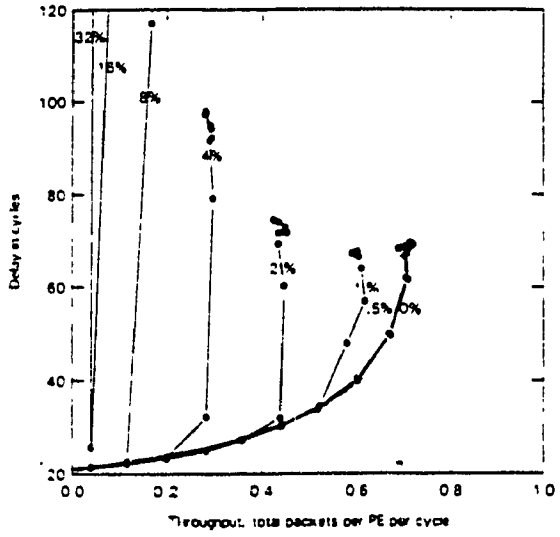


Fig. 4.4 Buffered Banyan performance under hotspot traffic [50].

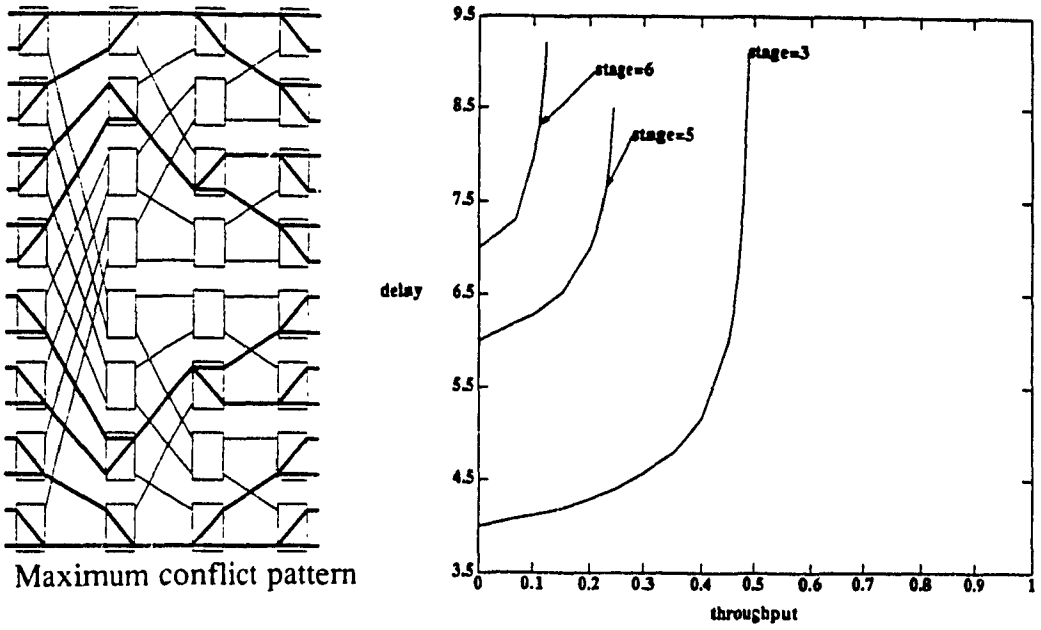


Fig. 4.5. Performance under maximum conflict point-to-point traffic [51].

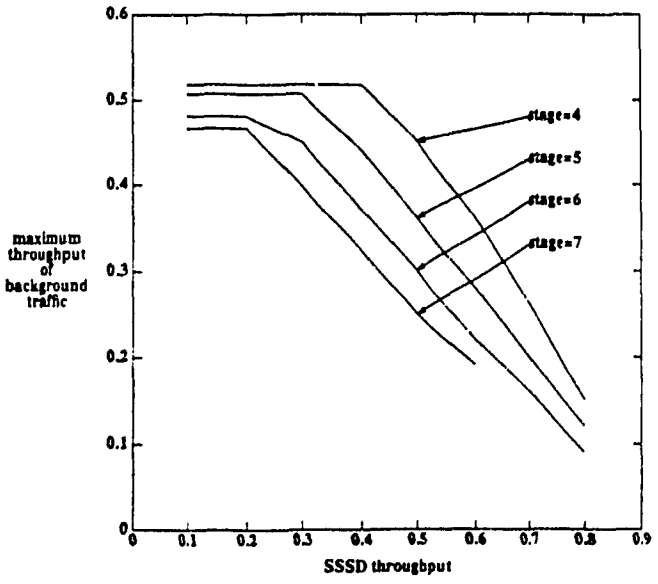


Fig. 4.6. Throughput characteristics with one point-to-point link [51].

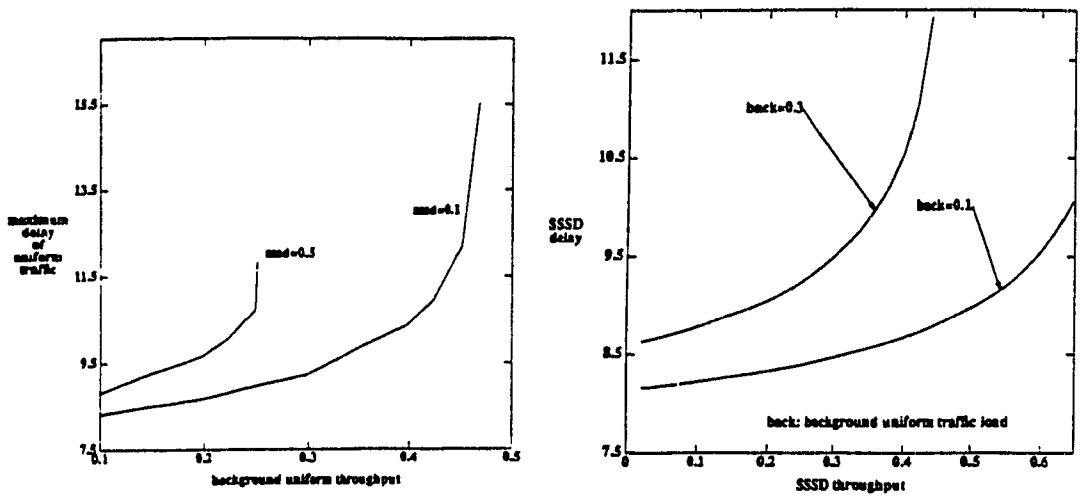


Fig. 4.7. Delay characteristics for the background and point-to-point traffic (stages=7) [51].

4.2.2. Knockout Switch Under Nonuniform Loads

In order to increase the throughput under nonuniform loads the Knockout switch was analyzed by Yoon, Liu and Lee [52]. In their paper they analyzed hotspot, point-to-point as well as the balanced G2 traffic pattern. The hotspot results are shown in Fig. 4.8 and Fig. 4.9. It is seen that while the performance still degraded severely as the hotspot traffic and the switch size increased, for smaller switches increasing the speedup reduced the loss substantially. Under point-to-point traffic it was shown, as seen in Fig. 4.10 and Fig. 4.11, that with L=9 the loss could be reduced to $1x10^{-6}$ at a load of 0.90. It was also

shown that the loss was independent of N and that L had to be increased by at most one to maintain a loss of $1x10^{-6}$ between a load of 70% and 100%. The results under balanced G2 traffic as seen in Fig. 4.12 showed that the loss as N goes to infinity could be reduced to $1x10^{-6}$ by increasing L to eleven.

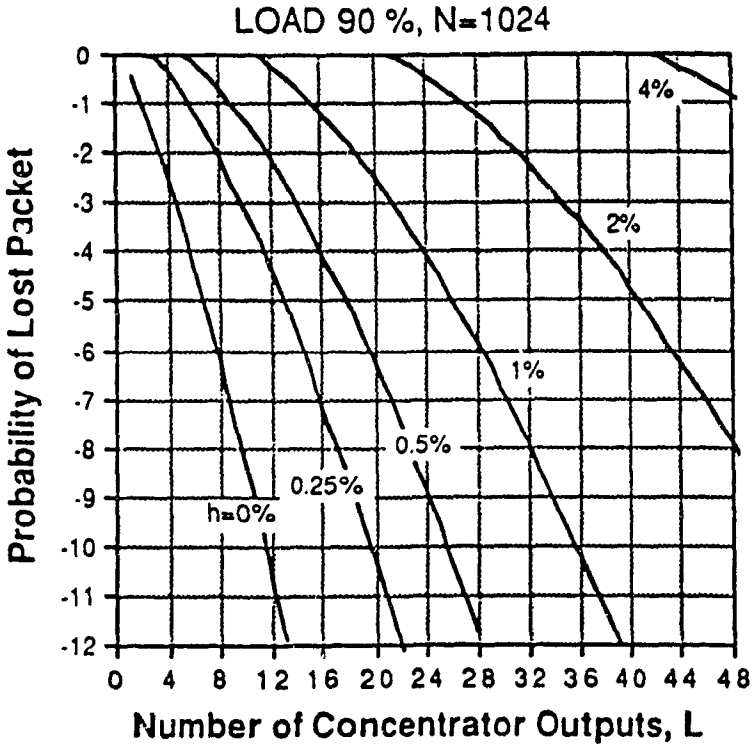


Fig. 4.8. Knockout performance with N=1024 under hotspot traffic [52].

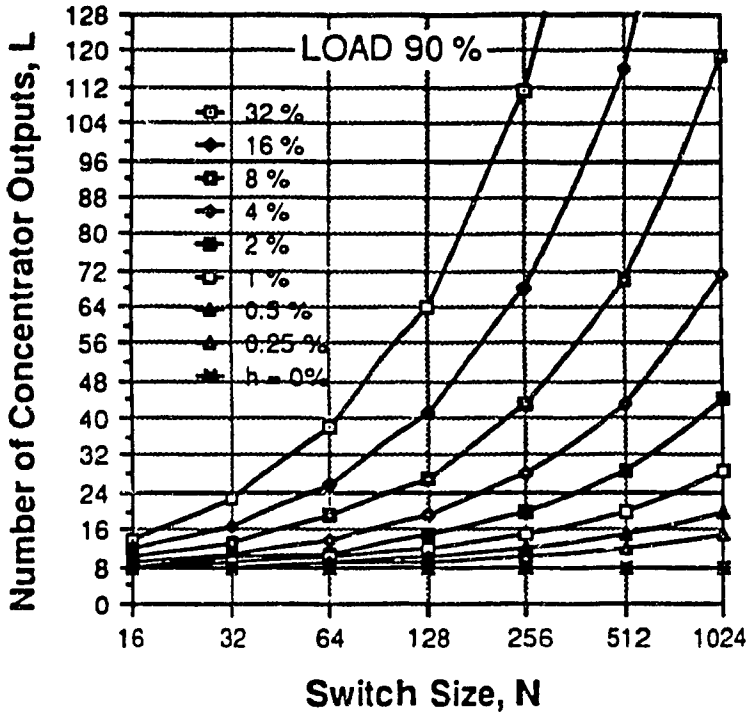


Fig. 4.9. Speedup required to maintain a loss of 1×10^{-6} as N increases [52].

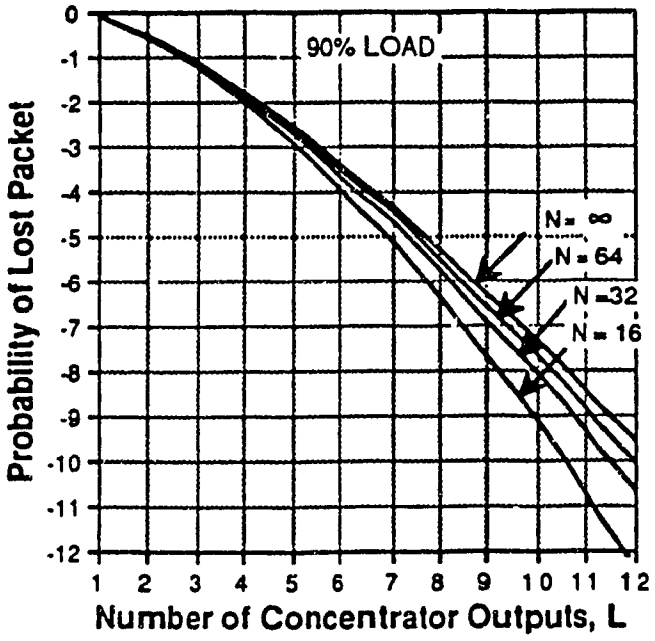


Fig. 4.10. Knockout performance under point-to-point traffic [52].

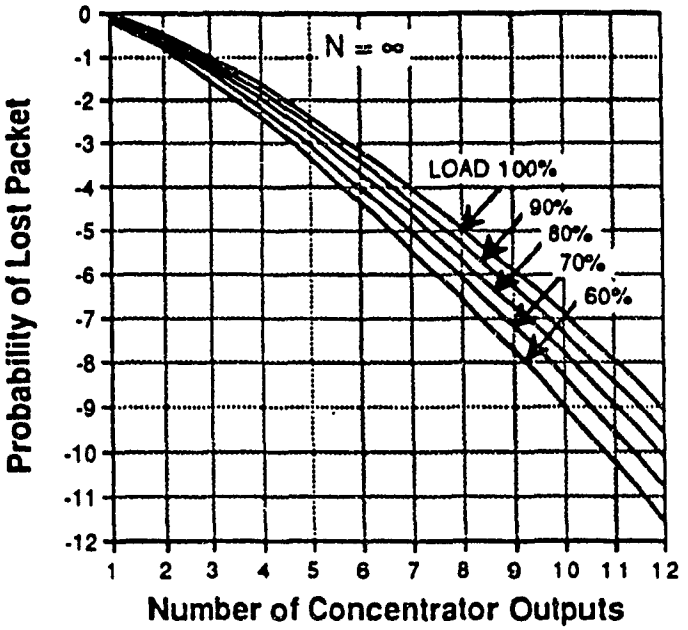


Fig. 4.11. Knockout point-to-point performance as the load is increased [52].

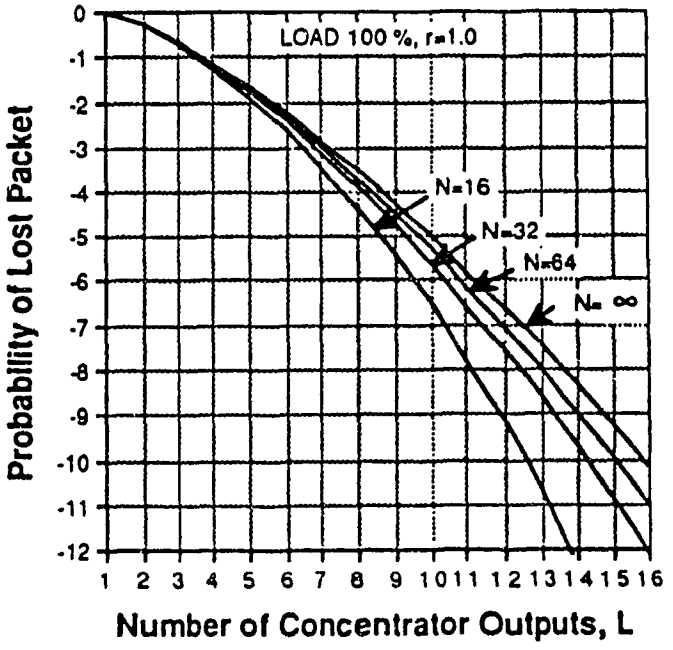


Fig. 4.12. Knockout balanced G2 performance with r=1 [52].

4.2.3. Internally Nonblocking Switches with Buffering Under Nonuniform Loads

In order to reduce the required speedup for a certain loss probability input buffering was examined by Li and Lee [53] for a generic internally nonblocking switch, while both input and output buffering were examined by Chen and Stern [54]. Both papers considered the G2 traffic pattern and examined the case with L=1, while Chen and Stern also examined the L=4 case. In both papers the results for the L=1 case showed that while the blocking is not as severe as that of the Banyan switch, the limited speedup also limits the throughput. The results from Chen and Stern, as seen in Fig. 4.13, show that the degradation in performance is very dependent on the intensity ratio, which is $\frac{P_{11}}{P_{12}}$, and the normalized group size which is just $\frac{O_1}{N}$. As the amount of traffic increased to a decreasing output group size the blocking probability increased and hence the throughput decreased. Increasing the speedup to four, as shown in Fig. 4.14, however substantially improved the performance. Even at high traffics for a small group of outputs the throughput was not significantly impaired. Chen and Stern used the same method described in section 3.2 for the nonuniform analysis except that the probability of joining one of the virtual M/M/L queues was different for each output. By assuming that the probability of going to output i was G_i, the arrival rate at each output was found as $\lambda_j^o = G_i \sum_{i=1}^{N} \lambda_j'$. For the balanced input case $\lambda_j' = \lambda$. Replacing λ by λ_j^o in eq. 3.15, the mean service time, $\overline{W_{S_j}}$, was found for each output. The total mean service time, $\overline{W_s}$, was then found by averaging $\overline{W_{S_j}}$ over G_i.

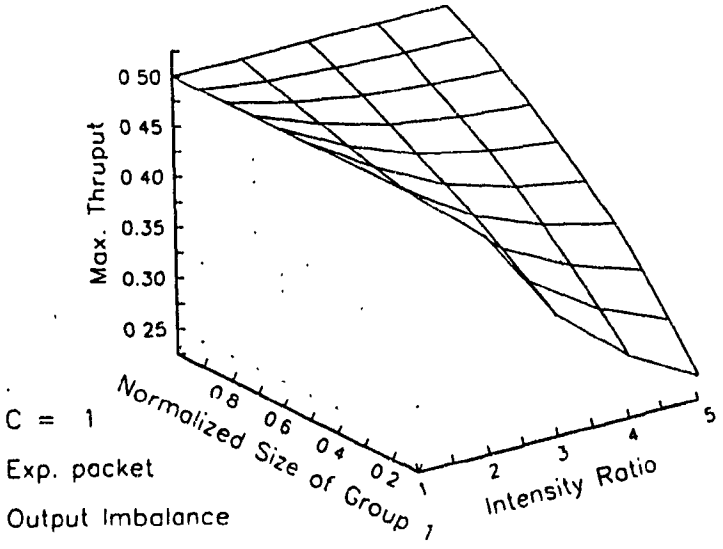


Fig. 4.13. Generic nonblocking switch performance with L=1 under G2 traffic with input and output buffering [54].

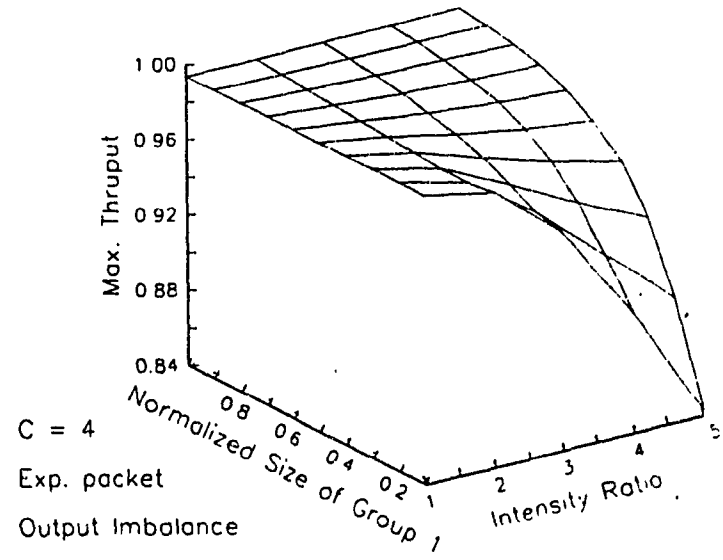


Fig. 4.14. Generic nonblocking switch performance with L=4 under G2 traffic with input and output buffering [54].

4.3. Performance of the Modified Knockout Switch Under Hotspot Traffic

The hotspot traffic pattern is one of the most severe traffic patterns which occurs frequently in many applications and is of special concern to a switched satellite system. One of the scenarios in which a satellite system might experience hotspot traffic is if many voice connections are extremely biased to one region due to time zones or other events such as emergencies. In the case of video users, hotspot traffic situations may arise in future dial-up video services in which many different transmissions may be selected by users in a certain region at certain times. In addition, interactive and file data users which are accessing a central computing facility will also encounter hotspot scenarios.

Hotspot traffic is the most severe type of nonuniform traffic in the sense that a proportion of each input port's traffic goes to a hotspot output and this proportion may be somewhere between zero and one, and more importantly may change sporadically due to unforeseen traffic needs. Since all the inputs send a proportion of their traffic to the hotspot the mean number of packets arriving at the hotspot is dependent on N. The number of packets arriving at a hotspot output is binomially distributed with a mean of $NHp+(1-H)p$. Here the first term, NHp, is the traffic generated due to the hotspot traffic and is dependent on N and the second term is the contribution due to the uniformly distributed traffic that also arrives at the hotspot which is independent of N. Therefore, for a given value of H the larger the size of the switch N the more speedup L is required to satisfy a minimum blocking probability. As was shown, the performance without buffering in a hotspot environment severely degrades as the switch size increases and as the hotspot traffic increases. In this section the results are extended for the input and

output buffered case in the hope of reducing the required speedup and the dependence on N. In particular it is of interest to see what effect priority has on the performance and to see what improvements the HLR algorithm affords.

4.3.1. Hotspot Analysis

As in the case with uniform traffic the switch is analyzed with and without HLR. In analyzing the hotspot case the distributions of the number of packets that arrive at the uniform and hotspot outputs are used to determine the blocking due to the switch for each case. These blocking probabilities are then averaged to find the mean blocking probability for the different services at the input to the switch. Using these blocking probabilities the discrete input buffer chains are solved and the new probability distributions found. Using the iterative procedure the distributions are then used to recalculate the blocking probability. Our computations have shown that only a few iterations are generally required before the blocking probability converges and a steady state reached. The distributions of the number of users going to a uniform output are given in equations (4.5)-(4.8) for video, voice, file data and interactive data respectively. Taking the video case, Wu_i, as an example the $\left(1-\Phi_{W_0}\right)R_W\left(1-H_W\right)$ term represents the traffic at a video input that is destined to any one output. The $\left(1-\Phi_{W_0}\right)R_W$ term represents the probability that there is something in the server upon service and $\left(1-H_W\right)$ is the proportion of the traffic that is uniformly distributed.

$$Wu_i = \binom{N}{i}\left[\frac{\left(1-\Phi_{W_0}\right) R_W \left(1-H_W\right)}{N}\right]^i \left[1-\frac{\left(1-\Phi_{W_0}\right) R_W \left(1-H_W\right)}{N}\right]^{N-i} \qquad (4.5)$$

$$Vu_i = \binom{N}{i}\left[\frac{\left(1-\Phi_{V_0}\right) R_V \left(1-H_V\right)}{N}\right]^i \left[1-\frac{\left(1-\Phi_{V_0}\right) R_V \left(1-H_V\right)}{N}\right]^{N-i} \tag{4.6}$$

$$FDu_i = \binom{N}{i}\left[\frac{\left(1-\Phi_{FD_0}\right) R_{FD} \left(1-H_{FD}\right)}{N}\right]^i \left[1-\frac{\left(1-\Phi_{FD_0}\right) R_{FD} \left(1-H_{FD}\right)}{N}\right]^{N-i} \tag{4.7}$$

$$IDu_i = \binom{N}{i}\left[\frac{\left(1-\Phi_{ID_0}\right) R_{ID} \left(1-H_{ID}\right)}{N}\right]^i \left[1-\frac{\left(1-\Phi_{ID_0}\right) R_{ID} \left(1-H_{ID}\right)}{N}\right]^{N-i} \tag{4.8}$$

The distributions of the number of users that are going to the hotspot output for the different traffics are given in equations (4.9)-(4.12). Taking the video case, Wh_i, as an example the $H_W + \frac{\left(1-H_W\right)}{N}$ term represents the proportion of the traffic at an input that goes to the hotspot output. The first term, H_W, is the probability of going to the hotspot and the second term, $\frac{\left(1-H_W\right)}{N}$, is the probability that the uniform portion also goes to the hotspot instead of one of the other N-1 outputs. Again the distribution takes on a binomial distribution due to the fact that all the inputs transmit H_W percent of their traffics to the hotspot and $\left(1-H_W\right)$ uniformly to all the outputs.

$$Wh_i = \binom{N}{i}\left[\left(1-\Phi_{W_0}\right) R_W \left(H_W+\frac{\left(1-H_W\right)}{N}\right)\right]^i \left[1-\left(1-\Phi_{W_0}\right) R_W \left(H_W+\frac{\left(1-H_W\right)}{N}\right)\right]^{N-i} \tag{4.9}$$

$$Vh_i = \binom{N}{i}\left[\left(1-\Phi_{V_0}\right) R_V \left(H_V+\frac{\left(1-H_V\right)}{N}\right)\right]^i \left[1-\left(1-\Phi_{V_0}\right) R_V \left(H_V+\frac{\left(1-H_V\right)}{N}\right)\right]^{N-i} \tag{4.10}$$

$$FDh_i = \binom{N}{i}\left[\left(1-\Phi_{FD_0}\right) R_{FD} \left(H_{FD}+\frac{\left(1-H_{FD}\right)}{N}\right)\right]^i \left[1-\left(1-\Phi_{FD_0}\right) R_{FD} \left(H_{FD}+\frac{\left(1-H_{FD}\right)}{N}\right)\right]^{N-i} \tag{4.11}$$

$$IDh_i = \binom{N}{i}\left[\left(1-\Phi_{ID_0}\right) R_{ID} \left(H_{ID}+\frac{\left(1-H_{ID}\right)}{N}\right)\right]^i \left[1-\left(1-\Phi_{ID_0}\right) R_{ID} \left(H_{ID}+\frac{\left(1-H_{ID}\right)}{N}\right)\right]^{N-i} \tag{4.12}$$

The blocking probability for the uniform and hotspot outputs can now be found using the same equations (3.25)-(3.28) as for the uniform traffic case, with the uniform distributions of equations (3.21)-(3.24) replaced by equations (4.5)-(4.8) for the uniform outputs and equations (4.9)-(4.12) for the hotspot outputs. Denoting the blocking at the uniform outputs by *Pbu* and the blocking at the hotspot outputs by *Pbh*, the mean blocking probability can now be found for each of the services respectively by averaging the uniform and hotspot blocking probabilities as shown in equations (4.13)-(4.16). In each of these equations the probability of going to either the hotspot or uniform output is multiplied by the blocking probability for that respective output. The sum then represents the mean blocking probability of the two possible events, either going to the hotspot and getting blocked or going to the uniform output and getting blocked.

$$Pmb_W = \left(H_W+\frac{\left(1-H_W\right)}{N}\right) Pbh_W + \left(1-H_W\right)\left(\frac{N-1}{N}\right) Pbu_W \tag{4.13}$$

$$Pmb_V = \left(H_V + \frac{(1-H_V)}{N}\right) Pbh_V + (1-H_V)\left(\frac{N-1}{N}\right) Pbu_V \tag{4.14}$$

$$Pmb_{FD} = \left(H_{FD} + \frac{(1-H_{FD})}{N}\right) Pbh_{FD} + (1-H_{FD})\left(\frac{N-1}{N}\right) Pbu_{FD} \tag{4.15}$$

$$Pmb_{ID} = \left(H_{ID} + \frac{(1-H_{ID})}{N}\right) Pbh_{ID} + (1-H_{ID})\left(\frac{N-1}{N}\right) Pbu_{ID} \tag{4.16}$$

With the initial mean blocking probabilities found the service probabilities for the input buffers for the case without HLR are given by the following equations for video, voice, file data and interactive data respectively.

$$U_W = (1-Pmb_W)\, R_W \tag{4.17}$$

$$U_V = (1-Pmb_V)\, R_V \tag{4.18}$$

$$U_{FD} = (1-Pmb_{FD})\, R_{FD} \tag{4.19}$$

$$U_{ID} = (1-Pmb_{ID})\, R_{ID} \tag{4.20}$$

To find the respective service probabilities for file and interactive data with HLR the probability of having a sister packet along with the blocking probability of the sister packet first has to be found. In order to find the blocking probability for sister packets the distributions of the number of empty spaces are first derived at the hotspot and uniform outputs in a similar manner as was done in Chapter 3. The distribution of the number of empty spaces left at the hotspot output before the sister packets compete is given by

$$Eh_{FD_i} = \binom{N}{L-i} \left(\Psi_h\right)^{L-i} \left(1-\Psi_h\right)^{N-L+i} \qquad i=1,2,3,\ldots,L$$
$$Eh_{FD_0} = 1 - \sum_{k=1}^{L} Eh_{FD_k} \tag{4.21}$$

where

$$\Psi_h = \left(1-\Phi_{W_0}\right) R_W \left(H_W + \left(\frac{1-H_W}{N}\right)\right) + \left(1-\Phi_{V_0}\right) R_V \left(H_V + \left(\frac{1-H_V}{N}\right)\right)$$
$$+ \left(1-\Phi_{FD_0}\right) R_{FD} \left(H_{FD} + \left(\frac{1-H_{FD}}{N}\right)\right) + \left(1-\Phi_{ID_0}\right) R_{ID} \left(H_{ID} + \left(\frac{1-H_{ID}}{N}\right)\right)$$

Here Ψ_h denotes the mean traffic from all the different services that is present at an input port of the switch which is going to the hotspot output. Ψ_h is found by weighting the probability of having a packet which goes to the hotspot output for each service by the probability of selecting the buffer. Equation (4.21) follows the same principle described earlier for eq. (3.31). Similarly the distribution of empty spaces at the uniform output is given by

$$Eu_{FD_i} = \binom{N}{L-i} \left(\frac{\Psi_u}{N}\right)^{L-i} \left(1-\frac{\Psi_u}{N}\right)^{N-L+i} \qquad i=1,2,3,\ldots,L$$
$$Eu_{FD_0} = 1 - \sum_{k=1}^{L} Eu_{FD_k} \tag{4.22}$$

where

$$\Psi_u = \left(1-\Phi_{W_0}\right) R_W \left(1-H_W\right) + \left(1-\Phi_{V_0}\right) R_V \left(1-H_V\right)$$
$$+ \left(1-\Phi_{FD_0}\right) R_{FD} \left(1-H_{FD}\right) + \left(1-\Phi_{ID_0}\right) R_{ID} \left(1-H_{ID}\right)$$

represents the mean traffic at an input port destined to a uniform output. Ψ_u is divided

by N in eq. (4.22) since the mean traffic as defined may go to any of the N outputs including the hotspot output.

The corresponding distribution of the number of empty spaces left at the hotspot output port for the interactive sister packets is found by averaging Eh_{FD_i} over all the possibilities that a given number of file sister packets will take the remaining empty slots available as follows

$$Eh_{ID_i} = \sum_{k=i}^{L} Eh_{FD_k} \binom{N}{k-i} \left(\Psi_{h_{FD}}\right)^{k-i} \left(\Psi_{h_{FD}}\right)^{N-k+i} \qquad i=1,2,3,....,L$$
$$Eh_{ID_0} = 1 - \sum_{k=1}^{L} Eh_{ID_k} \tag{4.23}$$

where

$$\Psi_{h_{FD}} = R_{FD}\, Pmb_{FD}\, Ps_{FD} \left(H_{FD} + \frac{\left(1-H_{FD}\right)}{N}\right)$$

$$Ps_{FD} = \left(1 - \Phi_{FD_0} - \Phi_{FD_1}\right)$$

$\Psi_{h_{FD}}$ is the traffic generated at each input due to file sister packets. This traffic is the probability that a file packet is blocked, Pmb_{FD}, a sister packet is available, Ps_{FD}, and that the sister packet goes to the hotspot output. Similarly the distribution of empty spaces at the uniform outputs is given by

$$Eu_{ID_i} = \sum_{k=i}^{L} Eu_{FD_k} \binom{N}{k-i} \left(\Psi_{u_{FD}}\right)^{k-i} \left(\Psi_{u_{FD}}\right)^{N-k+i} \qquad i=1,2,3,....,L$$
$$Eu_{ID_0} = 1 - \sum_{k=1}^{L} Eu_{ID_k} \tag{4.24}$$

where

$$\Psi_{u_{FD}} = R_{FD}\, Pmb_{FD}\, Ps_{FD} \left(\frac{1 - H_{FD}}{N} \right).$$

Here Eu_{FD_k} is averaged over the possibility that up to L file sister packets may go a uniform output buffer and use some or all of the remaining empty spaces. $\Psi_{u_{FD}}$ is the amount of file sister traffic to go to one of the uniform outputs. $\Psi_{u_{FD}}$ represents the probability that a file gets blocked either at the hotspot output or at a uniform output in the first round and that there is a sister packet available for the second round competition which is going to one of the uniform outputs.

The sister packet blocking probabilities can now be found using the method described previously for the uniform traffic case. The file sister packet blocking probability for the uniform output can then be written as

$$Psbu_{FD} = \frac{\sum_{k=0}^{L} Eu_{FD_k} \sum_{i=k}^{N} (i-k) \binom{N}{i} \left(\Psi_{u_{FD}}\right)^i \left(1 - \Psi_{u_{FD}}\right)^{N-i}}{\sum_{i=0}^{N} i \binom{N}{i} \left(\Psi_{u_{FD}}\right)^i \left(1 - \Psi_{u_{FD}}\right)^{N-i}}. \qquad (4.25)$$

$Psbu_{FD}$ is found by averaging the probability that a packet is blocked given that there are k empty spaces left at the uniform output over the probability distribution for the number of empty spaces. $\Psi_{u_{FD}}$ is the amount of sister packet traffic going to a uniform output as previously defined. Similarly, the sister blocking probability for at the hotspot output is found as

$$Psbh_{FD} = \frac{\sum_{k=0}^{L} Eh_{FD_k} \sum_{i=k}^{N} (i-k) \binom{N}{i} \left(\Psi_{h_{FD}}\right)^i \left(1-\Psi_{h_{FD}}\right)^{N-i}}{\sum_{i=0}^{N} i \binom{N}{i} \left(\Psi_{h_{FD}}\right)^i \left(1-\Psi_{h_{FD}}\right)^{N-i}}. \quad (4.26)$$

Here the distribution of the number of empty spaces at the hotspot before the file sister packets compete, Eh_{FD_k}, is used along with the hotspot file sister traffic, $\Psi_{h_{FD}}$, to find the blocking probability.

The interactive sister packet blocking probability is next calculated using the distribution of the number of empty spaces at the uniform and hotspot outputs, Eu_{ID_k} and Eh_{ID_k} respectively. The interactive sister packet blocking probability at the uniform output is then given by

$$Psbu_{ID} = \frac{\sum_{k=0}^{L} Eu_{ID_k} \sum_{i=k}^{N} (i-k) \binom{N}{i} \left(\Psi_{u_{ID}}\right)^i \left(1-\Psi_{u_{ID}}\right)^{N-i}}{\sum_{i=0}^{N} i \binom{N}{i} \left(\Psi_{u_{ID}}\right)^i \left(1-\Psi_{u_{ID}}\right)^{N-i}} \quad (4.27)$$

where

$$\Psi_{u_{ID}} = R_{ID}\, Pmb_{ID}\, Ps_{ID} \left(\frac{1-H_{ID}}{N}\right).$$

Similarly the interactive sister packet blocking at the hotspot output is given by

$$Psbh_{ID} = \frac{\sum_{k=0}^{L} Eh_{ID_k} \sum_{i=k}^{N} (i-k) \binom{N}{i} \left(\Psi_{h_{ID}}\right)^i \left(1-\Psi_{h_{ID}}\right)^{N-i}}{\sum_{i=0}^{N} i \binom{N}{i} \left(\Psi_{h_{ID}}\right)^i \left(1-\Psi_{h_{ID}}\right)^{N-i}} \quad (4.28)$$

where

$$\Psi_{h_{ID}} = R_{ID}\, Pmb_{ID}\, Ps_{ID} \left(H_{ID} + \frac{1-H_{ID}}{N} \right).$$

Now the mean file and interactive sister packet blocking denoted by $Pmsb_{FD}$ and $Pmsb_{ID}$ can be found by averaging the blocking probability for the uniform and hotspot outputs over the probability that the sister packet will go to that output respectively. This follows the same method that was used to calculate the mean blocking for the first round. The mean sister packet blocking probabilities are

$$Pmsb_{FD} = \left(H_{FD} + \frac{1-H_{FD}}{N} \right) Psbh_{FD} + (1-H_{FD}) \left(\frac{N-1}{N} \right) Psbu_{FD} \qquad (4.29)$$

and

$$Pmsb_{ID} = \left(H_{ID} + \frac{1-H_{ID}}{N} \right) Psbh_{ID} + (1-H_{ID}) \left(\frac{N-1}{N} \right) Psbu_{ID} \qquad (4.30)$$

for file and interactive data respectively.

The service probability of the input file and interactive buffers can now be found for the HLR case by combining all first round mean blocking probabilities along with the mean sister packet blocking probabilities. The service probability for file data is

$$U_{FD} = \left[\left(1-Pmb_{FD}\right) + Pmb_{FD}\, Ps_{FD} \left(1-Pmsb_{FD}\right) \right] R_{FD}. \qquad (4.31)$$

The first term inside the brackets represents the probability that the first round file packet is successful and the second term represents the probability that the first round file packet is blocked, a sister packet is found and t sister packet is successful. Similarly, the service probability for an interactive user is given by

$$U_{ID} = \left[\left(1 - Pmb_{ID}\right) + Pmb_{ID}\ Ps_{FD}\left(1 - Pmsb_{ID}\right)\right] R_{ID}\,. \tag{4.32}$$

Now that the service probabilities have been found the distribution of the input buffers can be solved for both cases with and without HLR by substituting the respective service probability into eq. (3.37). The mean input buffer delays and buffer overflows can also be found using equations (3.38) and (3.39). Similarly, the output buffers are modeled as in Chapter 3 for the different downlink ratios. To solve for the output buffer overflow and the queuing delay the distribution of the final empty spaces at both the hotspot and the uniform outputs have to be found. The final distribution of empty spaces at the hotspot output is the number of empty spaces left after the interactive sister packets compete for the empty spaces left by the file sister packets and is given by

$$\begin{aligned} Eh_i^{'} &= \sum_{k=i}^{L} Eh_{FD_k} \binom{N}{k-i} \left(\Psi_{h_{ID}}\right)^{k-i} \left(1 - \Psi_{h_{ID}}\right)^{N-k+i} \qquad i=1,2,3,\ldots,L \\ Eh_0^{'} &= 1 - \sum_{k=1}^{L} Eh_k^{'}\,. \end{aligned} \tag{4.33}$$

The distribution of the number of packets going to the hotspot output port is then just the reciprocal of the $Eh_i^{'}$ distribution

$$Pob_i = Eh_{L-i}^{'}\,, \qquad i=0,1,2,3,\ldots,L \tag{4.34}$$

Similarly, the uniform distribution for the final empty spaces at a uniform output is given

by

$$Eu_i^{\cdot} = \sum_{k=i}^{L} Eu_{FD_k} \binom{N}{k-i} \left(\Psi_{u_{ID}}\right)^{k-i} \left(1-\Psi_{u_{ID}}\right)^{N-k+i} \qquad i=1,2,3,.....,L$$
$$Eu_0^{\cdot} = 1 - \sum_{k=1}^{L} Eu_k^{\cdot} \tag{4.35}$$

where the probability that k-i interactive sister packets will go to an output is averaged over the probability distribution, Eu_{FD_k}, that there are k empty spaces left after the file sister packets competed to give the final distribution of the i empty spaces left at the uniform output. The distribution of the number of packets going to the uniform output is then

$$Pob_i = Eu_{L-i}^{\cdot}, \quad i=0,1,2,3,...,L. \tag{4.36}$$

Equations (4.34) and (4.36) can then be used in equations (3.42), (3.45) and (3.46) to solve the hotspot output and uniform output buffer distributions. The loss and delays are then found using equations (3.40), (3.41) and equations (3.44) and (3.45) respectively.

4.4. Hotspot Traffic Results

4.4.1. Comparison of the Loss With and Without Buffering, HLR and Speedup for Different Hotspot Traffics

The first set of graphs, shown in Figures 4.15-4.18, compares the loss probability for the discard case with the input buffer overflow probability for the buffered case, for each of the services in turn, for different ratios of hotspot traffic and different switch speedups. It is assumed throughout the analysis that all the services have the same hotspot traffic ratios denoted by HTR, i.e., $H_W=H_V=H_{FD}=H_{ID}$=HTR. The input buffer size in all cases is also assumed to be 30 and the switch size is assumed to be 32. In Fig. 4.15 the video traffic relationship between the discard and buffered case shows that with buffering the switch speedup can be drastically reduced while providing the same loss as the discard case. A speedup of L=3 for the buffered case for example, provides the same loss as the discard case with L=8 for a load of 0.80 and a hotspot ratio of 0.30. For loads less than 0.5 there is very little difference between the different speedup cases with buffering. The loss probability is also seen to be relatively independent of the hotspot traffic for the buffered case at a speedup of two and three, while the discard case shows a marked difference even with L=8. For second priority voice traffic, as shown in Fig. 4.16, the difference in the buffer overflow is much more dependent on the switch speedup and the hotspot traffic ratio. However, the buffered case provides a vast improvement over the discard case where a speedup as high as eight can only maintain a loss of $1x10^{-6}$ up to a load of 0.2 with HTR=0.3, while the buffered case for the same loss can maintain a load of 0.56 with L=2 and a load of 0.61 with L=3. For third priority file data the discard

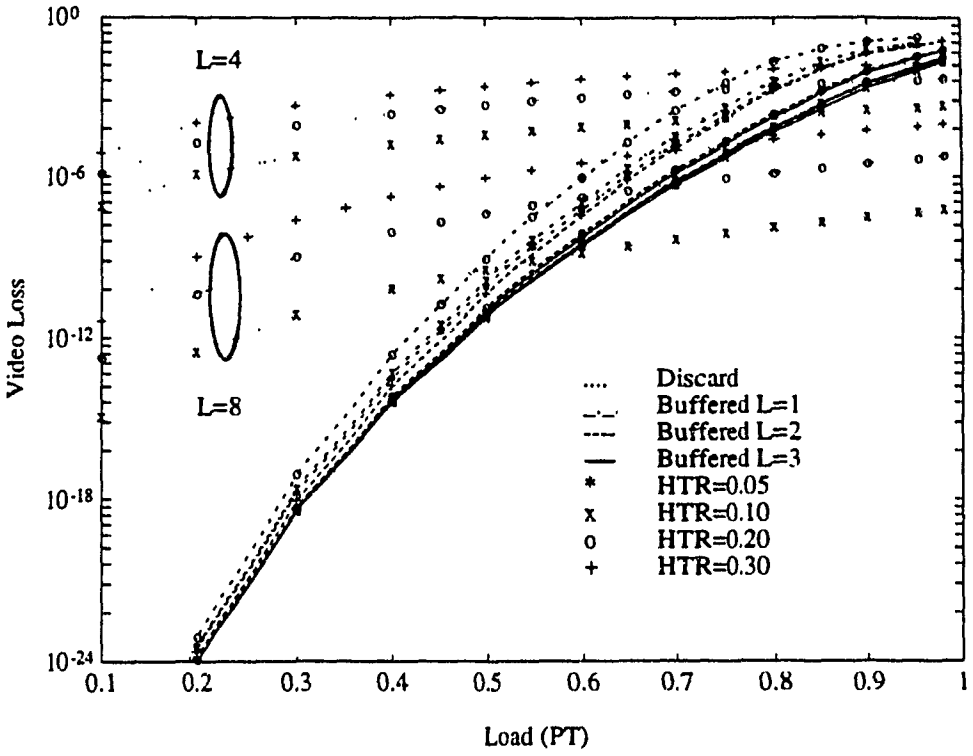


Fig. 4.15. Video loss with and without buffering.

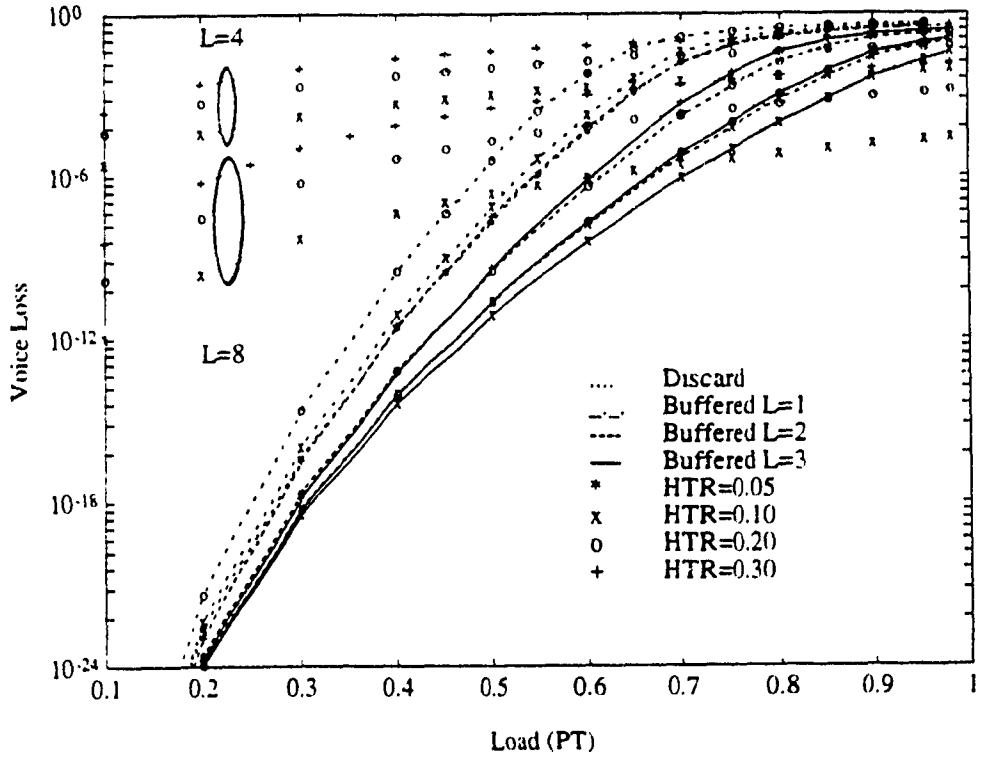


Fig. 4.16. Voice loss with and without buffering.

case, as shown in Fig. 4.17, has deteriorated even further where even with L=8 in order to maintain a loss of under $1x10^{-6}$ the load at HTR=0.3 has to be restricted to less than 0.15, while the buffered case with HLR can still maintain a load of 0.59 with L=3. It is also seen that the L=1 case quickly saturates and the loss probability can be reduced by at least four orders by increasing the speedup to two. There is also a larger difference between the L=2 and L=3 cases due to the HLR algorithm which takes advantage of the lower sister packet blocking probability. The results for interactive data are shown in Fig. 4.18. It is seen that a speedup of L=3 is still sufficient to provide a load of up to 0.56 at a loss of $1x10^{-6}$ for HTR=0.3, while the discard case reaches this loss at a load of only 0.1. The improvement in performance due to increasing the speedup is now very apparent. This larger improvement is a consequence of the fact that the lowest priority interactive data users have a higher initial blocking probability so that there is more potential for improvement. Thus, under hotspot traffic conditions input buffering with HLR and a modest speedup of two or three provides a significant increase in the performance of the system over that of the discard case which quickly saturates, especially for the lower priority users at very low loads.

To compare the improvement due to HLR on the input buffer performance for file and interactive data users the loss is plotted as a function of the load for different values of HTR with and without HLR in Figures. 4.19-4.21 for file data and in Figures 4.22-4.24 for interactive data users respectively . It is seen that for file data users the improvement due to HLR increases as the hotspot traffic increases due to the increasing blocking probability. It is also seen that HLR continues to provide a substantial improvement as

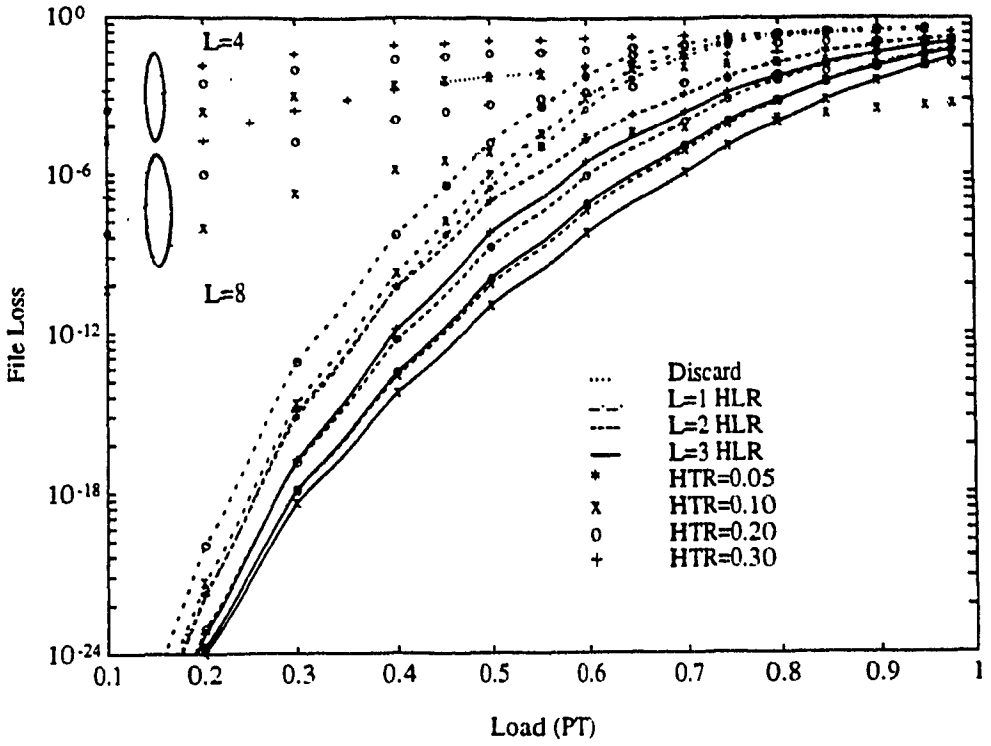


Fig. 4.17. File loss with and without buffering/HLR.

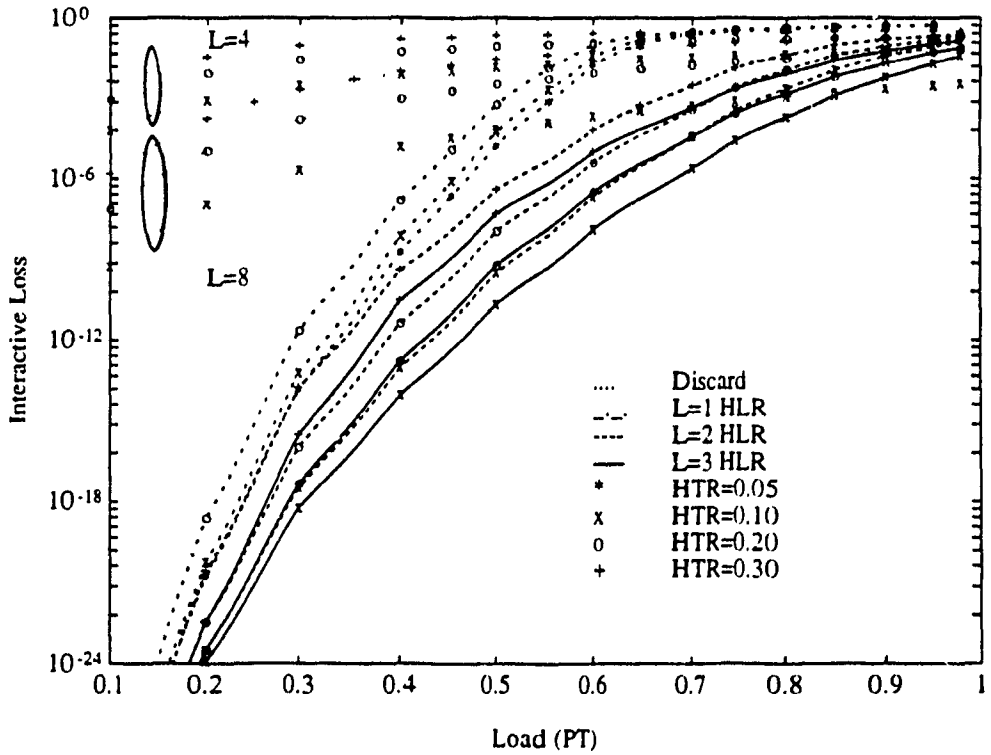


Fig. 4.18. Interactive loss with and without buffering/HLR.

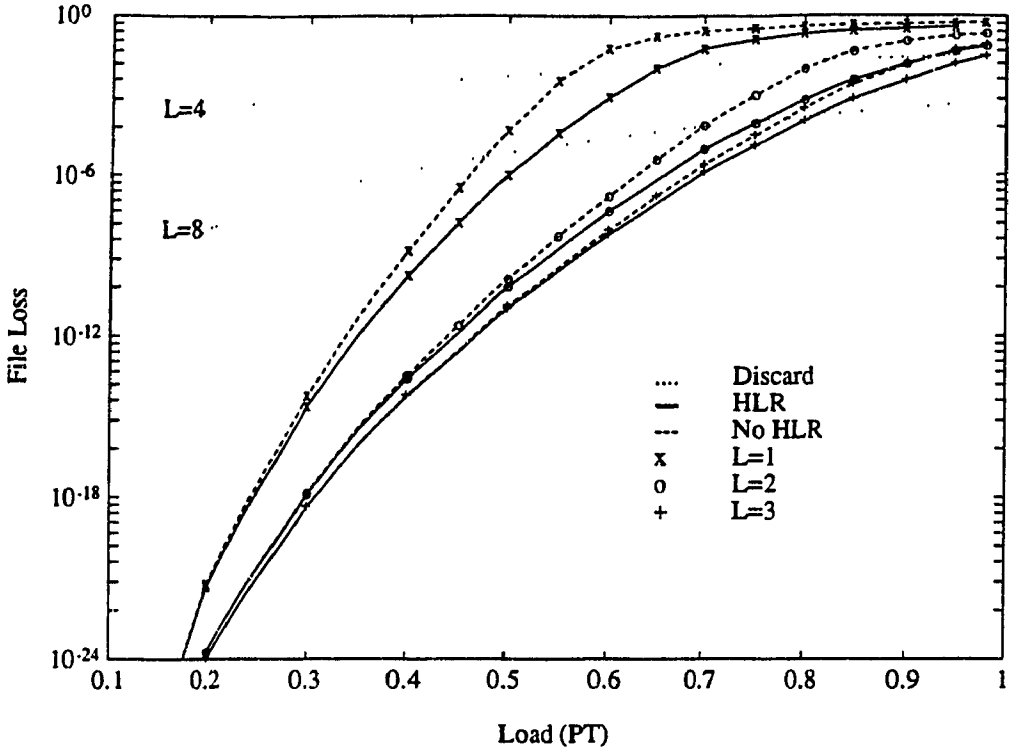


Fig. 4.19. File loss with and without HLR (HTR=0.1).

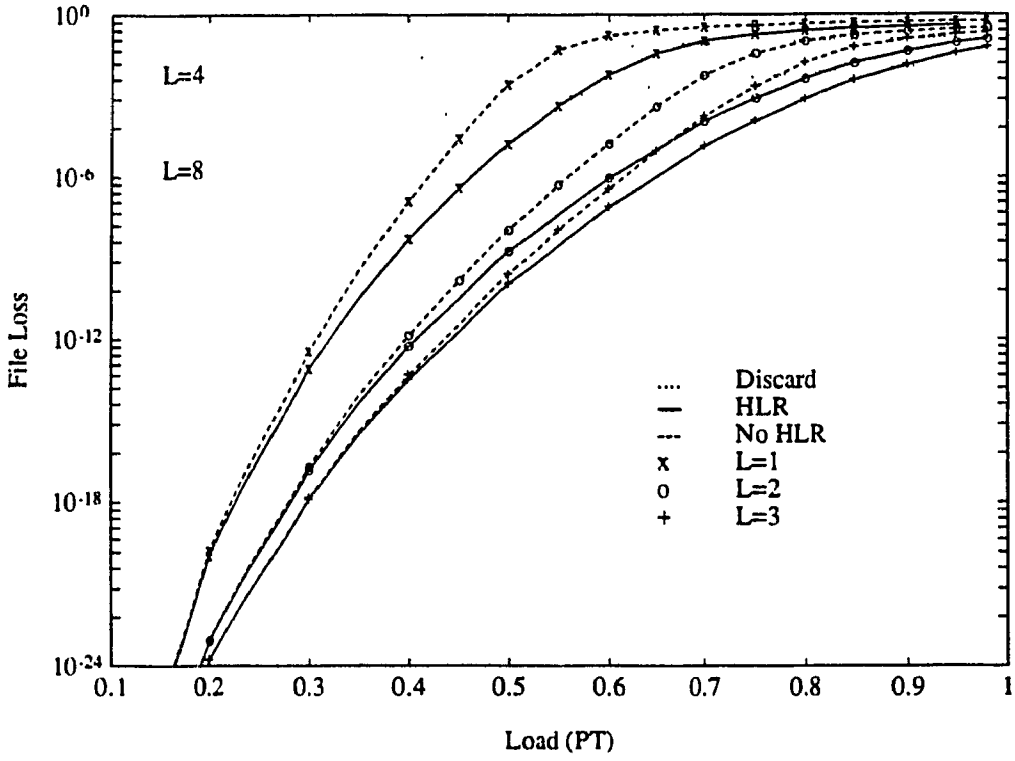


Fig. 4.20. File loss with and without HLR (HTR=0.2).

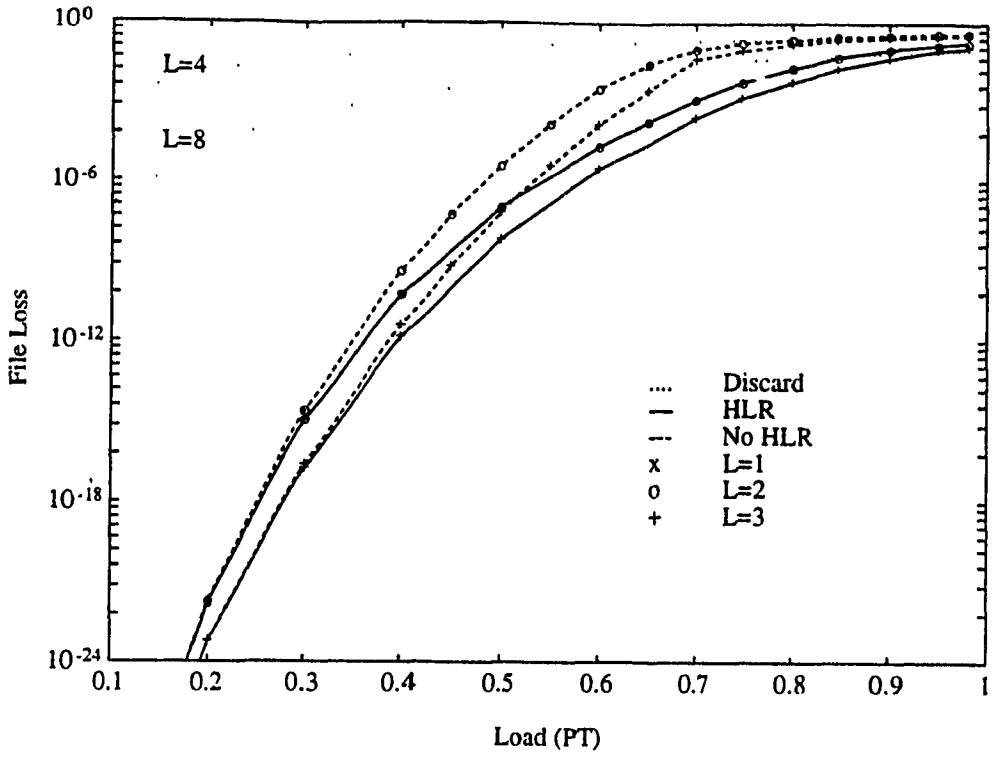


Fig. 4.21. File loss with and without HLR (HTR=0.3).

L increases. As seen for HTR=0.2 and a load of 0.5 as *L* is increased from one to three the file loss probability is reduced from $2.6\text{x}10^{-3}$, $1.07\text{x}10^{-8}$, $2.55\text{x}10^{-10}$ without HLR to $4.25\text{x}10^{-4}$, $1.8\text{x}10^{-9}$, $1.12\text{x}10^{-10}$ with HLR respectively. In addition it is seen that the improvement due to HLR at HTR=0.3 is equivalent to increasing the speedup by at least one at moderate loads and increases rapidly as the load is increased above 0.6. For interactive data users which have a higher blocking probability to start with there is already a greater improvement between the cases with and without HLR even at HTR=0.1. It is also seen that the loss is even more substantially reduced from $1.02\text{x}10^{-1}$, $1.77\text{x}10^{-7}$, $2.84\text{x}10^{-9}$ without HLR to $5.7\text{x}10^{-4}$, $1.08\text{x}10^{-8}$, $6\text{x}10^{-10}$ with HLR as the

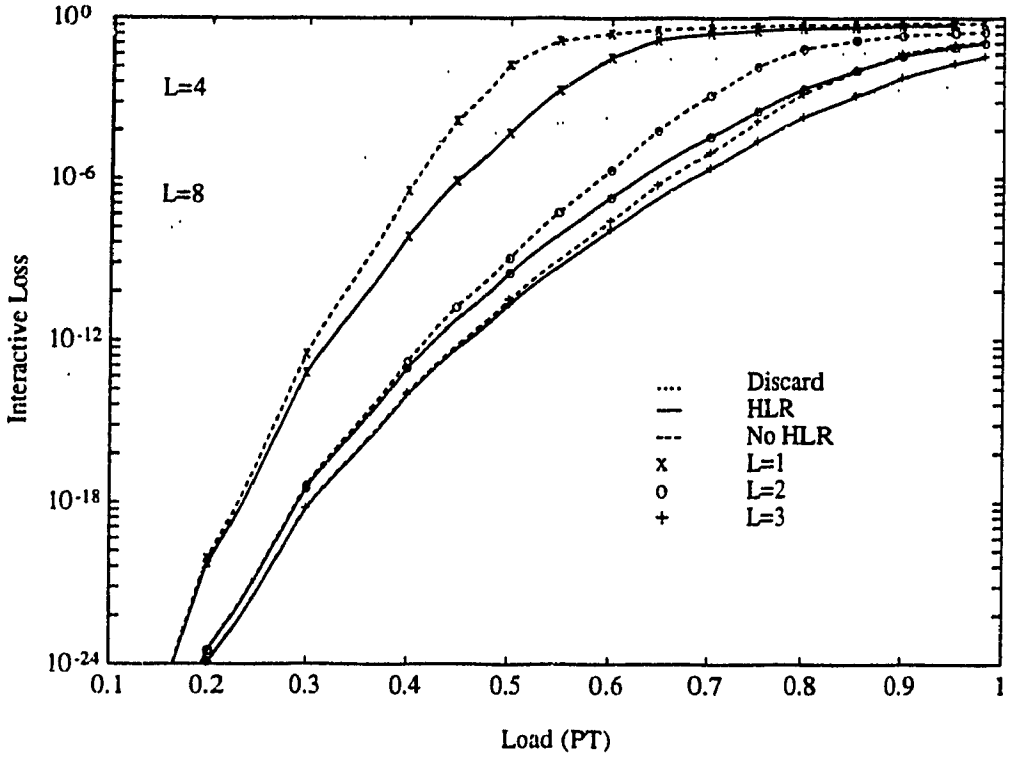


Fig. 4.22. Interactive loss with and without HLR (HTR=0.1).

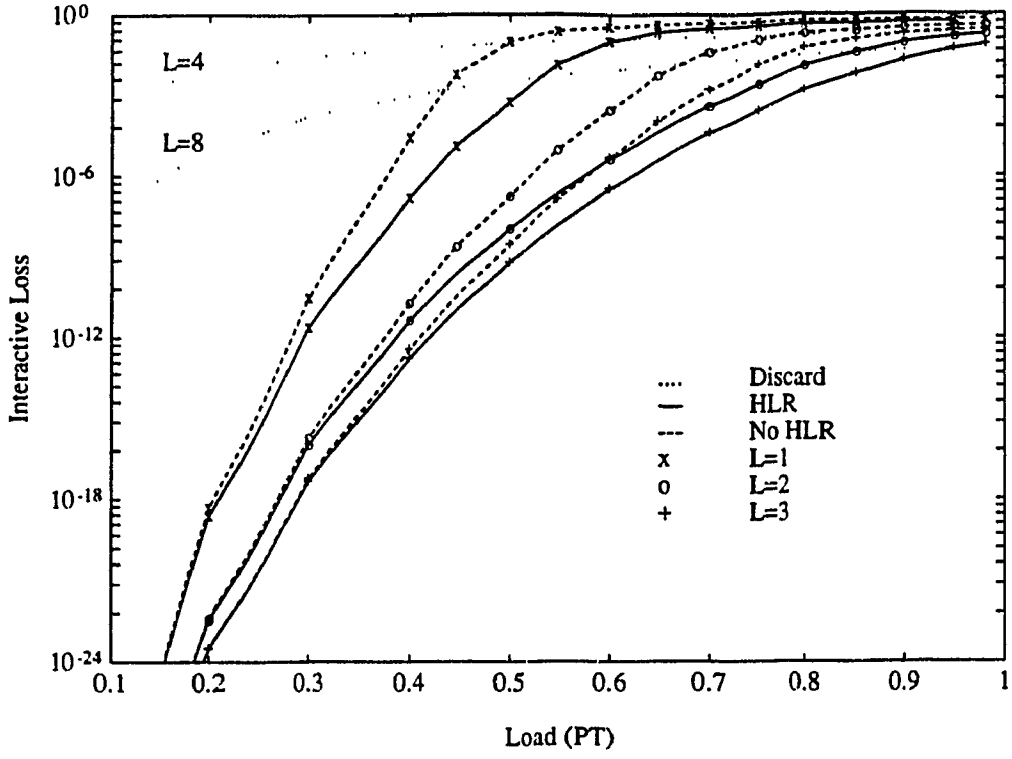


Fig. 4.23. Interactive loss with and without HLR (HTR=0.2).

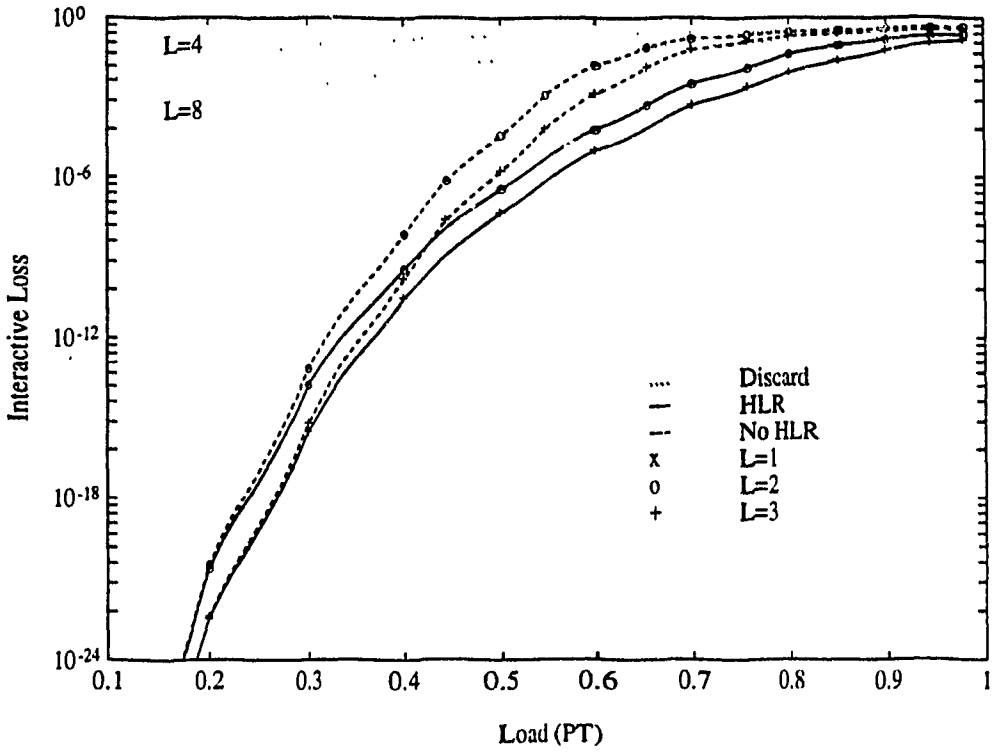


Fig. 4.24. Interactive loss with and without HLR (HTR=0.3).

speedup is increased from one to three respectively at a load of 0.5 with HTR=0.2. It is seen that due to the higher blocking probability that interactive data users gain more from the use of HLR. It can thus be concluded that for both file and interactive data users the buffered case drastically outperforms the discard case and in order to limit the loss probability buffering is necessary under hotspot traffic conditions especially at higher loads. Also in order to limit the required speedup at high loads HLR is indispensable especially for interactive data users which suffer the highest blocking.

4.4.2. Input Buffer Delays With and Without HLR and Speedup

To examine the relationship of the input buffer further for different values of hotspot traffic the input buffer delays are plotted versus the load in Figures 4.25-4.32. Fig. 4.25 shows the results for video traffic. As shown, even with a limited speedup of one the maximum load is still 0.71 at HTR=0.2. By increasing the speedup to two the load is substantially increased to 0.82. Due to the higher traffic there is also now a noticeable difference between L=2 and L=3. The maximum load between L=2 and L=3 at a delay of 20 as shown increases from 0.82 to 0.85 at HTR=0.2 and from 0.78 to 0.83 at HTR=0.30. Fig. 4.26 shows the results for voice users. For L=1 the system quickly saturates at a higher HTR of 0.2 driving the maximum load down to 0.55 for a delay of 20 from about 0.64 for HTR=0.05. Increasing the speedup from two to three increases the maximum load significantly from 0.72 to 0.79 for HTR=0.20 and from 0.64 to 0.71 for HTR=0.30. It is seen in this case of second priority traffic that there is a much greater spread as the hotspot traffic increases as compared to video. In particular there is a more significant difference between the L=2 and L=3 cases. Turning now to the low priority users, the results for file data are shown in Figures 4.27-4.29 for both cases with and without HLR. Here it is seen that HLR significantly improves the throughput for file data users especially under heavier hotspot traffic conditions. For L=1 the improvement at HTR=0.2 increases from a maximum load of 0.48 without HLR to 0.55 with HLR. The increase is more pronounced with L=2, where it is seen that with HTR=0.2 the maximum load can be increased from 0.65 without HLR to 0.75 with HLR. At L=3 the maximum load with HTR=0.2 for the case without HLR is about 0.71, while that with HLR is 0.8.

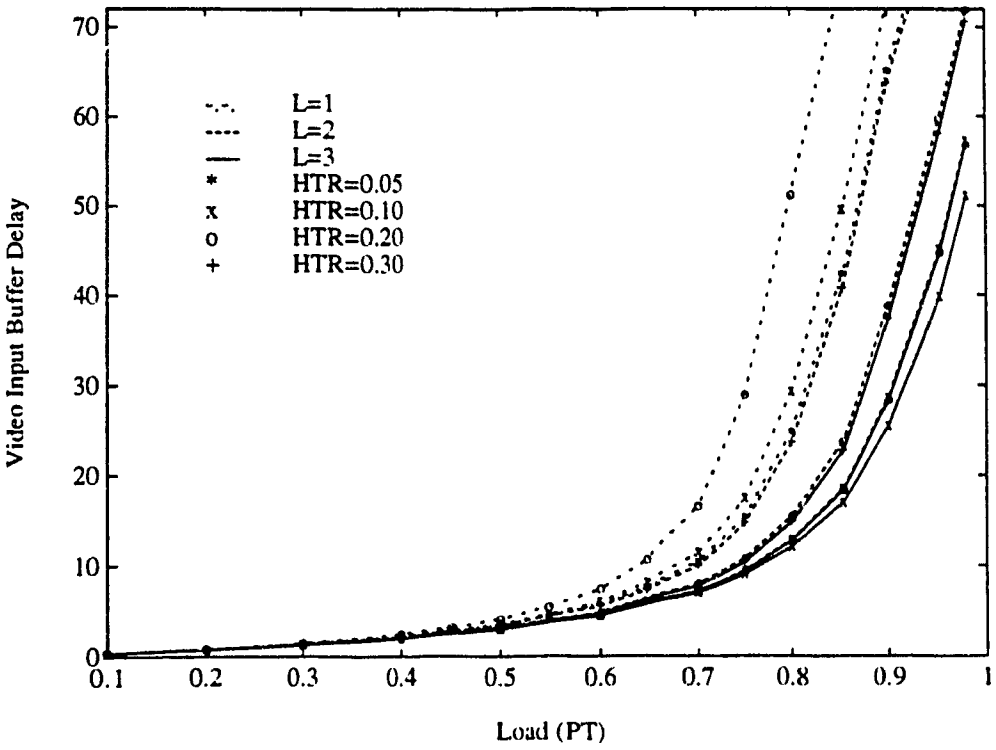


Fig. 4.25. Video input buffer delay for L=1,2,3 and HTR=0.05-0.30.

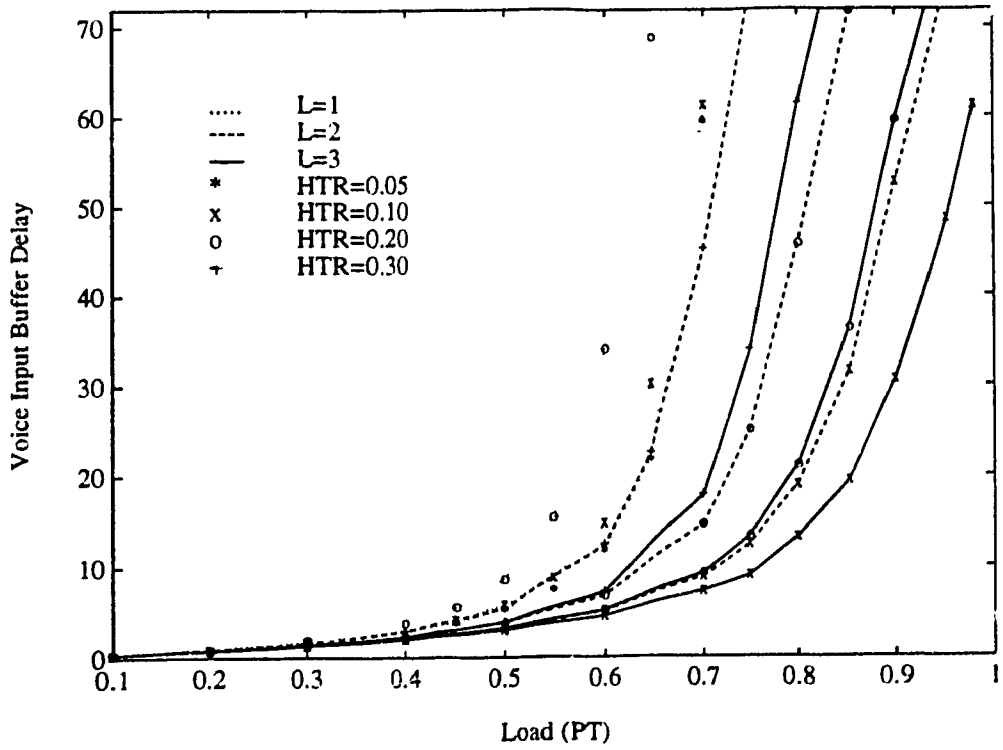


Fig. 4.26. Voice input buffer delay for L=1,2,3 and HTR=0.05-0.30.

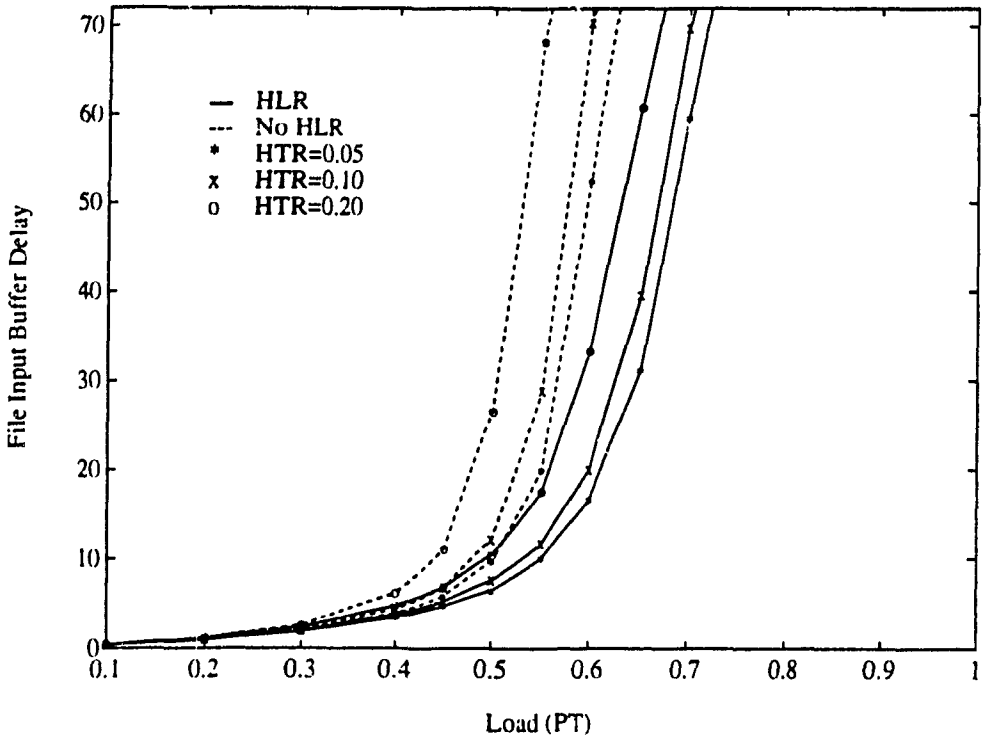


Fig. 4.27. File input buffer delay (L=1).

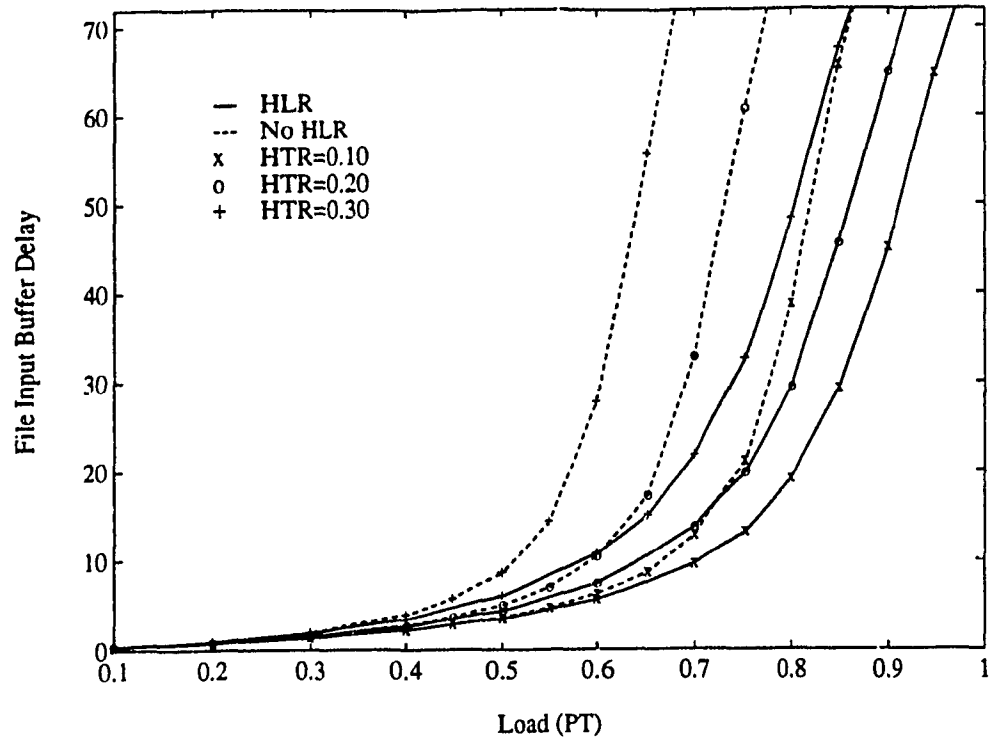


Fig. 4.28. File input buffer delay (L=2).

Thus, it is seen that with a low speedup the maximum throughput is lower due to the higher blocking. Also, the improvement due to HLR is less pronounced due to collisions of the sister packets. For higher speedups the HLR takes advantage of the probability of empty spaces available at other outputs and thus increases the throughput. In fact, with HLR the throughput of file data increases beyond that of voice when there is a high probability that a sister packet is available and is successful, as seen by comparing Fig. 4.26 and Fig. 4.29. This breach in priority is due to the fact that preference is given to file and interactive users and would not occur if HLR were used for all the services. The results for interactive data users are shown in Figures 4.30-4.32. Here it is again seen that interactive data improves the most with use of HLR. For a delay of 20 with L=1 and HTR=0.2 it is seen that the load increases from 0.41 to 0.5 with the use of HLR. At L=2 with HTR=0.2 it is seen that the improvement is even more prominent, increasing from 0.61 to 0.715, while at L=3 the improvement is from 0.68 without HLR to 0.78 with HLR. For both file and interactive data it is seen that as the speedup increases the effect of the HLR algorithm decreases. However it decreases much slower at higher values of HTR such as 0.2 and 0.3 and it is seen that even with L=3 HLR is necessary to achieve moderate to high loads. This is especially true for interactive data users.

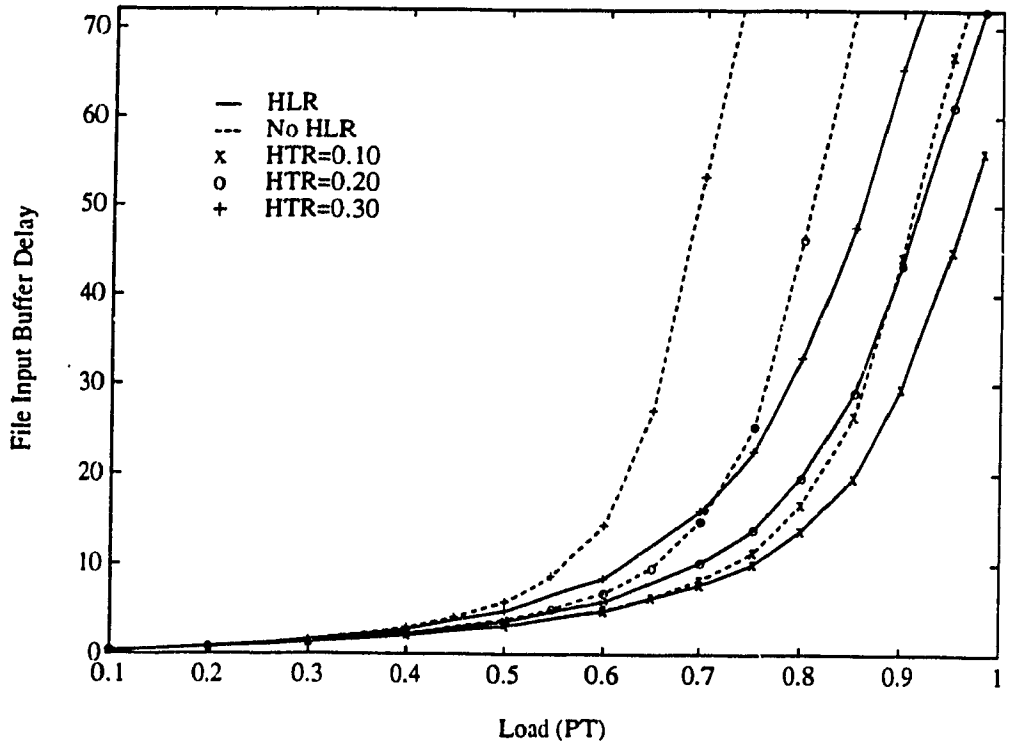


Fig. 4.29. File input buffer delay (L=3).

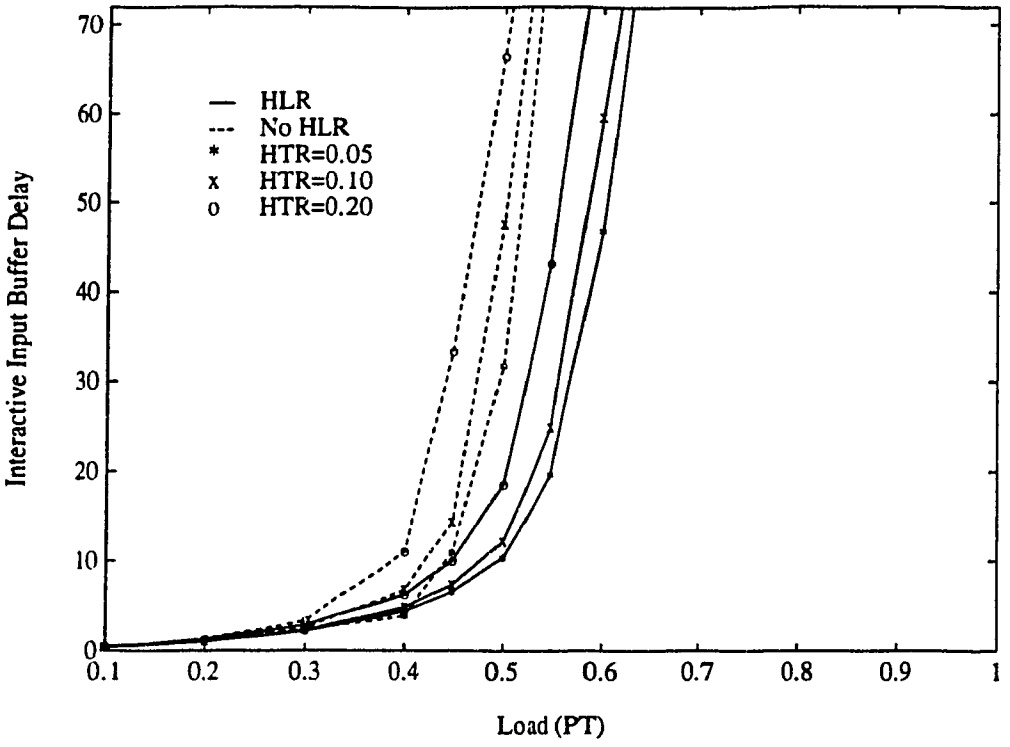


Fig. 4.30. Interactive input buffer delay (L=1).

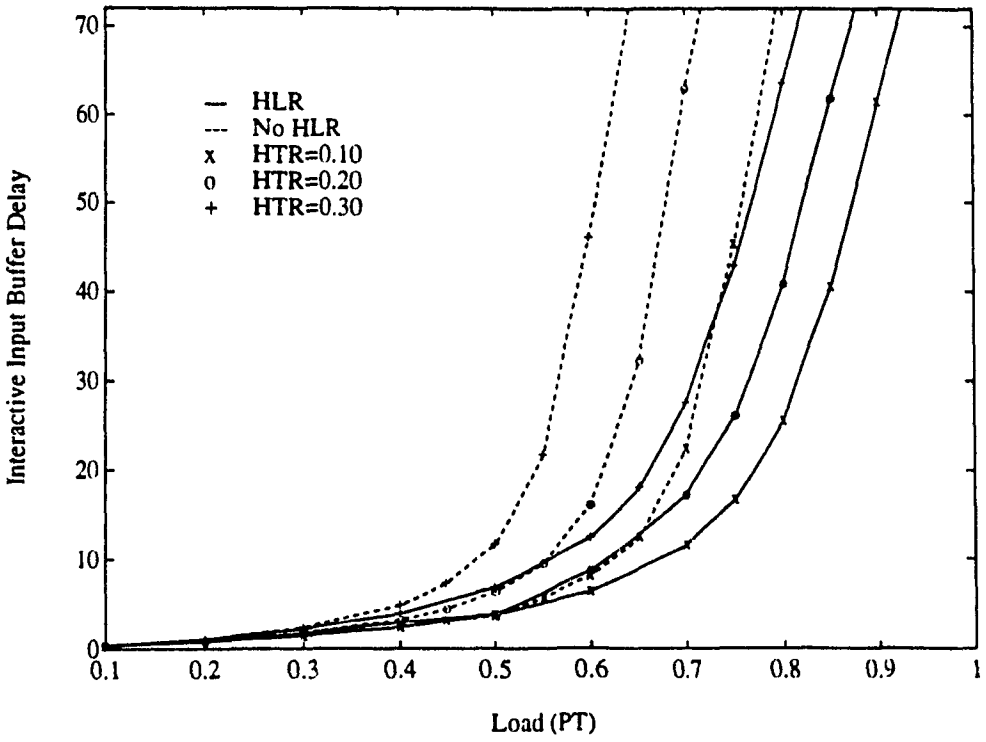


Fig. 4.31. Interactive input buffer delay (L=2).

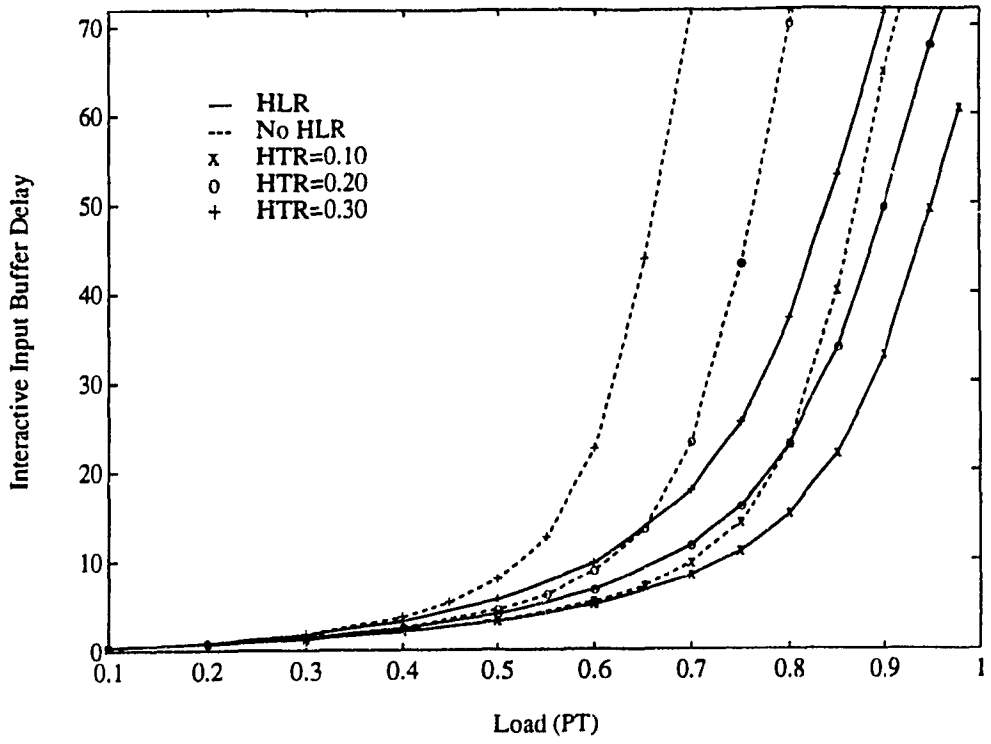


Fig. 4.32. Interactive input buffer delay (L=3).

4.4.3. Loss Performance for Different Input Buffer Sizes With and Without HLR and Speedup

In the following, the effect of the buffer length on the loss at the input buffer for each of the services is examined in order to determine the necessary buffer size required to provide an acceptable loss for hotspot traffic. The results for video, which are shown in Fig. 4.33 for a load of 0.7, indicate that a loss of $1x10^{-6}$ can be maintained with L=3 and HTR=0.3 with a minimum buffer size of 31. Increasing the buffer to 40 reduces the loss to $4x10^{-8}$ with L=3 or equivalently provides a loss of $7x10^{-7}$ for L=2. It is also seen for video that increasing L above three is not necessary at this load since there is very little difference between the three hotspot traffic ratios. This is more clearly seen in Fig. 4.34 where the load has been reduced to 0.5. Now a buffer size of 17 is sufficient with HTR=0.3 and L=3 while a buffer size of 20 is sufficient with L=1 and HTR=0.2 to limit the loss to $1x10^{-6}$ respectively. Fig. 4.35 shows the results for different voice input buffer sizes for a load of 0.7. At this load with L=3, only an HTR of 0.1 can be maintained with a buffer of size 30 at a loss of $1x10^{-6}$. The loss increases rapidly due to buffer saturation as HTR is increased and it is seen that for HTR=0.3 even with a buffer of size 40 a loss of $8x10^{-5}$ can only be maintained. Decreasing L to two also has a significant effect with the loss being $1x10^{-4}$ with HLR for HTR=0.2 at a buffer size of 30. Fig. 4.36 shows the results when the load is reduced to 0.5. Now for HTR=0.3 and L=3 a buffer of size 19 is sufficient to maintain a loss of $1x10^{-6}$ while for L=2 a buffer size of 24 is needed to maintain the same loss. Continuing, the results for file data are shown in Fig. 4.37 and Fig. 4.38 for a load of 0.7 and 0.5 respectively, with and without

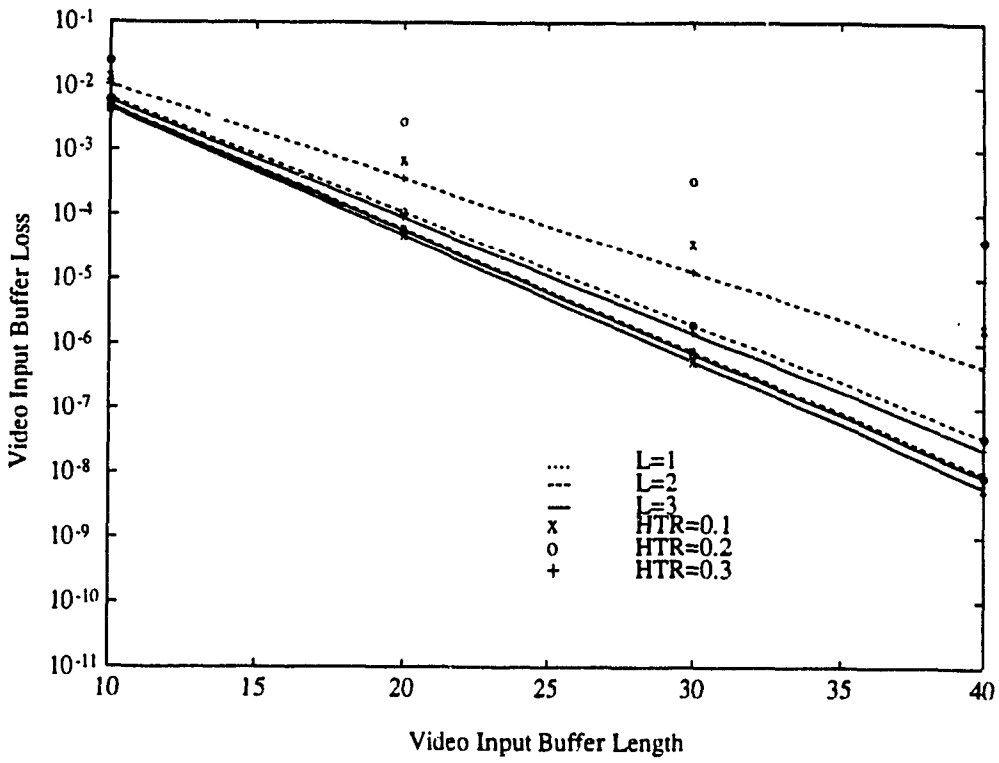


Fig. 4.33. Video input buffer loss versus buffer size (Load=0.7).

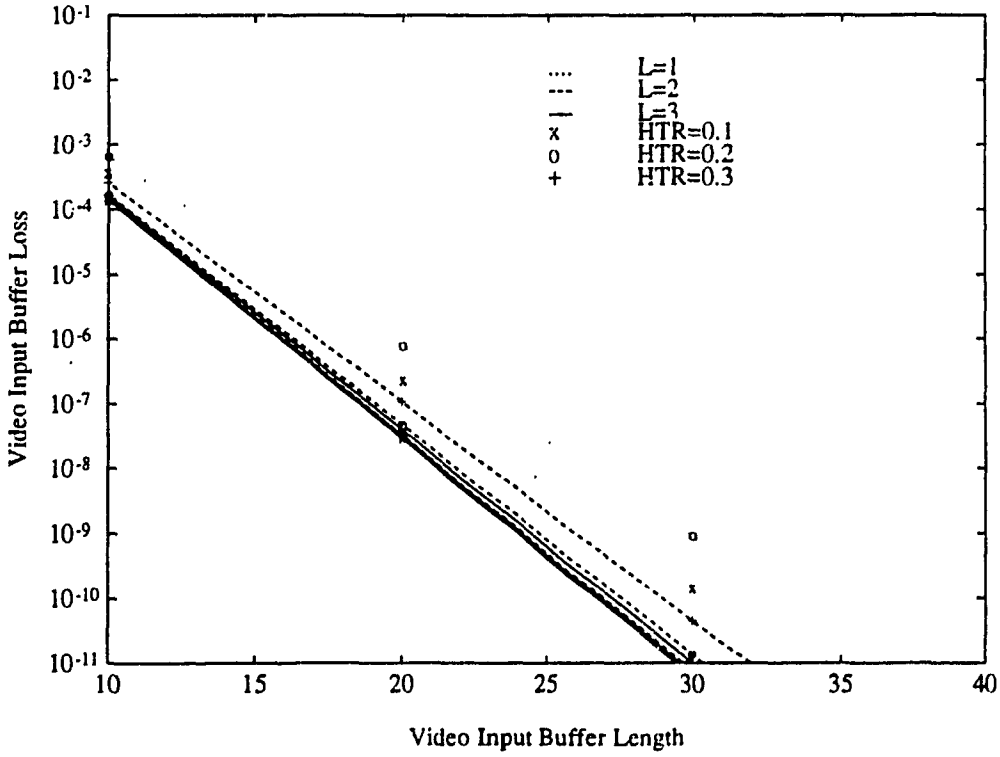


Fig. 4.34. Video input buffer loss versus buffer size (Load=0.5).

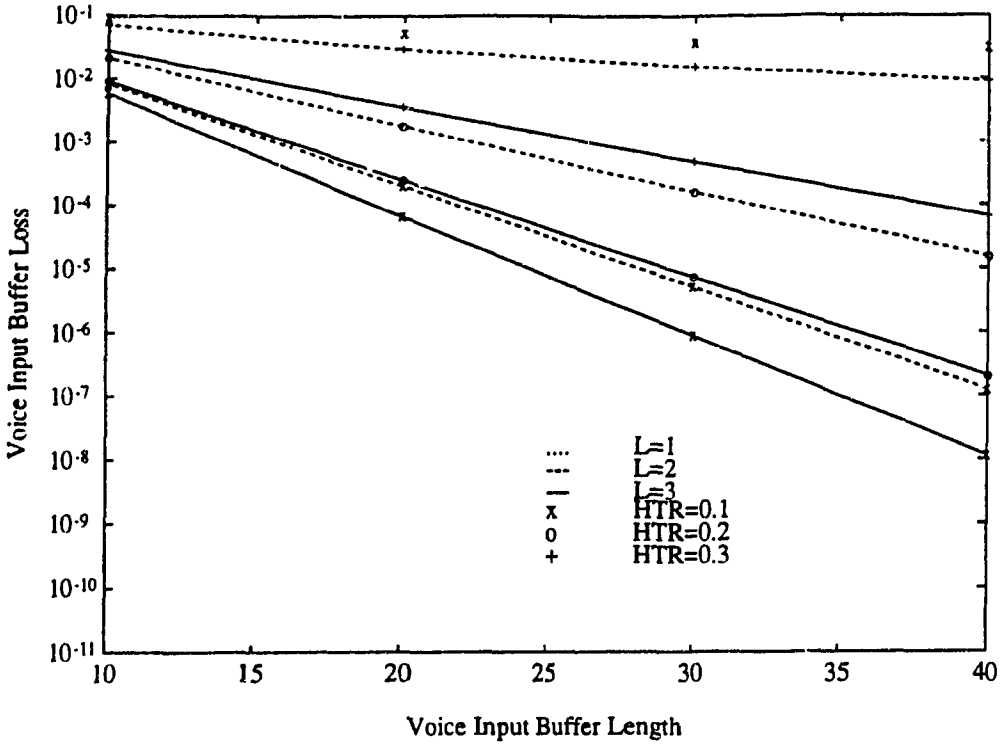


Fig. 4.35. Voice input buffer loss versus buffer size (Load=0.7).

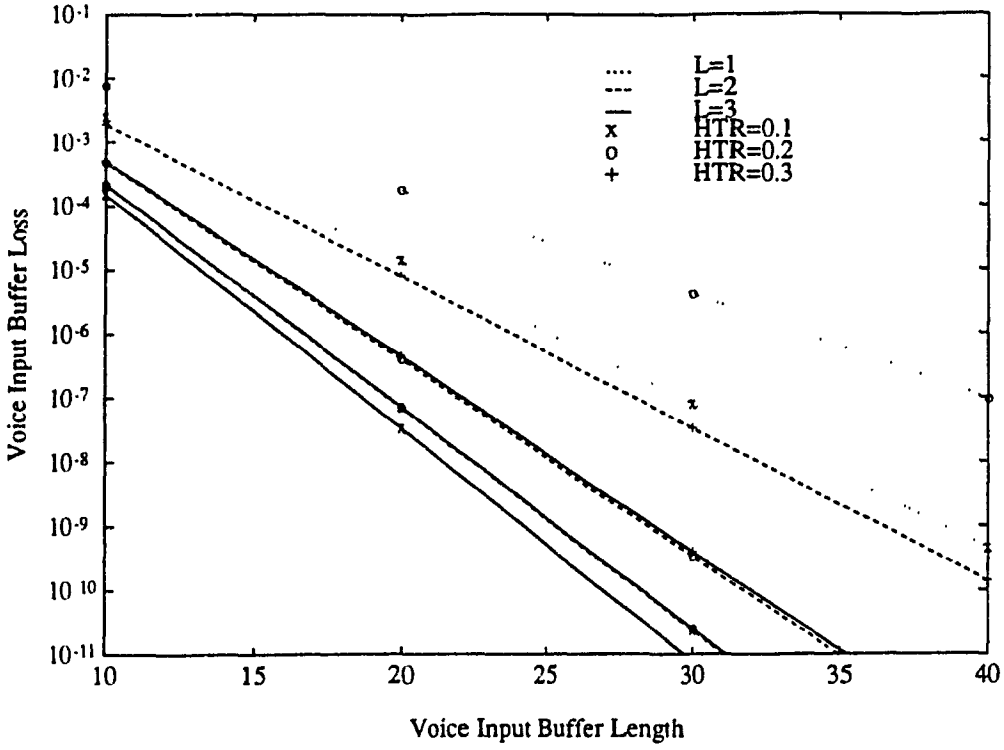


Fig. 4.36. Voice input buffer loss versus buffer size (Load=0.5).

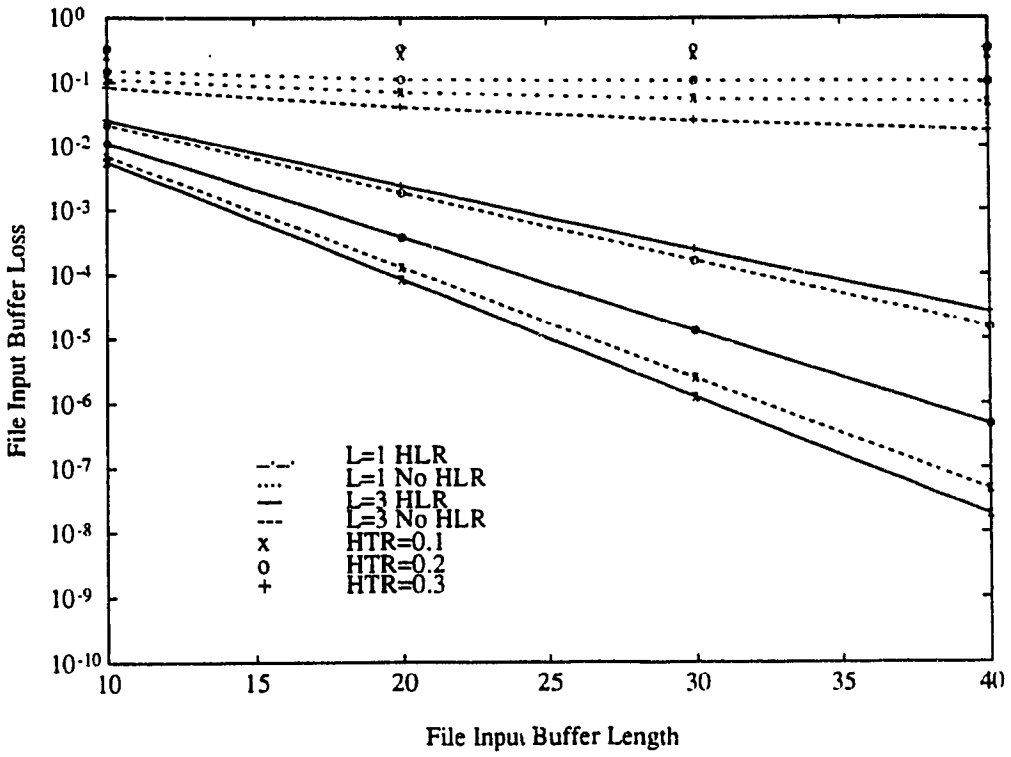


Fig. 4.37. File input buffer loss versus buffer size (Load=0.7).

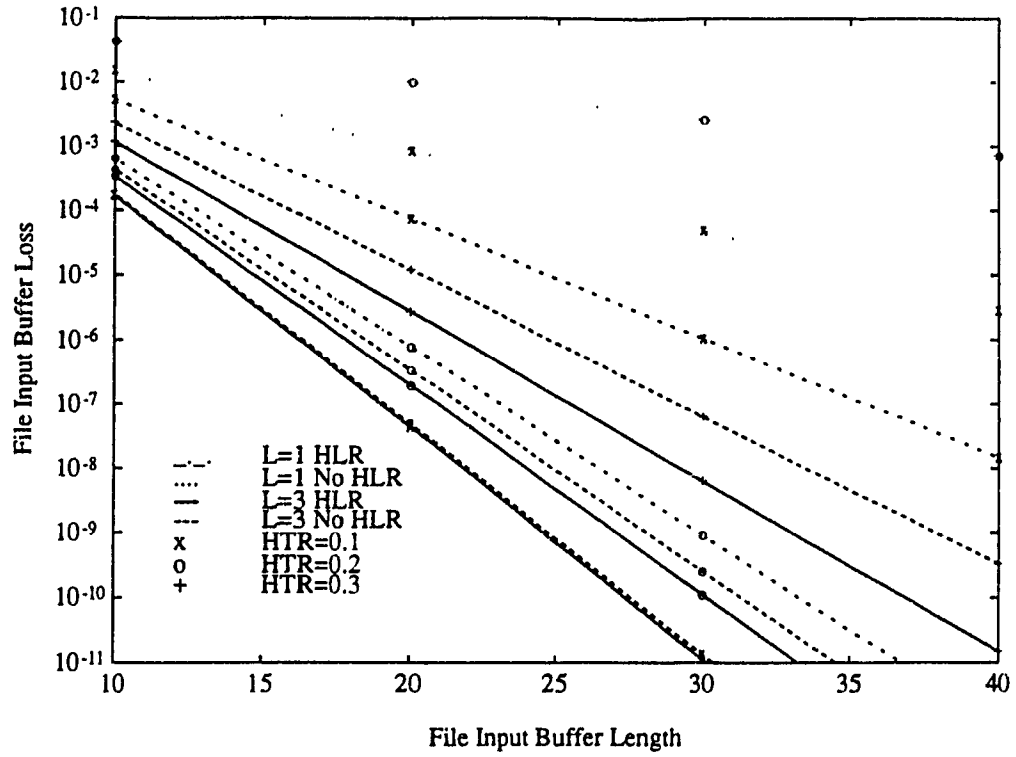


Fig. 4.38. File input buffer loss versus buffer size (Load=0.5).

HLR. As can be seen a loss of $1x10^{-6}$ can still be maintained with HLR at a load of 0.7 with HTR=0.1 and a buffer size of 29. The losses without HLR as seen before are close for L=3 for low hotspot traffics but increase rapidly as HTR is increased. Reducing the traffic to 0.5 it is seen that a buffer of size 22 is sufficient with HLR to keep the loss below $1x10^{-6}$ for L=3 and HTR=0.3, while the buffer has to be increased to 25 to meet this loss under the same conditions for the case without HLR. It is also seen that the slope in each case for HLR is greater than that for the case without HLR, so that the benefits of the HLR algorithm increase with an increase in buffer size, which is what is expected since the probability of finding a sister packet is proportional to the buffering capacity. For the interactive data users the results are shown in Fig. 4.39 and Fig. 4.40, for loads of 0.7 and 0.5 respectively. As shown, for a load of 0.7 the L=3 case with HLR can still maintain a loss of less than $1x10^{-6}$ at HTR=0.1 with a buffer length of 32, while the case without HLR requires a buffer length of 37. At this high load, as can be seen, there is a significant difference between the HLR case and the case without HLR. For a lower traffic load of 0.5, a buffer of size 25 with HLR and 32 without HLR are required to maintain a loss of $1x10^{-6}$ at HTR=0.3. Thus, for both file and interactive data users there is a greater difference between the case with HLR and the case without HLR under hotspot traffic due to the increased blocking of the lowest priority users. However, even for interactive data the combination of a relatively small buffer and the HLR algorithm significantly decreases the loss over that of the case without buffering for high hotspot traffic, while maintaining a minimum load of between 0.5 and 0.6.

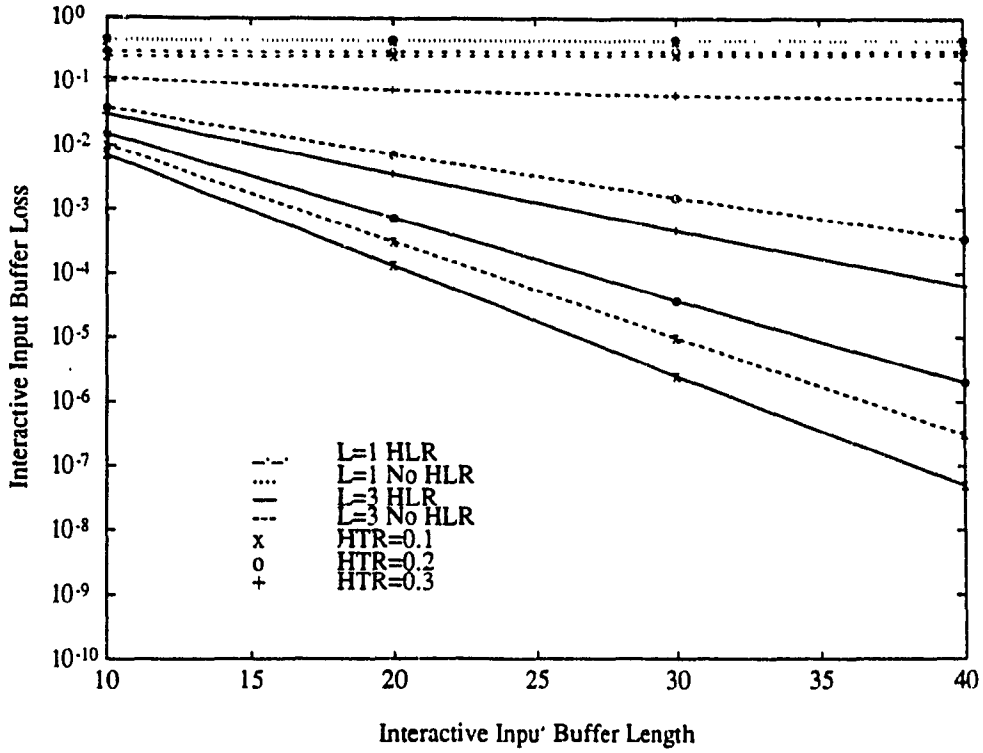


Fig. 4.39. Interactive input buffer loss versus buffer size (Load=0.7).

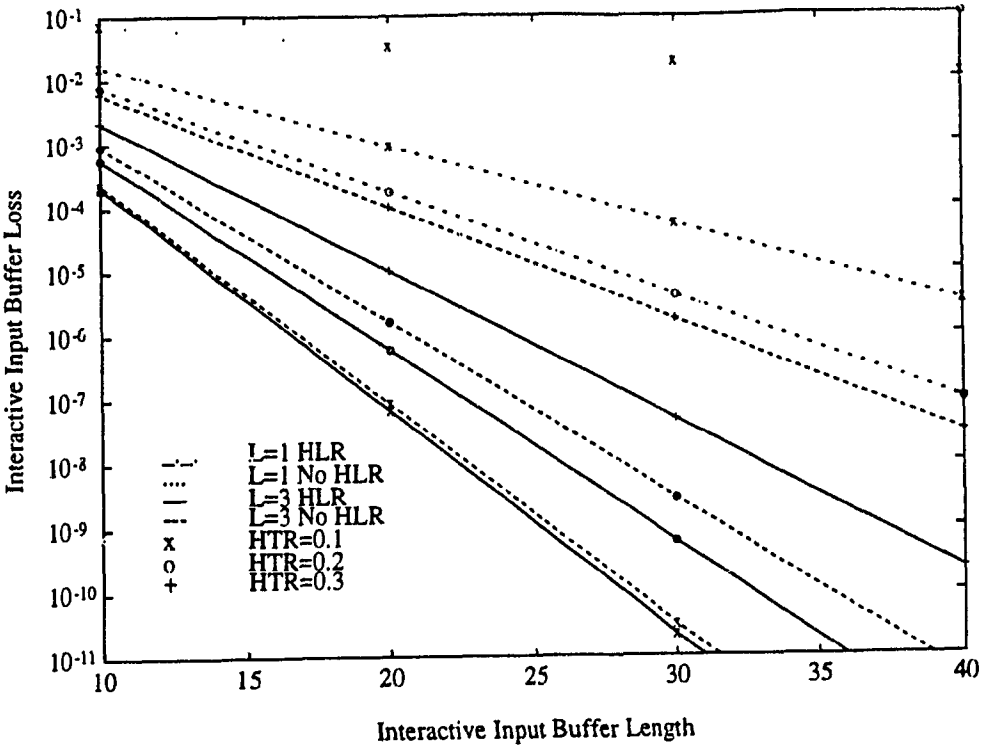


Fig. 4.40. Interactive input buffer loss versus buffer size (Load=0.5).

4.4.4. Output System Design: Buffer Size and Downlink Rate

At the hotspot output buffer careful design of the buffer size and the downlink ratio have to be engineered in order to ensure that the overflow loss is much lower than that at the input. To this end, the output buffer loss and delays for different switch speedups and different loads are examined. The output buffer loss for L=2 is shown in Fig. 4.41 where it is seen that the loss is highly dependent on the load and the value of HTR. Even for very small loads with HTR=0.1 the buffer quickly saturates so that for practical throughputs the *DLR* of the buffer has to be increased. The corresponding output buffer delay is shown in Fig. 4.42 for different hotspot traffics and it is seen again that the buffer saturates at low loads and is heavily dependent on HTR. The results for the loss and delay for the case with L=3 for both downlink ratios are shown in Figures 4.43-4.46. As seen the maximum achievable load has increased, however it is still low as compared to that which can be carried at the input. Due to this low throughput the buffer size has to be at least 58 in order to provide an overflow probability of $1x10^{-6}$, as shown in Fig. 4.43 for a load of 0.5 with HTR=0.1. Increasing HTR to 0.2 it is seen in Fig. 4.44 that the situation becomes even worse with a load of 0.27 requiring a buffer of size 37 to maintain a loss of $1x10^{-6}$ with *DLR*=2. At a higher hotspot traffic of HTR=0.3, it is seen in Fig. 4.45 that even with a buffer of size 60 at a load of 0.2 the minimum loss is only $3x10^{-5}$. The delay results are plotted in Fig. 4.46 and it is seen that increasing L to three increases the load at the output, so with *DLR*=1 at HTR=0.1 the maximum load is reduced to 0.21 from 0.25 under the same conditions with L=2 (Fig. 4.42). For a downlink ratio of two the maximum load can now be extended to 0.5 with HTR=0.1,

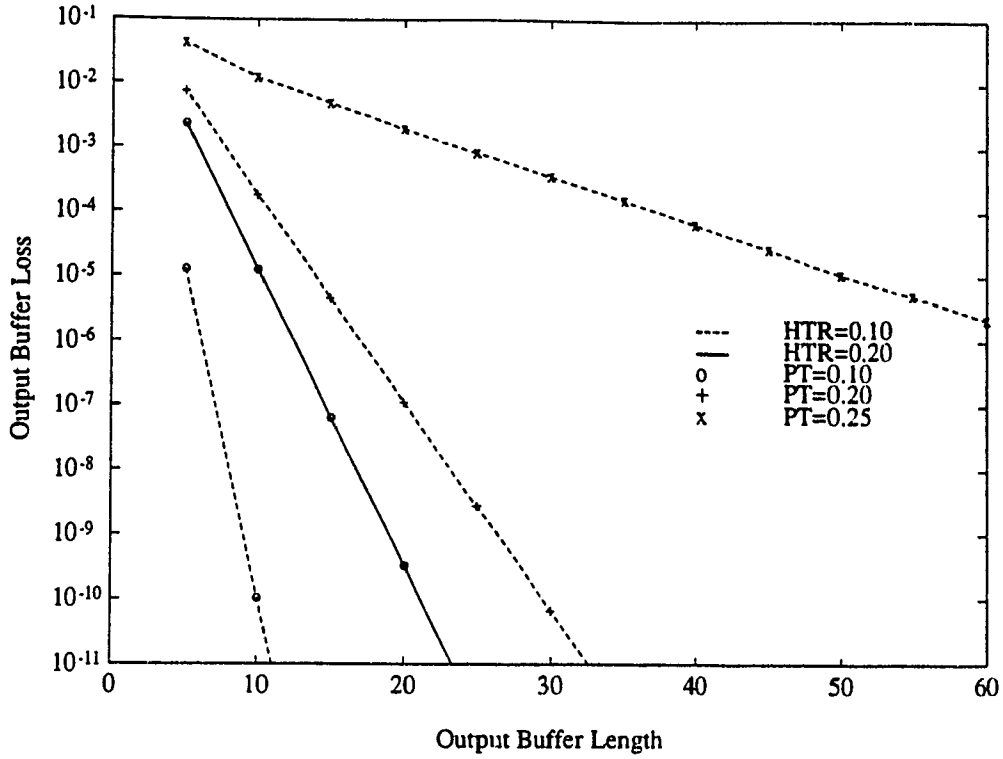


Fig. 4.41. Output buffer loss versus buffer length (L=2).

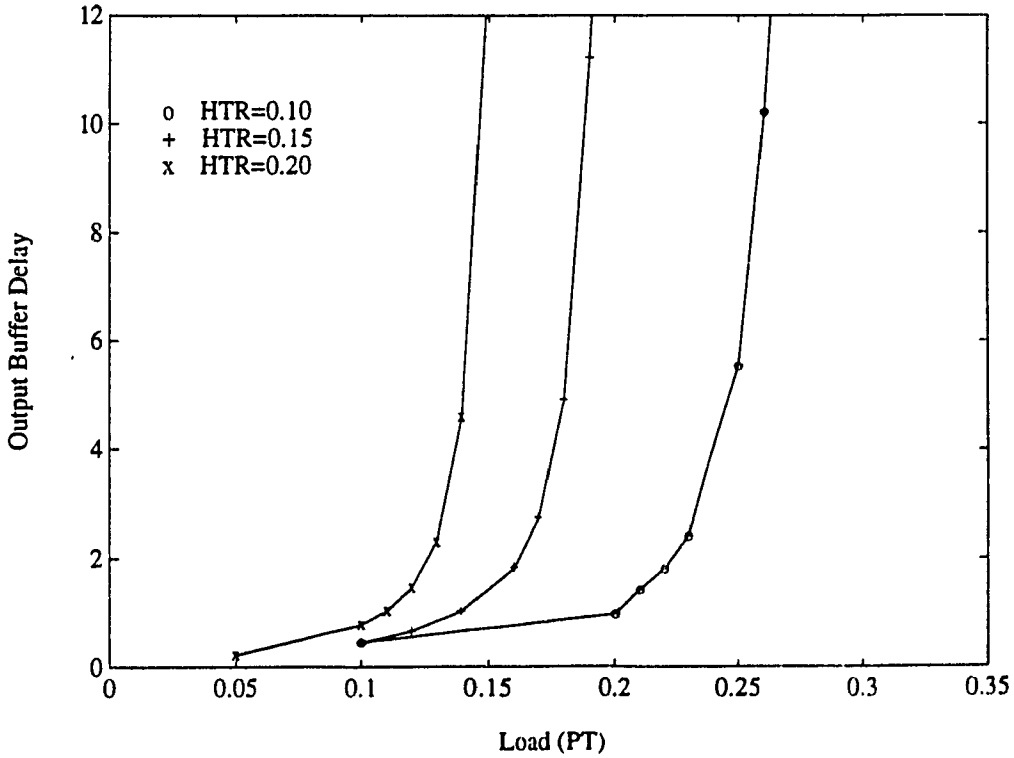


Fig. 4.42. Output buffer delay (L=2).

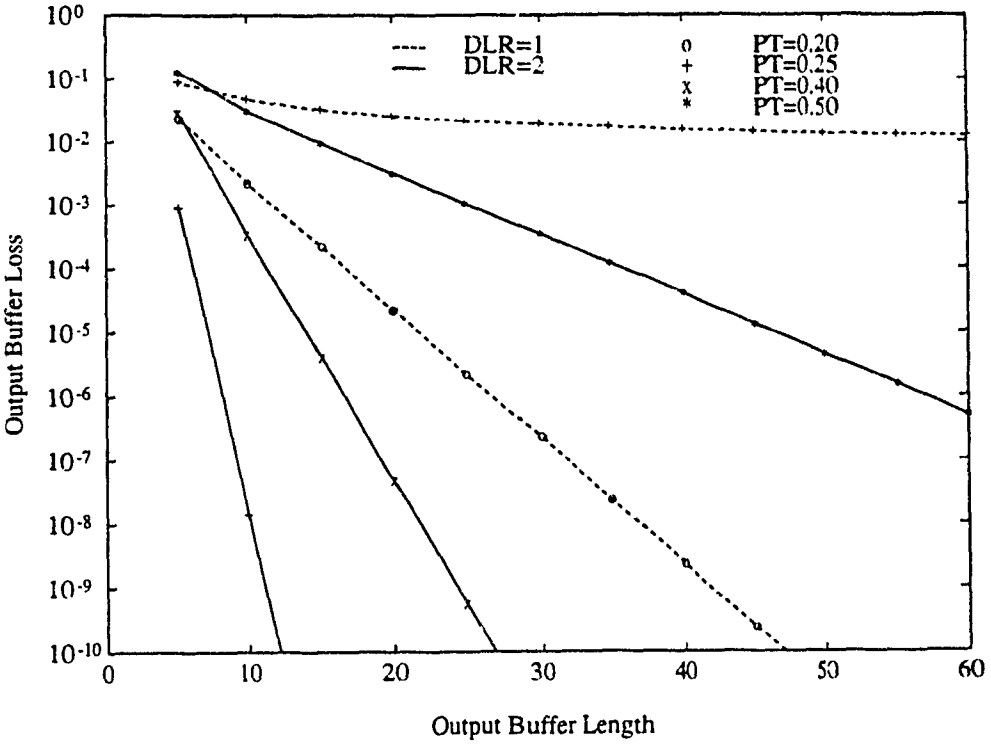


Fig. 4.43. Output buffer loss versus buffer length (L=3, HTR=0.1).

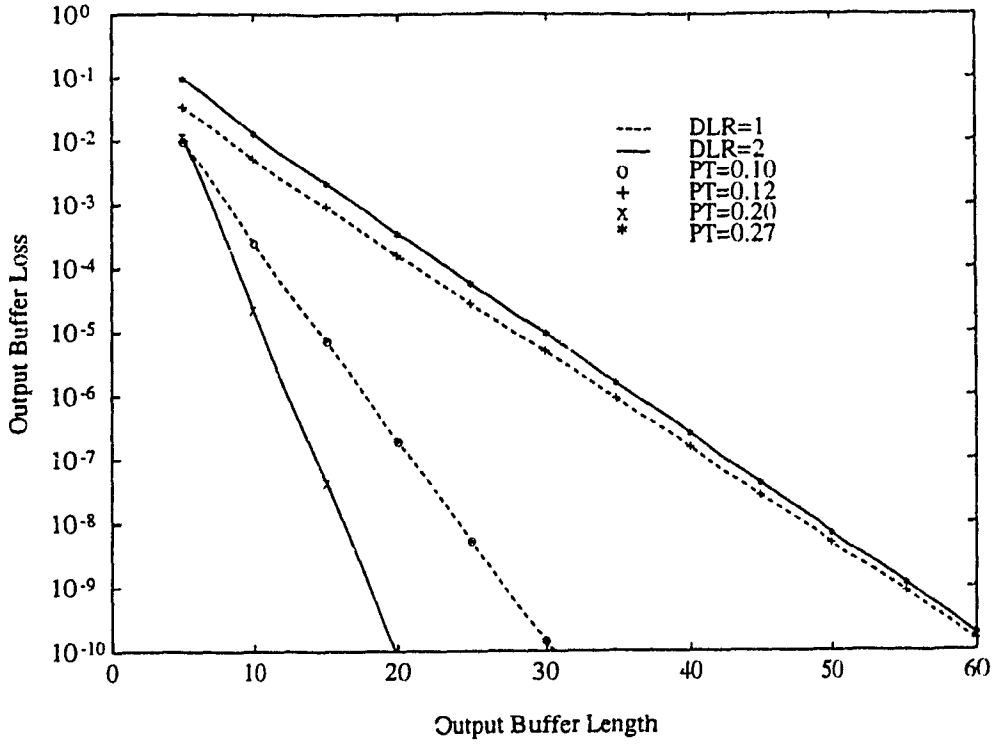


Fig. 4.44. Output buffer Loss versus buffer length (L=3, HTR=0.2).

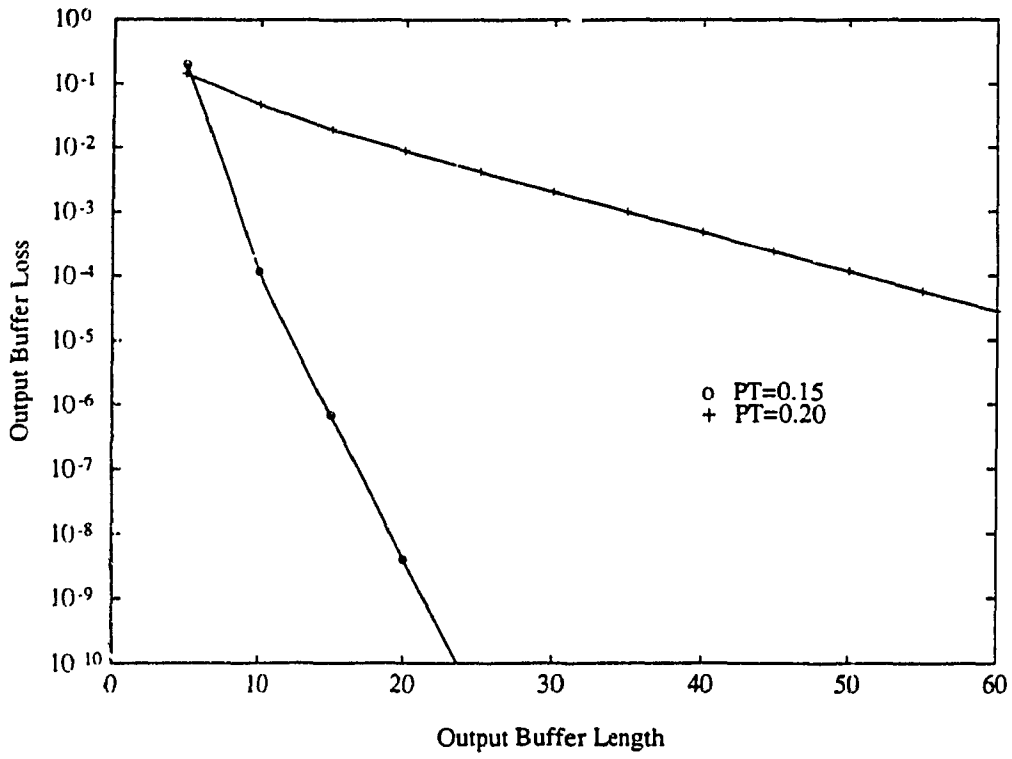


Fig. 4.45. Output buffer loss versus buffer length for *DLR*=2 (*L*=3, HTR=0.3).

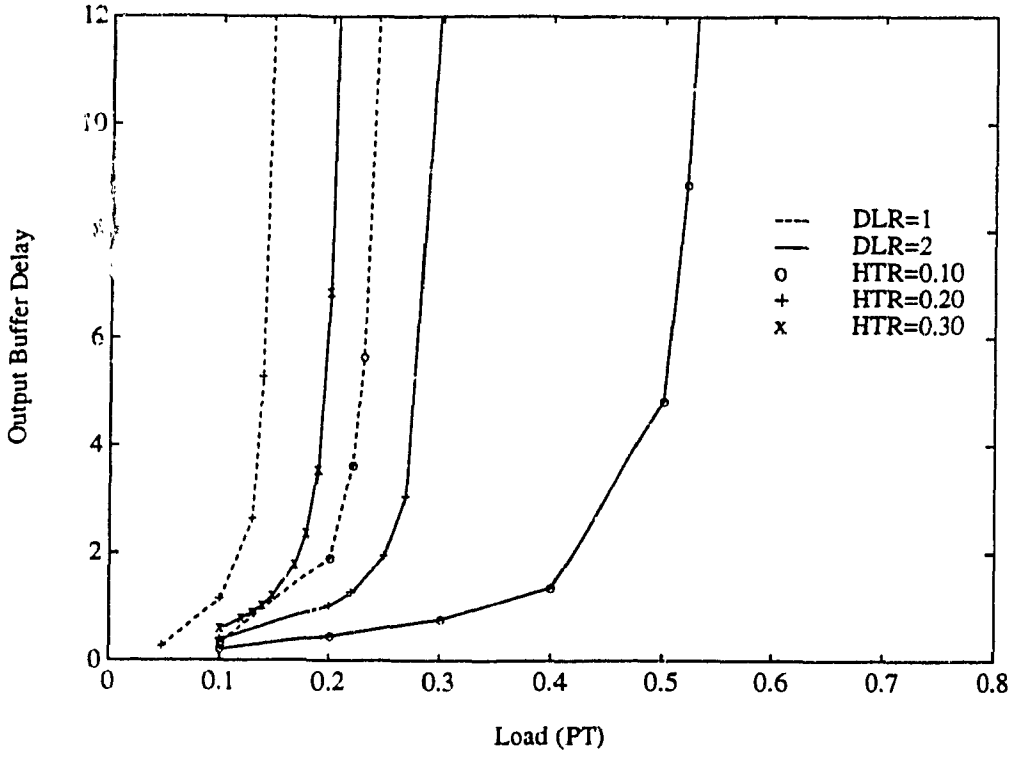


Fig. 4.46. Output buffer delay (*L*=3).

however this shrinks to 0.19 as HTR is increased to 0.30. The results for the case when the switch speedup is four are shown in Figures 4.47-4.50. Fig. 4.47 shows that for HTR=0.1 a buffer length of 31 is sufficient with *DLR*=3 to reduce the loss to $1x10^{-6}$ for a load of 0.7. As HTR is increased however, as shown in Fig. 4.48 and Fig. 4.49 for HTR=0.2 and HTR=0.3 respectively, the performance decreases rapidly even with *DLR*=3. For HTR=0.2 with *DLR*=3 a buffer of 34 is now needed to carry a load of just 0.40 at a loss of $1x10^{-6}$. At HTR=0.3 with *DLR*=3, even a buffer of length 60 is insufficient to limit the loss to less than $1x10^{-4}$ at a load of 0.3. The results for the delay are shown in Fig. 4.50. It is seen that while the load can be extended to 0.75 for *DLR*=3 and HTR=0.1 increasing HTR to 0.3 quickly reduces the maximum load to 0.27. From the different results it can be concluded that for HTR<0.1 the *DLR* may be reduced by at least one factor bellow that of the switch speedup when $L>2$ and provide enough throughput so that the bottleneck of the system remains at the input. However, for all other cases the output buffer saturates at very low loads even with *DLR* just one factor bellow that of the switch speedup so that the output queue becomes the bottleneck of the system. Under these circumstances, there is no alternative but to eliminate output buffering altogether and make *DLR*=*L*.

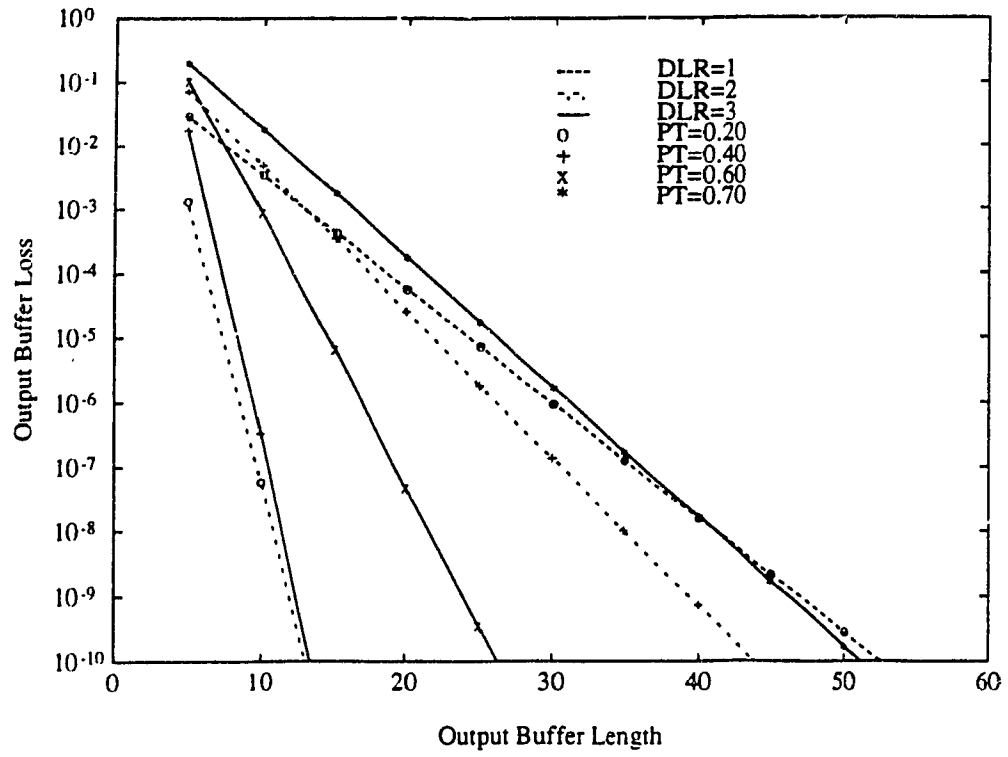


Fig. 4.47. Output buffer loss versus buffer length (L=4, HTR=0.1).

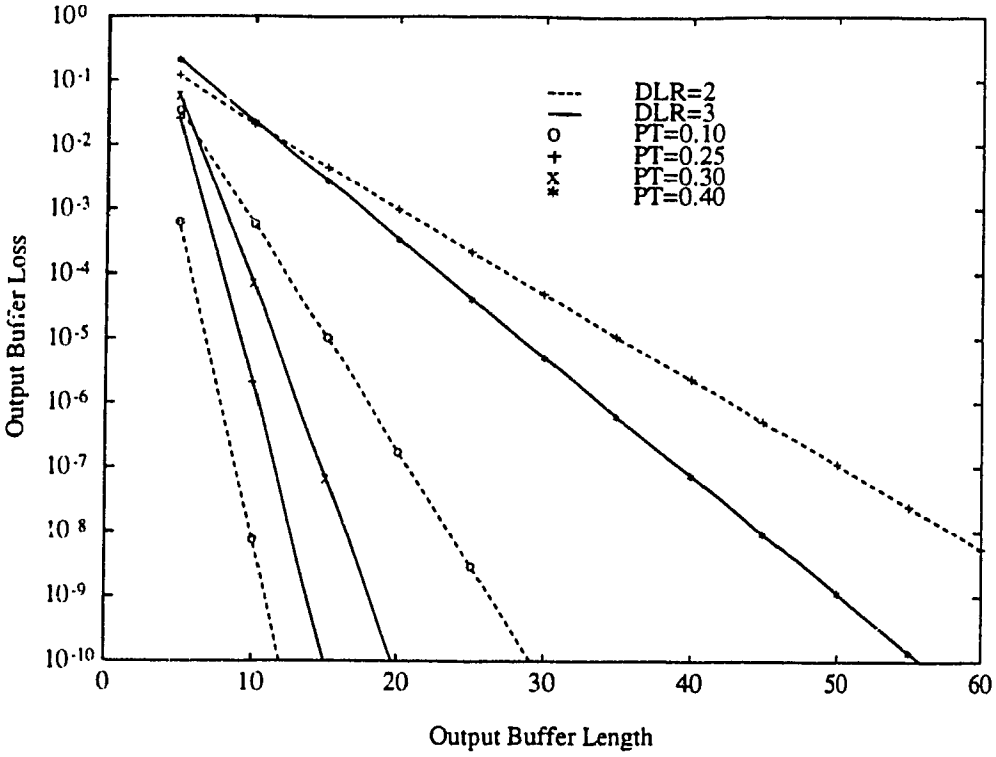


Fig. 4.48. Output buffer Loss versus buffer length (L=4, HTR=0.2).

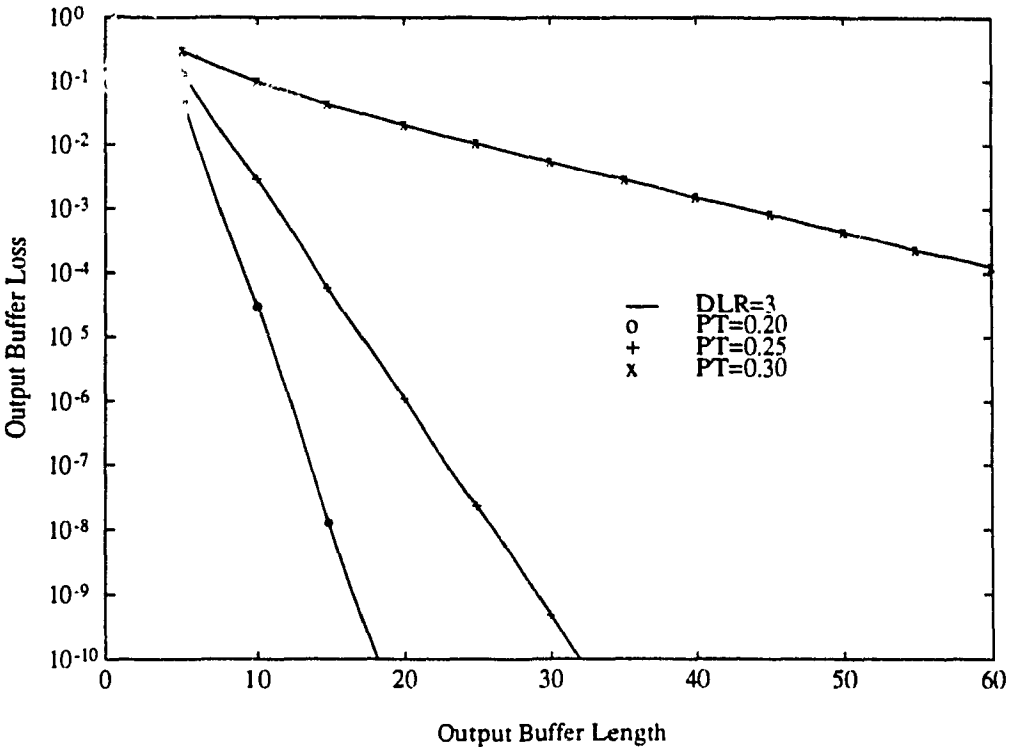


Fig. 4.49. Output buffer loss versus buffer length (L=4, HTR=0.3).

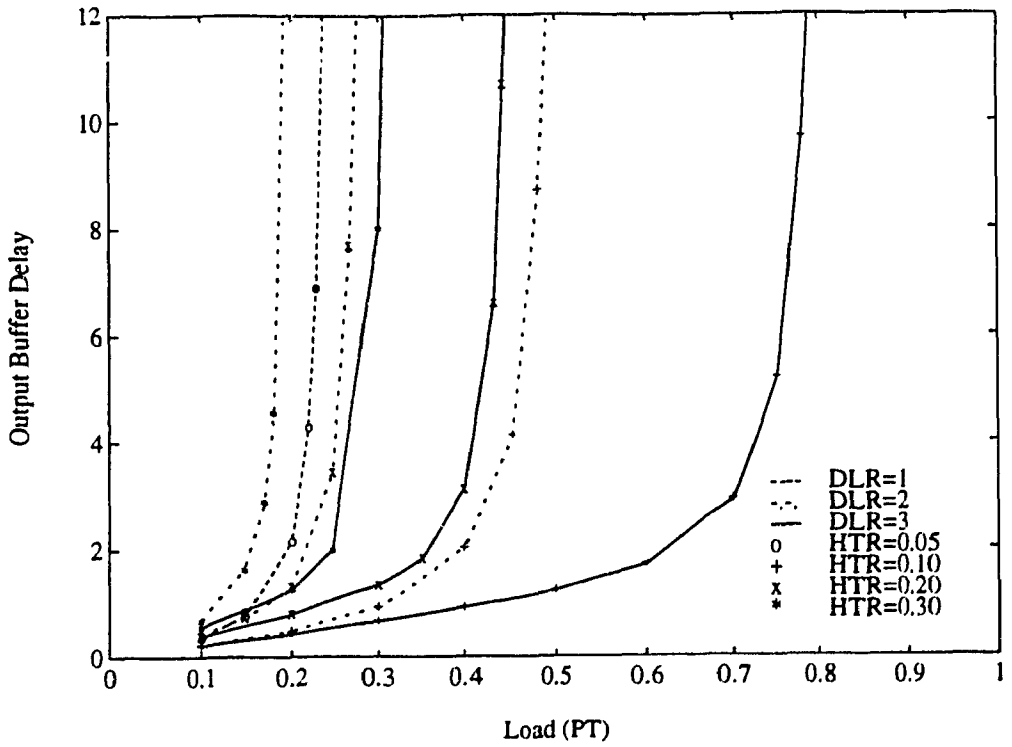


Fig. 4.50. Output buffer delay (L=4).

4.4.5. Effect of Buffering on the Performance as the Switch Size is Increased

In this section the performance of the switch is investigated as a function of the switch size. All three scenarios, namely discard, with buffering, and buffering with HLR are investigated to determine the effect of buffering and the HLR algorithm on the performance degradation as the size of the switch is increased. In all cases it is assumed that the input buffer length is 30 and that HTR=0.2. The results for video traffic are shown in Fig. 4.51 and Fig. 4.52 for a load of 0.6 and 0.8 respectively. As shown in Fig. 4.51 the degradation is significantly less dependent on the switch size with buffering compared to that of the discard case. In the discard case with L=8 for example the video loss probability increases rapidly from $1x10^{-8}$ for N=22 to $1x10^{-5}$ for N=57. However with buffering and L=1 the loss increases only from $4x10^{-7}$ for N=22 to $7x10^{-6}$ for N=57. Further for N=60, increasing L to two as shown limits the loss to less than $5x10^{-8}$ and increasing L to three limits the loss to less than $8x10^{-9}$. The slope of the degradation between N=60 and N=25 is also seen to decrease as L is increased, where it is seen that for N=25 the loss is $6x10^{-9}$ and $4x10^{-9}$ for L=2 and L=3 respectively. As the load is increased to 0.8, it is seen that the discard case with L=8 now outperforms that of the buffered case due to buffer saturation, as was seen previously. However, even at such a high load the buffered case still is relatively independent of the switch size. Turning now to the next lower priority, voice users, it is seen in Fig. 4.53 that for a fixed load of 0.5 that the L=1 buffered case continues to outperform the discard case with L=8 and that at N=60 there is at least a two order difference in the loss probability. At a higher load of 0.7, as shown in Fig. 4.54, L has to be increased to at least two in order to compete with

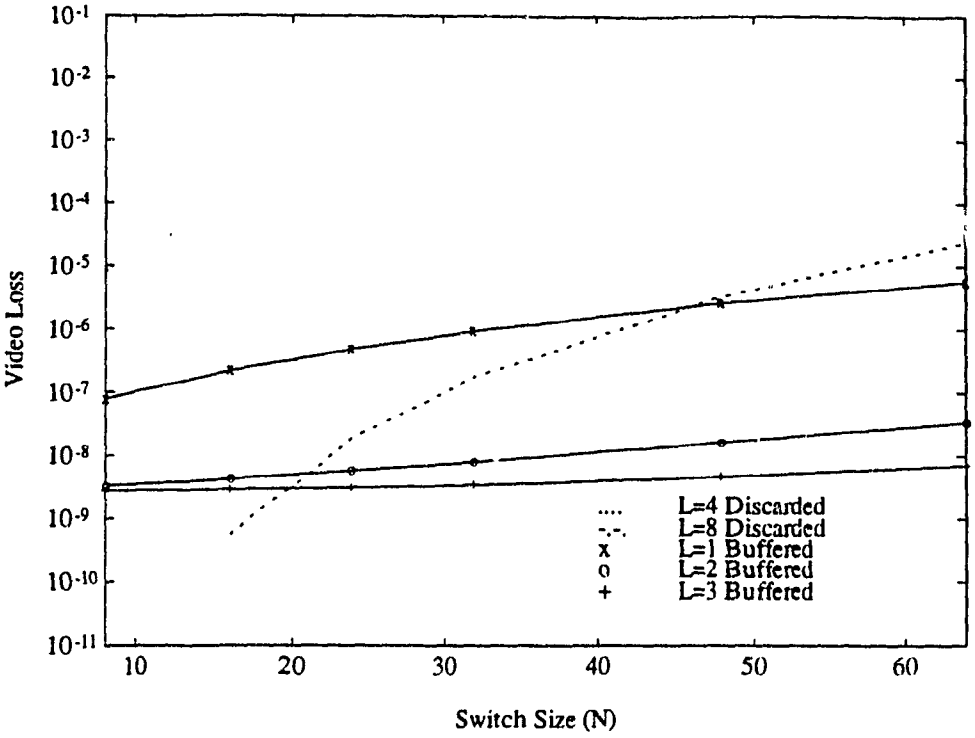


Fig. 4.51. Video loss as N is increased (HTR=0.2, Load=0.6).

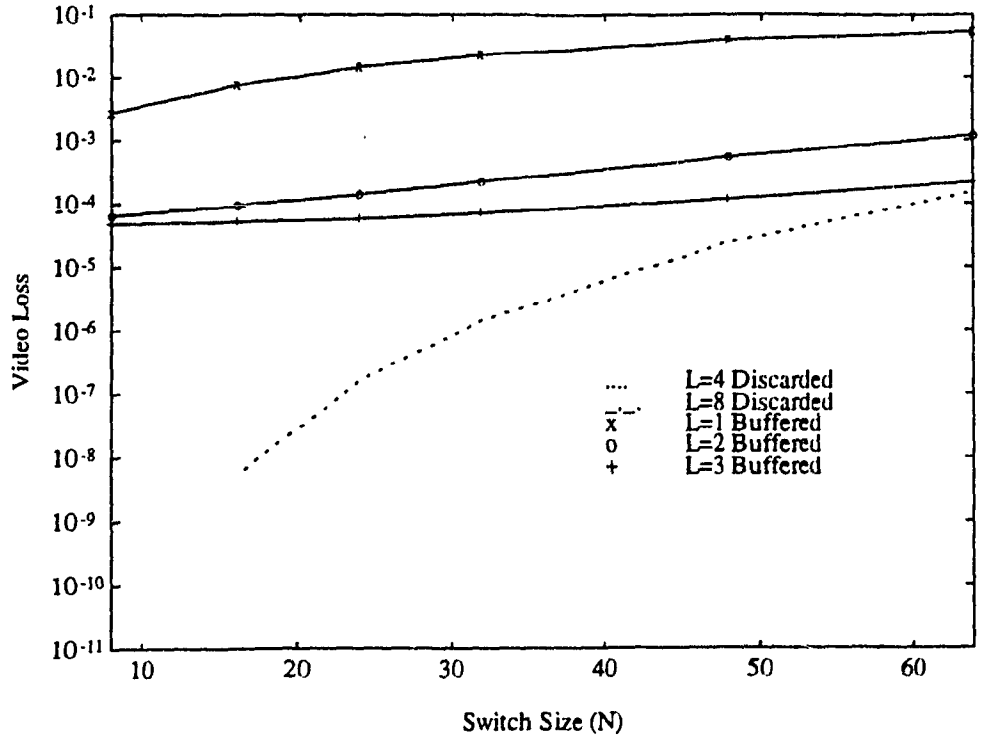


Fig. 4.52. Video loss as N is increased (HTR=0.2, Load=0.8).

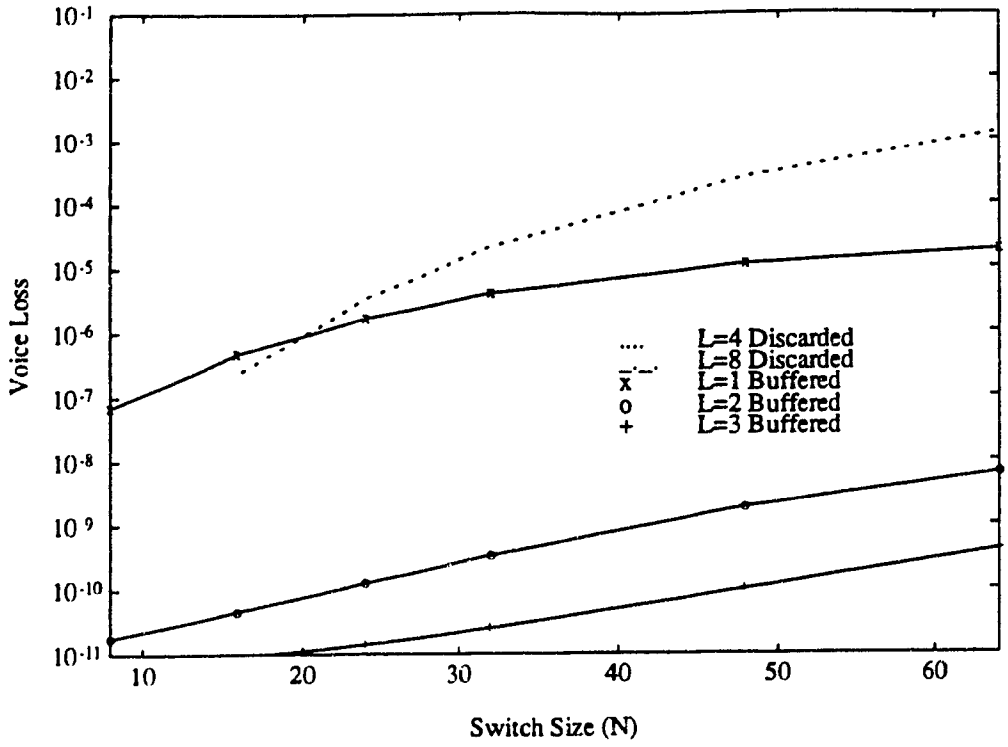


Fig. 4.53. Voice loss as N is increased (HTR=0.2, Load=0.5).

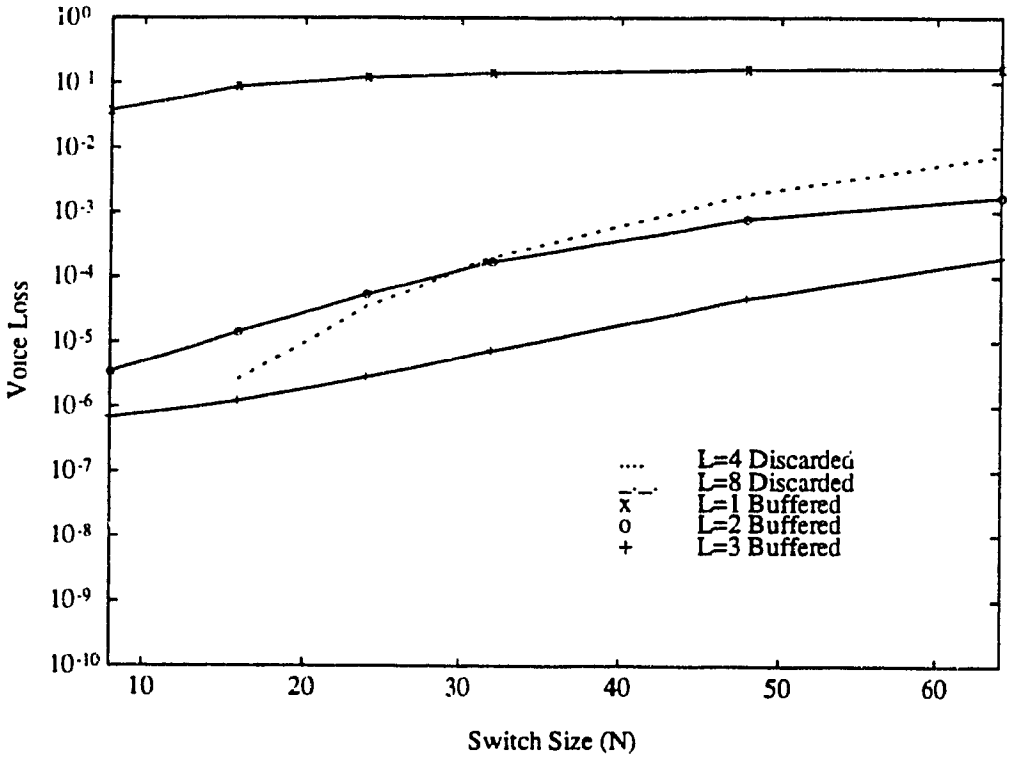


Fig. 4.54. Voice loss as N is increased (HTR=0.2, Load=0.7).

the L=8 discard case. In each case though, the discard results deteriorate much faster than those with input buffering.

For file data users the results with and without HLR for a load of 0.4 are shown in Fig. 4.55. It is seen that the HLR algorithm further reduces the dependence on N by increasing the service to the input buffer. At this load it is seen that even with L=1 the buffered case outperforms the L=8 discard for all switch sizes. Increasing the load to 0.6, as shown in Fig. 4.56, it is seen that the HLR algorithm improves the performance substantially especially as N increases, even at a speedup of three. At L=3 with a switch size of 25 it is seen that the loss is reduced from 7×10^{-8} without HLR to 2×10^{-8} with HLR but at a switch size of 60 the improvement has increased, with the loss being 9×10^{-6} without HLR and 7×10^{-7} with HLR. It is also seen that the L=2 case with HLR for switch sizes above 42, at this load, outperforms the L=3 case without HLR. Intuitively, this is expected at high loads and high hotspot traffic ratios since as the switch size increases the blocking rapidly increases at the hotspot output and increasing L has less of an effect on the service rate. However with HLR, while the initial packet going to the hotspot output may be blocked a sister packet destined to a uniform output will most likely be successful, even at a lower speedup.

The results at a load of 0.4 for interactive data users are shown in Fig. 4.57. Here the improvement due to HLR is already apparent at this low load, with the improvement increasing rapidly as the speedup is reduced and as N is increased. At a higher load of 0.6, as shown in Fig. 4.58, the difference between the HLR case and that without HLR is seen to be much greater. For L=3 and a switch size of 25 the loss is reduced from

$1x10^{-6}$ without HLR to $1x10^{-7}$ with HLR, while at a switch size of 60 the loss is reduced from $2.5x10^{-5}$ without HLR to $9x10^{-6}$ with HLR. For interactive data users at this load with L=2 the HLR algorithm again outperforms the L=3 buffered case without HLR with the threshold switch size having decreased to 32.

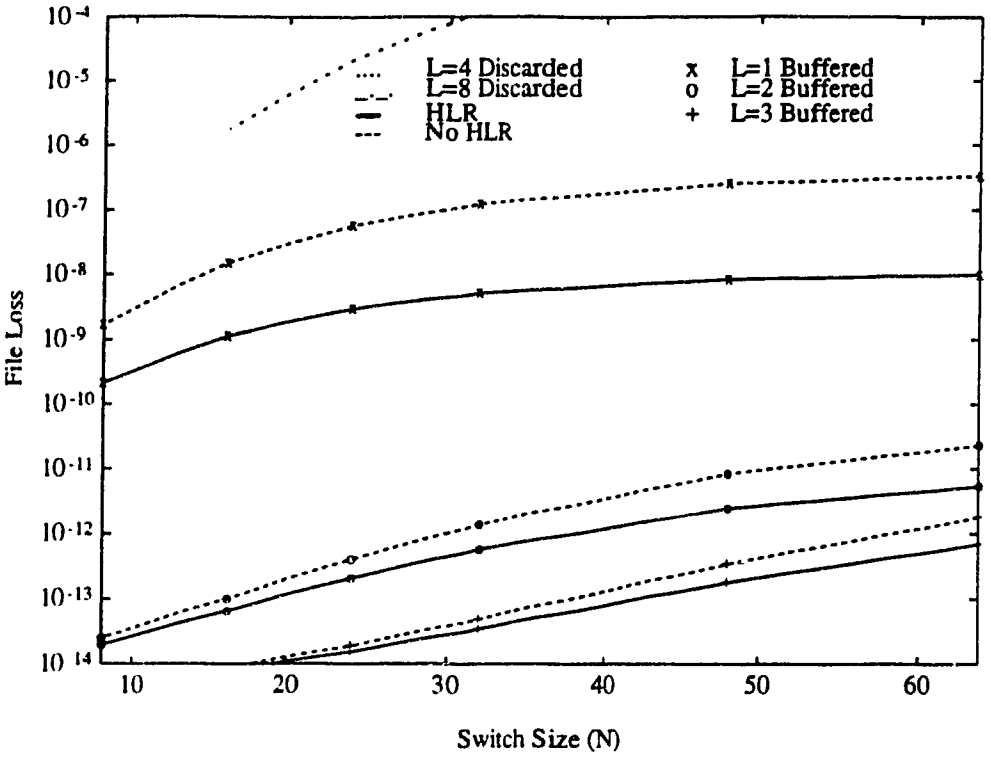


Fig. 4.55. File loss as N is increased (HTR=0.2, Load=0.4).

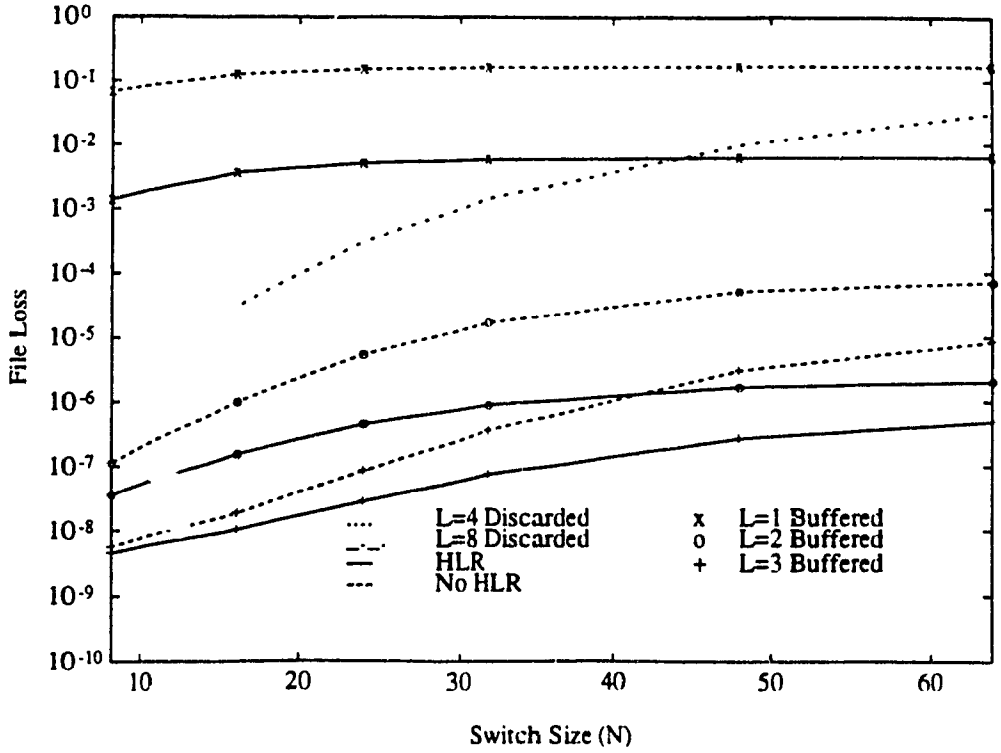


Fig. 4.56. File loss as N is increased (HTR=0.2, Load=0.6).

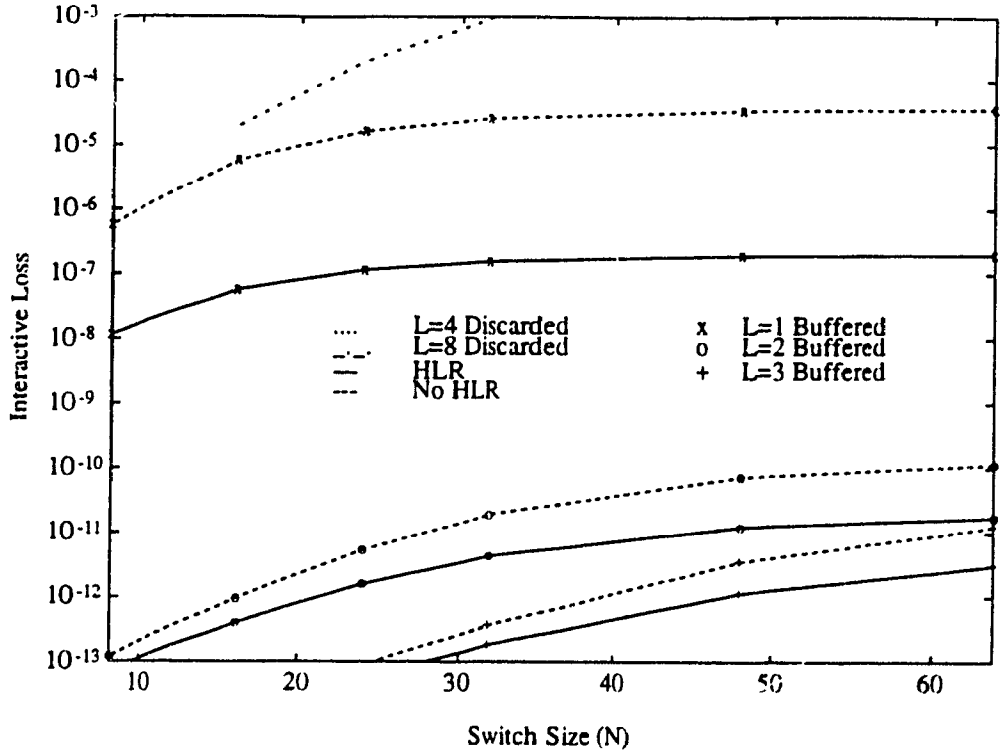


Fig. 4.57. Interactive loss as N is increased (HTR=0.2, Load=0.4).

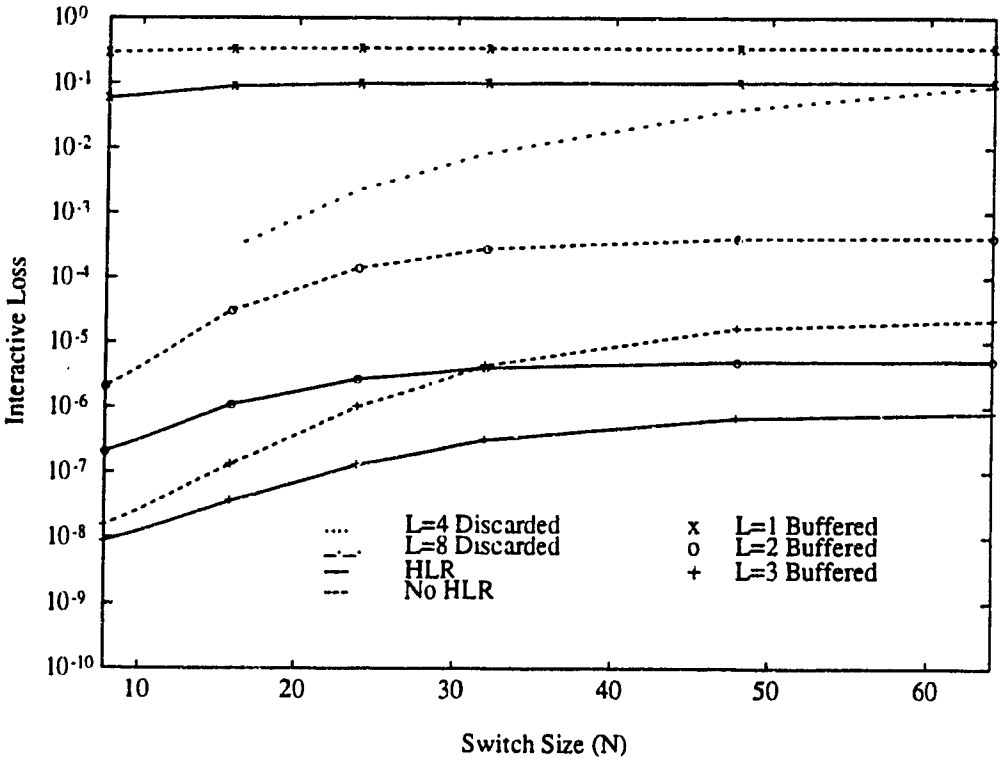


Fig. 4.58. Interactive loss as N is increased (HTR=0.2, Load=0.6).

Chapter 5

Conclusions and Suggestions for Further Research

5.1. Conclusions

A movable boundary/random-demand assigned access scheme was presented in Chapter 2 as a means of accommodating a large number of users in a future integrated services VSAT environment. This scheme tried to balance the needs of video, voice and file data users which have relatively long call durations and high utilizations with those of bursty interactive data users. To this end video, voice and file data were provided with a connection type of service while the interactive data users were provided with a connectionless service. In the MB/R-DAMA scheme connection users used a reservation mechanism based on Slotted Aloha random access to reserve slots on the frame for the duration of their call. In this way, a high throughput and a constant delay could be maintained at the expense of an initial round trip propagation delay for the reservation. To ensure a low delay for the sporadic interactive data users Slotted Aloha random access was combined with a moving boundary policy in order to make efficient use of the available frame capacity. Under this policy, interactive users were allowed to compete in both their dedicated portion of the frame as well as in any unused connection user slots.

The analysis showed that video traffic due to its high data rate and low generating rate was independent of the other traffics, while the dependence on the number of allowable voice and file data users was heavily dependent on video traffic as well as the traffic generated by one another. However, it was seen that with careful design of the

frame a single 150 Mbps channel has the capability of integrating services for user populations in the thousands to tens of thousands within the boundary of a single beam. Assuming a 16 ms frame, it was seen that it is possible to accommodate up to 20 video users at 45 Mbps, $5x10^3$ voice and $1x10^4$ file data users at 64 Kbps and $1x10^4$ interactive data users at 1 Kbps. Alternatively, the number of voice and file data users could be rearranged with the frame supporting $7.5x10^3$ voice and $2x10^3$ file data users. By reducing the number of video users to 10, the extra capacity could be used to accommodate up to $8x10^3$ voice users and $1x10^4$ file and interactive data users respectively. It was also seen that the amount of overhead incurred in the frame to implement the reservation scheme was minimal with only one minislot required per frame to support up to 100 video users at a delay of 0.271 seconds while 5 and 50 minislots were sufficient for $1x10^4$ file and voice users, respectively, at a delay of 0.3 seconds. For interactive users, it was seen that by designing the frame with 100 dedicated slots the frame could support a minimum population of $1x10^4$ users at a delay of 0.3 seconds when all the connection capacity was being used. At lower utilizations of the connection capacity, the moving boundary scheme substantially increased the potential interactive user population. At 50% utilization of the connection users frame capacity, the population could be increased to $1x10^5$ at a mean delay of 0.3 seconds, while at 0% utilization, a population of $2.5x10^5$ interactive users potentially could be accommodated.

With the integration of services in a future on-board switching satellite the on-board packet switch will have to implement a priority structure in order to meet the various delay and throughput requirements of the different services. To this end, a new

priority Knockout switch with input and output buffering and a head of line resolution algorithm was presented in Chapter 3, along with an analysis under uniform traffic conditions. The results under uniform traffic showed that with moderate buffering and speedup, it was possible to drastically reduce the loss probability which quickly deteriorated for the discard case as the priority level decreased. For the lowest priority interactive data users, an input buffer length of 30 was still sufficient to reduce the loss to $1x10^{-6}$ for a speedup of three at a load of 0.7, while the discard case required a speedup of eight to match this same loss. It was also seen in the buffered case that as the priority decreased from video to interactive data the speedup only had to be increased from L=2 to L=3 in order to maintain a loss of $1x10^{-6}$ at a load of 0.7. In contrast, in the discard case the speedup had to be increased from L=4 to L=8 in order to maintain the same loss.

The HLR algorithm also played a major role in extending the throughput and reducing the loss for file and interactive data users at low speedups when the switch blocking probability was high. For file users, the HLR algorithm increased the maximum load for L=1, at a delay of 20, from 0.56 to 0.61 and for L=2 from 0.79 to 0.84. As the speedup increased to three, however, the speedup reduced the blocking enough so that very little could be gained from the use of HLR.

At the output buffer it was seen that a moderate buffer length of 30 with DLR=1 was sufficient to limit the loss to $1x10^{-6}$ for loads up to 0.80 with L as high as four. However, as the downlink ratio was increased to two the output buffer length could be reduced to less than 15 for L=4, while still providing a loss of $1x10^{-11}$ for a load as high

as 0.90. This loss was at least six orders smaller than the interactive input buffer loss at this load, thereby insuring that the bottleneck of the system remained at the input buffer.

While the uniform traffic analysis yielded important information about the design of the on-board switching system, it represented the best case scenario. In future integrated networks nonuniform traffic patterns will arise and the system has to be designed in order to maintain an acceptable service under such conditions. One of the most severe types of nonuniform traffic, namely hotspot traffic, was analyzed in Chapter 4 and it was shown that with buffering, HLR and speedup of three that the performance could be substantially reduced over that of the discard case. For video users it was seen that there was already a large difference between the buffered case and the discard case with the discard case requiring a speedup of eight at HTR=0.3 for a load of 0.8 in order to match the loss of the L=3 buffered case with a buffer length of only 30. As the priority level decreased, the discard loss increased rapidly. For voice, it was seen that the discard case with L=8 could only maintain a load of 0.2 for HTR=0.3 at a loss of $1x10^{-6}$ but with buffering and L=3, the load could be extended to 0.61. For both file and interactive data users HLR was shown to be required with L=3 to maintain acceptable losses. For file users the load had to be restricted to only 0.15 for the L=8 discard case at HTR=0.3 in order to maintain a loss of $1x10^{-6}$, while the buffered case with HLR continued to maintain a load of 0.59 with L=3 for the same loss. Similarly, for interactive data the discard case with L=8 could only maintain a load of 0.10 at HTR=0.3 for a loss of $1x10^{-6}$, while the buffered case with HLR continued to maintain a load of 0.56.

It was also shown that while the input buffer maintained a relatively high

throughput, the hotspot output buffer saturated at very low loads and low values of HTR due to the high traffic. In general, it was concluded that when HTR<0.1 the value of *DLR* may be reduced by at least one factor below that of the switch speedup for $L>2$ and continue to provide enough service so that the output buffer overflow remained lower than that at the input. However, for all other cases, the output buffer saturated at very low loads and the output became the bottleneck of the system. Under these conditions, there was no alternative but to eliminate output buffering altogether and set $DLR=L$.

Finally it was shown that moderate input buffering could drastically reduce the deterioration in performance of the discard case as the switch size was increased, thereby making it possible to accommodate larger switch sizes. For both file and interactive data, it was seen that HLR further reduced the dependence on N, especially at high loads. It was also seen that under high loads, as the switch size increased, the HLR algorithm outperformed the case without HLR for an increase in speedup of one. This indicated that there was a threshold where increasing L had less of an effect than the success probability of sister packets at the uniform output ports, due to the large blocking probability at the hotspot output.

5.2. Suggestions for Further Study

To examine the operation of the uplink performance further, the open-loop blocked call cleared model that was assumed in the uplink could be extended to include call buffering on-board the satellite. In addition, the performance of the on-board switch could be extended to include different mixtures of uniform, hotspot and point-to-point traffic in

order to examine the effect of a priority structure under even more severe blocking scenarios. Dynamic buffer allocation could also be explored as a means of coping with buffer constraints. A more sophisticated head of line resolution algorithms could also be implemented in which information about the state of the input and output buffers is used to select the sister packet. Because of the added complexity however, any such adaptations would most likely have to be examined through the use of simulation.

References

[1] R. Lovell and C.L. Cuccia, "Global interconnectivity in the next two decades," *12th Int. Sat. Syst. Conf.*, Arlington VA., March 1988, pp. 39-49.

[2] L.C. Palmer *et al.* "A personal Communications Network using a Ka-band satellite," *IEEE J. Select Areas in Commun.*, vol. 10, no. 2, pp.401-417, Feb. 1992.

[3] C. E. Mahle, G. Hyde and T. Inukai, "Satellite scenarios and technology for the 1990's," *IEEE J. Select. Areas Commun.*, vol. SAC-5, pp. 556-570, May 1987.

[4] F. M. Naderi and W.W. Wu, "Advanced satellite concepts for future generation VSAT networks," *IEEE Commun. Mag.*, pp. 13-21, Sept. 1988.

[5] S. J. Campanella, B. Pontano and H. Chalmers, "Future switching satellites," *12th Int. Sat. Syst. Conf.*, Arlington VA., March 1988, pp. 264-273.

[6] W. Berner, W. Grassman, M. Piontek, "MOBS - A modular on-board switching satellite", *Globecom'88 Conf. Rec.*, 1988, pp. 1769-1773.

[7] S.E. Dinwiddy, "Advanced on-board processing satellite system concept", *Proc. Int. Symp. on Satellites*, Austria, 25-27 Sept. 1985.

[8] G. Pennoni, "A TST/SS-TDMA telecommunications system", *ESA Journal*, 1984, vol. 8.

[9] M. Horstein, "Advanced communications architecture study," TRW contract NAS3-24743, Prepared for NASA Lewis, 1986.

[10] M. Horstein, "A satellite system architecture for single-hop VSAT networks," *12th Int. Sat. Syst. Conf.*, Arlington VA., March 1988, pp. 625-635.

[11] D. E. Santarpia and J.W. Bagwell, "Status of NASA's multibeam communications technology program," *Eascon'83*, Washington D.C., Sept. 1983, pp. 53-68.

[12] G. Beck, "The NASA advanced communications technology satellite (ACTS)," *RCA Engineer*, 30-1, pp. 29-37, Jan./Feb. 1985.

[13] L.C. Palmer and L.W. White, "Demand Assignment in the ACTS LBR system," *IEEE Trans. Commun.*, vol. 38, pp. 684-692, May 1990.

[14] W.M. Holmes and G.A. Beck, "The ACTS flight system cost-effective advanced communications technology," *Int. Commun. Sat. Syst. Conf.*, Orlando Florida, March 1984, pp. 196-201.

[15] D. Raychaudhuri and K. Joseph, "Ku-band satellite data networks using very small aperture terminals-part I: multi-access protocols," *Int. J. Satell. Commun.*, vol. 5., pp. 198-212, 1987.

[16] D. Raychaudhuri, "Selective reject Aloha/FCFS: an advanced VSAT channel access protocol," *Int. J. Satell. Commun.*, vol. 7, pp. 435-447, 1989.

[17] C.F. Pavey, R.P. Rice and E.J. Cummins, "A performance evaluation of the PDAMA satellite access protocol," *INFOCOM'86 Conf. Proc.*, pp. 580-589.

[18] D. Raychaudhuri and K. Joseph, "Channel access protocols for Ku-band VSAT networks: a comparative evaluation," *IEEE Commun. Mag.*, vol. 26, no. 5, pp.34-44, May 1988.

[19] T. Suda, H. Miyahara and T. Hasegawa, "Performance Evaluation of an Integrated access scheme in a satellite communication channel," *IEEE J. Select Areas Commun.*, vol. SAC-1, no. 1, Jan. 1983.

[20] H.W. Lee and J.W. Mark, "Combined random/reservation access for packet-switched transmission over a satellite with on-board processing-part II: multibeam satellite", *IEEE Trans. Commun.*, vol. COM-32, no. 10, pp. 1093-1104, Oct. 1984.

[21] D. Spears, "Broadband ISDN switching capabilities from a service perspective," *IEEE J. Select Areas Commun.*, vol. SAC-5, no. 8., pp. 1222-1230, Oct. 1987.

[22] T.M. Chen and D.G. Messerschmitt, "Integrated voice/data switching," *IEEE Commun. Mag.*, vol. 26, no. 6, pp. 16-25, June 1988.

[23] J.G. Gruber and N. H. Lee, "Performance requirements for integrated voice/data networks," *IEEE J. Select Areas Commun.*, vol. SAC-1, no. 6, pp. 981-1005, Dec. 1983.

[24] L. Kleinrock, *Queueing Systems Vol. 1: Theory*, New York: Wiley, 1976.

[25] U.N. Bhat and M.J. Fischer, "Multichannel queuing systems with heterogeneous classes of arrivals," *Naval Res. Logist. Quarterly*, vol. 23, no. 2.,pp. 271-283., 1976.

[26] H. Ahmadi and W.E. Denzel, "A survey of modern high-performance switching techniques," *IEEE J. Select. Areas Commun.*, vol. 7., pp. 1091-1103, Sept. 1989.

[27] J.H. Patel, "Performance of processor-memory interconnections for multiprocessors," *IEEE Trans. Commun.*, vol. C-30, pp. 771-780, Oct. 1981.

[28] M. Kumar and J.R. Jump, "Performance of unbuffered shuffle-exchange networks," *IEEE Trans. Comput.*, vol. C-35, no. 6, pp. 573-577, June 1986.

[29] Y. Jenq, "Performance analysis of a packet switch based on a single-buffered banyan network," *IEEE J. Select. Areas Commun.*, vol. SAC-1, pp. 1014-1021, Dec. 1983.

[30] D.M. Dias and J.R. Jump, "Analysis and simulation of buffered delta networks," *IEEE Trans. Comput.*, vol. C-30, no. 4, pp. 273-282, April 1981.

[31] T. Theimer, E. Rathgeb and M. Huber, "Performance analysis of buffered banyan networks," *IEEE Trans. Commun.*, vol. 39, pp. 269-277, Feb. 1991.

[32] M.N. Huber, E. P. Rathgeb and T.H. Theimer, "Banyan networks in an ATM environment," *Proc. ICCC'88*, Tel Aviv, Israel, 1988, pp. 167-174.

[33] E. P. Rathgeb, T.H. Theimer and M.N. Huber, "Buffering concepts for ATM switching networks," *GLOBECOM'88 Conf. Rec.*, 1988, pp. 1277-1281.

[34] C.P. Kruskal and M. Snir, "The performance of multistage interconnection networks for multiprocessors," *IEEE Trans. Comput.*, vol. C-32, pp. 1091-1098, Dec. 1983.

[35] C.P. Kruscal, M. Snir and A. Weiss, "The distribution of waiting times in clocked multistage interconnection networks for multiprocessors," *IEEE Trans. Comput.*, vol. C-37, pp. 1337-1352, Nov. 1988.

[36] K.E. Batcher, "Sorting networks and their applications," *Spring Joint Comput. Conf. Proc.*, AFIPS, 1968, pp. 307-314.

[37] A. Huang and S. Knauer, "Starlite: A wideband digital switch," *GLOBECOM'84 Conf. Rec.*, Atlanta, GA, Dec. 1984, pp. 121-125.

[38] J. Hui and E. Arthurs, "A broadband packet switch for integrated transport," *IEEE J. Select. Areas Commun.*, vol. SAC-5, pp. 1264-1273, Oct. 1987.

[39] Y.S. Yeh, M.G. Hluchyj and A.S. Acampora, "The Knockout switch: A simple modular architecture for high-performance packet switching," *IEEE J. Select. Areas Commun.*, vol. SAC-5, pp. 1274-1283, Oct. 1987.

[40] K.Y. Eng, M.G. Hluchyj and Y.S. Yeh, "A knockout switch for variable-length packets," *IEEE J. Select. Areas Commun.*, vol. SAC-5, pp. 1426-1435, Dec. 1987.

[41] K.Y. Eng, "A photonic knockout switch for high-speed packet networks," *IEEE J. Select. Areas Commun.*, vol. 6., pp. 1107-1115, Aug. 1988.

[42] M.J. Karol, M.G. Hluchyj and A.P. Morgan, "Input versus output queueing on a space-division packet switch," *IEEE Trans. Commun.*, vol. COM-35, pp. 1347-1356, Dec. 1987.

[43] H.G. Hluchyj and M.J. Karol, "Queueing in high-performance packet switching," *IEEE J. Select Areas Commun.*, vol. 6, no. 9, pp. 1587-1597, Dec. 1988.

[44] J.S.Chen and T.E. Stern, "Optimal buffer allocation for packet switches with input and output queueing," *ICC'90 Conf. Proc.*, pp. 1936-1941.

[45] I. Iliadis and W.E. Denzel, "Performance of packet switches with input and output queueing," *ICC'90 Conf. Proc.*, 1990, pp. 747-753.

[46] Y. Oie *et al.*, " Effect of Speedup in nonblocking packet switch," *ICC'89 Conf. Proc.*, 1989, pp. 410-414.

[47] L.K. Platzman, J.C. Ammons, and J.J. Bartholdi III, "A simple and efficient algorithm to compute tail probabilities from transforms," *Operat. Res.*, vol. 36, no. 1, pp. 137-144, Jan. 1988.

[48] S.C. Liew, "Performance of input-buffered and output-buffered ATM switches under bursty traffic: simulation study", *ICC'90 Conf. Proc.*, 1990, pp. 1919-1925.

[49] U. Garg and Y. Huang, "Decomposing banyan networks for performance analysis,", *IEEE Trans. Compt.*, vol. C-37, pp.371-376, March 1988.

[50] G.F. Pfister, V.A. Norton, "Hot-Spot contention and combining in multistage interconnection networks", *IEEE Trans. on Comput.*, vol. C-34, no. 10, pp. 943-948, Oct. 1985.

[51] H.S. Kim, A. Leon-Garcia, "Performance analysis of buffered banyan networks under nonuniform traffic patterns," *IEEE Trans. Commun.*, vol. 38, pp. 648-658, May 1990.

[52] H. Yoon, T. Liu and K.Y. Lee, "The knockout switch under nonuniform traffic," *GLOBECOM'88 Conf. Rec.*, 1988, pp. 1628-1634.

[53] S.Q. Li and M.J. Lee, "A study of traffic imbalances in a fast packet switch," *INFOCOM'89 Conf. Proc.*, pp. 538-547.

[54] J.S. Chen and T.E. Stern, "Throughput analysis, optimal buffer allocation, and traffic imbalance study of a generic nonblocking packet switch." *IEEE J. Select Areas Commun.*, vol. 9, pp. 439-449, April 1991.

Appendix A

Discrete-Time M/D/1 Queue

The discrete-time M/D/1 queue system, shown in Fig. A.1, can be described by a binomial input process, A, which has the binomial probabilities

$$a_i = \Pr[A=i] = \binom{N}{i}\left(\frac{p}{N}\right)^i\left(1-\frac{p}{N}\right)^{N-i} \quad i=0,1,2,\ldots,N \tag{A.1}$$

with a probability generating function

$$A(z) \doteq \sum_{i=0}^{N} Z^i a_i = \left(1-\frac{P}{N}+z\frac{p}{N}\right)^N. \tag{A.2}$$

As $N\rightarrow\infty$, a_i becomes a poisson distribution

$$a_i \doteq \frac{p^i e^{-P}}{i!} \quad i=0,1,2,\ldots \tag{A.3}$$

with a probability generating function equal to

$$A(z) = \sum_{i=0}^{N} z^i a_i = e^{-p(1-z)}. \tag{A.4}$$

Let Q_m denote the number of packets at the end of the mth time slot and A_m denote the number of packet arrivals during the mth time slot. Then Q_m can be expressed as

$$Q_m = \max\left(0, Q_{m-1}+A_m-1\right). \tag{A.5}$$

The probability generating function of Q_m can be found as

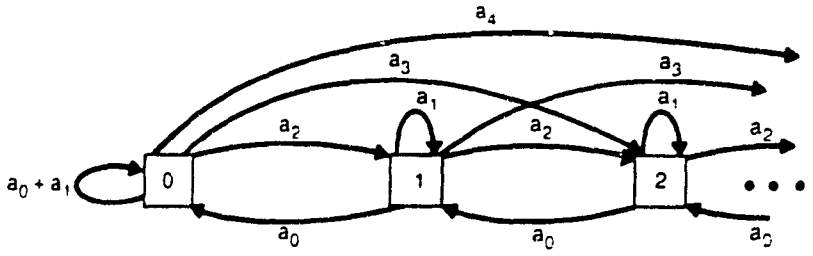


Fig. A.1. Discrete-time M/D/1 queue chain [42].

$$Q(z) = \frac{(1-p)(1-z)}{A(z)-z}. \tag{A.6}$$

Substituting in $A(z)$, differentiating with respect to z and taking the limit as z goes to one, the mean steady state queue size is found as

$$\lim_{z\to 1} \frac{dQ(Z)}{dz} = \overline{Q} = \frac{(N-1)}{N}\frac{p^2}{2(1-p)} = \frac{N-1}{N}\overline{Q}_{M/D/1} \tag{A.7}$$

where $\overline{Q}_{M/D/1}$ denotes the mean queue size for a M/D/1 queue. As N goes to infinity, $\overline{Q} \to \overline{Q}_{M/D/1}$. The probability generating function similarly converges to that of and M/D/1 queue

$$\lim_{N\to\infty} Q(z) = \frac{(1-p)(1-z)}{e^{-p(1-z)} - z}. \tag{A.8}$$

By expanding eq. (A.8) in a Maclaurin series the queue size distribution, q_i, can be found.

$$
\begin{aligned}
q_{i=0} &= \Pr(Q=0) = (1-p)e^{P} \\
q_{i=1} &= \Pr(Q=1) = (1-p)e^{p}(e^{p}-1-p) \\
&\vdots \\
q_{i=n} &= \Pr(Q=n) = (1-p)\sum_{j=1}^{n+1}(-1)^{n+1-j}e^{jp}\left[\frac{(jp)^{n+1-j}}{(n+1-j)!}+\frac{(jp)^{n-j}}{(n-j)!}\right] \quad \text{for } n\geq 2 .
\end{aligned}
\tag{A.9}
$$

While eq. (A.9) is a closed form expression, for large values of n eq. (A.9) leads to inaccurate results due to the alternating series which expresses small steady state probabilities as the difference between very large numbers. More accurate results are obtained by solving the markov chain numerically from the steady state balance equations [42].

Appendix B

Analytic Solution of the Discrete-Time Input Buffer System Markov Chain

Taking the video input buffer as an example, with arrival probability P_W and service probability U_W, the main transition probabilities shown in Fig. 3.27 can be abbreviated as

$$\dot{P_W} = P_W\left(1-U_W\right) \tag{B.1}$$

$$\dot{U_W} = U_W\left(1-P_W\right). \tag{B.2}$$

The balance equations can then be written as

$$\begin{aligned}
\Phi_{W_0} &= (1-P_W)\Phi_{W_0} + \dot{U_W}\Phi_{W_1} \\
\Phi_{W_1} &= P_W\ \Phi_{W_0} + \left(1-\dot{P_W}-\dot{U_W}\right)\Phi_{W_1} + \dot{U_W}\Phi_{W_2} \\
\Phi_{W_2} &= \dot{P_W}\Phi_{W_1} + \left(1-\dot{P_W}-\dot{U_W}\right)\Phi_{W_2} + \dot{U_W}\Phi_{W_3} \\
\Phi_{W_3} &= \dot{P_W}\Phi_{W_2} + \left(1-\dot{P_W}-\dot{U_W}\right)\Phi_{W_3} + \dot{U_W}\Phi_{W_4} \\
&\ \ \vdots \qquad\qquad\qquad \vdots \\
\Phi_{W_{SI-1}} &= \dot{P_W}\Phi_{W_{SI-2}} + \left(1-\dot{P_W}-\dot{U_W}\right)\Phi_{W_{SI-1}} + \dot{U_W}\Phi_{W_{SI}} \\
\Phi_{W_{SI}} &= \dot{P_W}\Phi_{W_{SI-1}} + \left(1-\dot{U_W}\right)\Phi_{W_{SI}}
\end{aligned} \tag{B.3}$$

where Φ_W is the probability distribution and SI is the length of the discrete markov system chain. Φ_{W_1} can then be written as

$$\Phi_{W_1} = \xi_1 \Phi_{W_0}, \quad \text{where } \xi_1 = \frac{P_W}{\dot{U_W}} \tag{B.4}$$

and Φ_{W_2} can be expressed as

$$\begin{aligned} \dot{U_W}\Phi_{W_2} &= \left(\dot{P_W}+\dot{U_W}\right)\Phi_{W_1}-P_W\Phi_{W_0} \\ \dot{U_W}\Phi_{W_2} &= \left(\frac{P_W}{\dot{U_W}}\right)\left(\dot{P_W}+\dot{U_W}\right)\Phi_{W_0}-P_W\Phi_{W_0} \\ \Phi_{W_2} &= \xi_2\Phi_{W_0}, \quad \text{where } \xi_2 = \left(\frac{P_W^2(1-U_W)}{\left((1-P_W)U_W\right)^2}\right). \end{aligned} \tag{B.5}$$

Similarly, Φ_{W_3} can be solved as

$$\begin{aligned} \dot{U_W}\Phi_{W_3} &= \left(\dot{P_W}+\dot{U_W}\right)\Phi_{W_2}-\dot{P_W}\Phi_1 \\ \dot{U_W}\Phi_{W_3} &= \left(\frac{P_W^2(1-U_W)}{\left((1-P_W)\right)^2}\right)\left(P_W+U_W\right)-\frac{P_W\dot{P_W}}{\dot{U_W}}\Phi_{W_0} \\ \Phi_{W_3} &= \xi_3\Phi_{W_0}, \quad \text{where } \xi_3 = \left(\frac{P_W^3(1-U_W)^2}{\left((1-P_W)U_W\right)^3}\right). \end{aligned} \tag{B.6}$$

Continuing in this manner, the general form for Φ_{W_i} is

$$\Phi_{W_i} = \xi_i\,\Phi_{W_0} \qquad i=1,2,3,\ldots,SI \tag{B.7}$$

where

$$\xi_i = \left(\frac{P_W^i(1-U_W)^{i-1}}{\left((1-P_W)U_W\right)^i}\right).$$

The solution of the probability distribution can then be found by summing the distribution and solving for Φ_{W_0}

$$\sum_{j=0}^{SI} \Phi_{W_j} = 1 \tag{B.8}$$

$$\Phi_{W_0}\left(1+\sum_{j=1}^{SI}\xi_j\right) = 1 .$$

Φ_{W_0} can then be expressed as

$$\Phi_{W_0} = \frac{1}{1+\sum_{j=1}^{SI}\xi_j} . \tag{B.9}$$

A closed form expression for the summation can be found by noticing that

$$\begin{aligned} 1+\sum_{j=1}^{SI}\xi_j &= 1+X^1K^0+X^2K^1+X^3K^2+\ldots+X^{SI}K^{SI-1} \\ -\; X^1K^1\left(1+\sum_{j=1}^{SI}\xi_j\right) &= \quad X^1K^1+X^2K^1+X^3K^2+\ldots.+X^{SI+1}K^{SI} \\ \hline \left(1-X^1K^1\right)\left(1+\sum_{j=1}^{SI}\xi_j\right) &= 1+X^1K^0-X^1K^1-X^{SI+1}K^{SI} \end{aligned} \tag{B.10}$$

where

$$X = \frac{P_W}{\left(1-P_W\right)U_W} \tag{B.11}$$

and

$$K=\left(1-U_W\right). \tag{B.12}$$

The final closed form expression is then

$$1+\sum_{J=1}^{SI}\xi_j = \frac{1+X^1K^0-X^1K^1-X^{SI+1}K^{SI}}{\left(1-X^1K^1\right)}. \qquad \text{(B.13)}$$

By substituting eq. (B.13) back into eq. (B.9), the value of Φ_{W_0} can be found, which can then be used to solve to entire distribution using equations (B.4)-(B.7). This recursive method is applicable to any chain which has only one backward transition, i.e., only one server.