Recovery of Cell Loss in ATM Networks Using Forward Error Correction Coding Techniques

Liren Zhang, B.E.(Hons.), M.Eng.
Department of Electrical and Electronic Engineering
The University of Adelaide

A thesis submitted in fulfillment of the requirement for the degree of Doctor of Philosophy.

5th June, 1992

Contents

SUMMARY

In asynchronous transfer mode (ATM) networks [1] based on fast packet switching techniques, cells may be lost due to transmission errors, switching errors, misrouting, and network congestion. Because of the very high speed of cell transmission in these networks, in addition to a need to support real time services, in this thesis, we shall investigate methods for recovery of this lost information using forward error control coding techniques.

Firstly, this thesis studies the statistical characteristics of cell loss due to network congestion using both analytical method and simulation. Attention is focussed on a designated traffic stream as it goes through the network along a virtual path. Both random and bursty traffic are considered.

Secondly, coding schemes for forward recovery of cell loss at the cell level are proposed. This is done by a two-dimensional code [8], in which error detection for a single cell is represented by the x-axis, while the cell error correction and lost cell recovery among a block of consecutive cells is represented by the y-axis. The coding schemes operate between network edges for the virtual path connections (VPCs) or between end-to-end source-destination pairs for the virtual channel connections (VCCs) [66]. The performance with coding, and the design of suitable coding schemes utilizing the statistics of cell loss, are investigated.

Finally, the strategies of forward cell loss control at the burst level are studied. A powerful approach of forward error recovery along a virtual path or a virtual channel based on a joint usage of threshold controlled selective discarding of cells, two-dimensional coding, and interleaving is described. The long-term performance is evaluated in terms of average cell loss probability, outage probability, and delay.

The results and analysis throughout this thesis will show that coding technique can be a useful supplement to the conventional use of congestion control techniques against cell loss in ATM networks. To devise the methods of counteracting the cell loss in ATM networks subject to high link utilization, the traditional congestion control techniques are enhanced with innovative two-dimensional coding techniques. This extends the range of loads and conditions over which the ATM can be safely exploited. Since the CCITT has not decided

yet what type of coding schemes will be used in ATM networks for loss recovery, the consideration of the two-dimensional coding schemes is timely and important.

STATEMENT OF ORIGINALITY

This thesis contains no material which has been accepted for the award of any other degree or diploma and, to the best of the author's knowledge and belief, contains no material previously published or written by another person, except where due reference is made in the text.

Liren Zhang

June 1992

E: ..... COURSE: Ph.D

onstent to this copy of my thesis, when deposited in the University Library, being available f
otocopying.

ED: ..... DATE: 14/7/92

ACKNOWLEDGEMENTS

The author wishes to express his deepest appreciation to his academic supervisors Dr. K.W. Sarkies and Professor M.J. Miller for their continuous guidance and encouragement through this research. A special expression of gratitude is to the staff of Telecom Australian Research Laboratories for helpful discussions and releasing some important CCITT documents. The author wishes to thank his friends in the Department of Electrical and Electronic Engineering, the University of Adelaide.

The financial support of a postgraduate research scholarship from the University of Adelaide is acknowledged with gratitude, without that this research could not have materialized. The assistance of Australian Telecommunication and Electronic Research Board is gratefully acknowledged.

Finally, the author wishes to express his heartfelt gratitude to his wife, Xiaohua and his 12 year old daughter, Haiyan for their company during his 48 months stay in Australia. The deepest heartfelt gratitude is also to his parents for their visit and company with their son in Australia for 16 months. A heartfelt gratitude is also to his sisters for their support through writing letters from China and U.S.A.

Chapter 1

INTRODUCTION

1.1 Broadband ISDN and asynchronous transfer mode (ATM)

The current telecommunications environment is influenced by rapid technological change. The range of voice, video, data and image services is expanding with a growth in demand for services such as video conference, picture videotex, and file transfer. One of the major problems facing telecommunications service providers is that demand for the ever-increasing range of services is difficult to predict.

Since about 1985, worldwide activities to evolve the Integrated Services Digital Network (ISDN) into an optical-fibre-based universal broadband network have resulted in a new approach to integration and switching. In initial baseline documents, such as CCITT Recommendation I.121 [1], the asynchronous transfer mode (ATM) has been established as a key element of a new packet-switched broadband ISDN (B-ISDN). With the introduction of B-ISDN, the telecommunication industry will be able to bring digital connectivity out to the end users and allow the true integration of communication services. This development will bring significant economic benefits to industries and service providers who are dependent on the efficient collection, dissemination, and digestion of information. B-ISDN can bring an exciting variety of new services to residential users as well as opening new markets for the telecommunication companies.

The B-ISDN is conceived as an all-purpose digital network [1]. It will provide an integrated

access that will support a wide variety of applications for its customers in a flexible and cost-effective manner. The network capabilities will include support for:

- **Interactive and distributive services:**
  The network serves as a common carrier of both interactive and distributive communications, which may include voice, video, image and data.

- **Broadband rate:**
  Switched digital streams may operate at rates of up to 600 Mbit/s, with the possibility of raising this limit in the future.

- **Burst and continuous traffic:**
  Some services such as voice and video may need guaranteed bandwidth to meet performance requirements (low information loss and delay). On the other hand, burst data or image services may be provided more cost-effectively with a lower grade of service, using resources that can be shared statistically.

- **Connection-oriented and connectionless services:**
  Some applications, including most continuous-bit-stream communications, are best served by connection-oriented services. The connection-oriented services are characterized by a separation of the procedures for establishment and the end-to-end transfer of user information. Connection establishment procedures, which must precede information transfer, determine a route and set up a path between users. Other services, including mail and other data-oriented communications, may not warrant separate connection establishment and information transfer phases. These connectionless services provide routing and user information carried in the same message.

- **Digital signal processing:**
  Some services require transparent transport of digital bit streams through the network; others may take advantage of code compression capabilities used to conserve internal network bandwidth or code conversion capabilities provided to overcome terminal incompatibilities.

- **Point-to-point and complex communications:**
  Some services require a single point-to-point connection, either undirectional or bidi-

rectional, between two end points; other "multimedia" applications benefit from parallel connections between end points, connections among multiple users, and even combinations of multiple-connection, multi-user calls. Some multi-user services may involve digital signal processing.

The CCITT Study Group XVIII, the current focus for international standards on B-ISDN, generally calls the switching and multiplexing aspects the "transfer modes". The transfer mode chosen by the CCITT Study Group XVIII Task Group on ISDN Broadband Aspects as the basis of B-ISDN is called the Asynchronous Transfer Mode (ATM) [1] [22].

ATM represents a high bandwidth, low delay, packet switching and multiplexing technique. In ATM, time is divided into equal length time intervals or "slots" called cells. Each slot may carry an information cell, a signalling cell, or an idle cell. The information cells are identified and switched on the basis of a label in the header, as is illustrated in Figure 1.1. The CCITT has recommended as the standard for ATM cell structure [1] [66], that each ATM cell contain an information field of 48 bytes and a cell header of 5 bytes. Each header contains a virtual channel identifier (VCI) and virtual path identifier (VPI), payload and priority indicators, and one octet for one-bit header error forward correction and for the self-delineation of cell boundaries.

Figure 1.1 ATM cell using virtual path technique.

The term "asynchronous" refers to the fact that cells allocated to the same connection may exhibit an irregular recurrence pattern, as cells are filled according to the actual demand. In this aspect, ATM is similar to the well-known traditional packet transfer mode [2]. In most other aspects, especially in regarding to transfer rates, they are significantly different. The ATM is distinguished from the existing STDM (Synchronous Time Division Multiplexing) by a combination of several special features.

Firstly, the ATM multiplexers and switches are less dependent on consideration of bit rates for particular services. ATM can flexibly support a wide variety of services with

different information transfer rates without the complications associated with terminating a multiplicity of positioned channels. Consequently, multiple overly rate-dependent circuit fabrics are unnecessary, thus enhancing network integration.

Secondly, the ATM is suitable for both bursty and continuous-bit-rate services. ATM, operating in a deterministic mode, can support real-time continuous-bit-rate services by circuit-mode emulation, provided that sufficient switching and transport bandwidth is guaranteed. Bandwidth is allocated as part of a connection establishment signalling procedure that precedes end-to-end information exchange. The information transfer mode of a deterministic service must meet minimal end-to-end performance requirements for delay and loss so that a receiver can accurately reconstitute a periodic signal with minimal time and amplitude distortion.

ATM is also able to operate in a statistical mode to concentrate bursty data. Bursty data traffic may be assigned a lower priority, so that it does not adversely affect deterministic services. A bursty data service does not require continuous allocation of its peak bandwidth. This allows other bursty data services to statistically share the resources.

1.2 Virtual channel and virtual path

At the present time, it is planned that ATM network routing will make use of the concept of a virtual path. A virtual path (VP) is an ATM based information transfer technique that allows groups of calls sharing a common path through the network to be identified and processed using state information contained in the label of every ATM cell. This route state information referred to as the virtual path identifier (VPI), is processed at each exchange in the VP. Individual calls within the group are identified at the VP end-points by a second piece of route state information, referred to as a virtual channel identifier (VCI), also contained in the cell header.

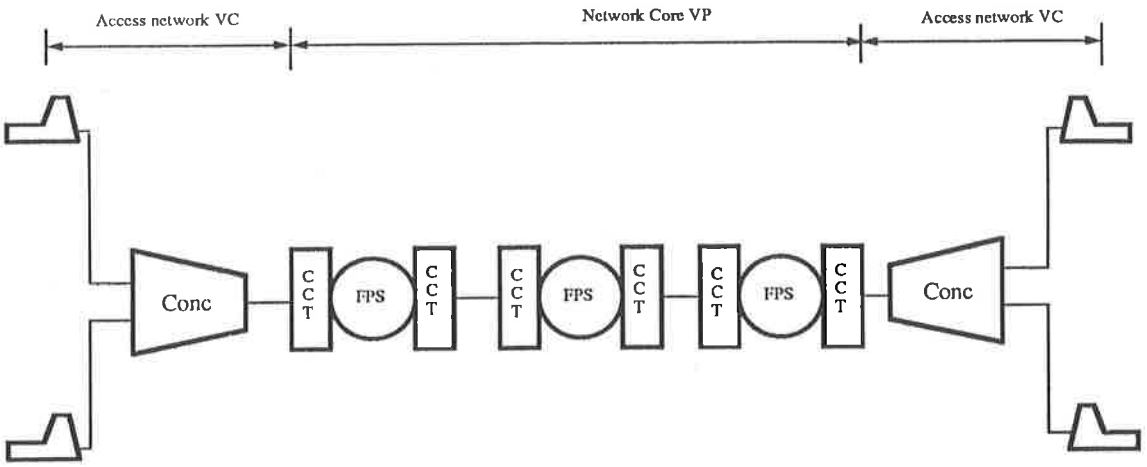


Figure 1.2 B-ISDN with virtual channel and virtual path.

A typical set of connections using virtual channel and virtual path techniques is illustrated in Figure 1.2. In these connections, the local access network with concentrators employs a virtual channel from the customer terminal equipment to the local exchange, with a mapping of the local channel number at each concentrator along the way. This can be done using a virtual channel identifier (VCI). Concentrators then perform a mapping of the VCI only. Since each VP would be already in existence at the time a subscriber is connected to the network, the concentrators need not execute any call handling functions. In the core network, the switch performs a mapping of the virtual path identifier (VPI) only. This reduces the amount of call processing required by the network. Therefore, an end-to-end connection would require communication using three types of virtual connections in the series. One VP is required in the core network, and one virtual channel is used at each end of the connection between a terminal equipment and concentrators in the access network. This is illustrated in more detail in Figure 1.3.

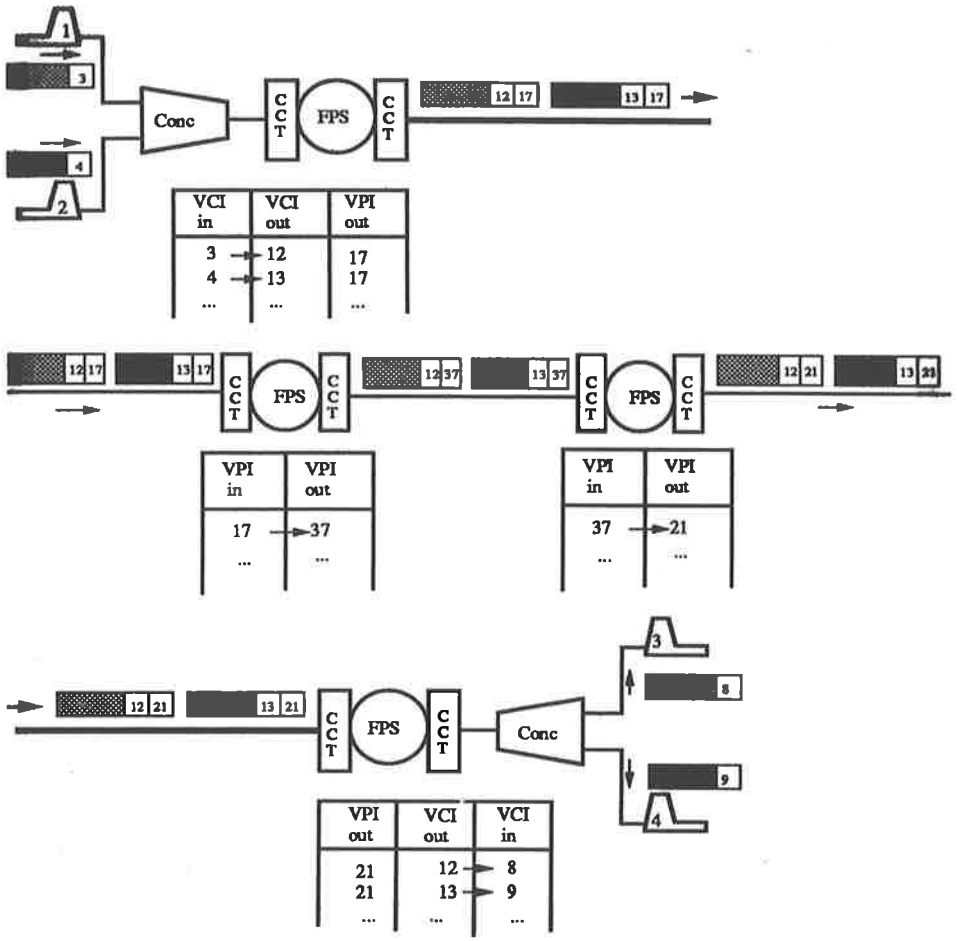


Figure 1.3 ATM network using virtual path technique.

In this illustration, VP routing is shown for calls between terminals 1 and 3 and terminals 2 and 4, respectively. The same virtual path is used between each pair of network switches. The first switch shown here is the concentrator (Conc) which forms part of the access network structure. The local access network with concentrators employs a virtual channel from the customer terminal equipment to the local exchange, with a mapping of the VCI at each concentrator along the way. The first call connect table (CCT) as the junction between VPs is used to separate the local access network and core network. In this case, the CCT is used to connect a number of individual calls to the network VPs and make an end-to-end connection. In this CCT, the virtual channel identifier (VCI) of an incoming cell is mapped to an outgoing VCI and an outgoing virtual path identifier (VPI), which is used to transit the exchanges to switch the group of virtual channels. There is a unique

mapping of VCIs for each call passing through the CCT (the call between 1 and 3 uses the VCI mapping 3 → 12, whilst the call from 2 to 4 uses the VCI mapping , 4 → 13), but all calls using the same VP are mapped to the same outgoing VPI (the 1-3 call and the 2-4 call both use the outgoing VPI 17). The cells of the 1-3 and 2-4 calls then enter a CCT at the next switch in the path along the VP. At this CCT, only the incoming VPI (17) is used to identify the switch output port and the outgoing VPI (37) for these cells. The VCIs identifying the individual calls (12 and 13) are not processed in any way along the VP. Further mappings of the VPI number occur at CCTs along the VP (37 → 21). When the cells of the 1-3 and 2-4 calls arrive at the CCT on the destination switch, the incoming VPI is stripped off. The incoming VCIs are mapped to the corresponding outgoing VCIs (12 → 8 and 13 → 9), and routed to the destination terminal equipment (3 and 4 respectively).

The identification and switching of traffic flows instead of individual calls can lead to advantages with regard to [3]:

- A reduced load on network control equipment.
- Lower call establishment delay.
- A flexible means of providing service protection to improve network availability.
- A flexible means of controlling network congestion. However the network efficiency is lower.

1.3 Congestion control in ATM networks

The fast packet switching mode of transmission in ATM networks allows great flexibility and opportunity for efficiently sharing of resources. However, it also introduces significant problems with regard to control and prevention of congestion leading to information loss [56][57][79].

Usually, traffic flows in the ATM networks are decomposed into 3 different levels [56][57]: i) call level; ii) burst level; iii) cell level. To guarantee a properly defined overall quality

of service (QOS) [56] to the users, the network has to control the traffic flow accordingly. Otherwise, congestion may happen in each of these three levels. An overload situation at the call connection level leads to call blocking whereas congestion at the lower levels leads to loss of cells. At the burst level, congestion occurs when the total arrival rate exceeds the capacity of the output link for sufficiently long that the available buffers become full. At the cell level, congestion is caused by simultaneous cell arrivals occurring in a time span equivalent to a source inter-cell time [56][72][78][79]. However, the congestion control in ATM is difficult because of the high link speed, diverse service requirements, and diverse characteristics of the traffic ATM is expected to support. Therefore, a whole set of control schemes will be applied whose actions normally work at each of these three levels [56][57][79].

Control at the call connection level

Upon arrival of a new call, the admission control process must allocate resources for the call at each switching node along a proposed route. During this process, a switching node must determine if it has enough resources to support the new call. This is done based on a set of traffic descriptors which represent traffic statistics given by existing calls. The switching node accepts the new call only if its performance requirements are met [67][79]. The sources are allowed to transmit within preset constraints. The determination of an appropriate minimum set of traffic descriptors is addressed in [68], but it still seems to require further discussions. Some problems with proposed the admission control schemes include the large amount of memory required, and possible excessive processing overhead [80].

Since ATM is packet-switched, the network may not explicitly reserve any bandwidth for calls. Instead, expected traffic is used as an indicator of the bandwidth required to support the new call to determine if the network can adequately support more calls. The problem is to find a computationally efficient algorithm which can decide how to combine traffic flows, and how to achieve rapid route selection. However, the statistics used for bandwidth allocation should allow the accurate calculation of how many different burst services can be multiplexed together. The difficulty of this is that many types of services such as bursty data or image transfer applications do not know even approximately what

their expected bandwidth will be [4][13]. On the other hand, the statistics need to be calculated in a very limited available time. It has been suggested in [5] that key traffic statistics should use the mean and variance of the traffic rate, but that a running variance would be too difficult to calculate. Currently it is not clear exactly which traffic statistics should be chosen for admission control apart from the peak and average rates.

Control at the burst level

Because of the high throughputs and low delay required in ATM network, reactive congestion control which detects congestion, and signals sources to reduce their traffic, is considered inappropriate [78]. High link rate increases the propagation delay-bandwidth product, i.e., a large amount of traffic can be in transit during the time required to send a choke cell to the sender to stop or slow the transmission when the reactive congestion control is activated. Large retransmission windows can cause severe buffer management problems, and therefore make many of the current control schemes ineffective. By contrast, preventative congestion control, which shapes the traffic from all sources so that congestion is minimized, is considered to be likely solution [54][67][68]. However, this solution cannot avoid some information loss without sacrificing the desired efficiency gains from statistical multiplexing.

The simplest approach would be to use only the total peak rate as the traffic statistic and to accept calls if the sum of the peak rates of all the calls is less than the capacity of the transmission link [54][67][68]. If the number of sources transmitting onto a particular link is such that the total peak rate of all sources combined is less than the capacity of the link, then information loss may be minimized, and the requested performance can be met without jeopardizing the performance of existing calls on the route. However, the efficiency gains of the ATM technique over simple broadband packet switching and multiplexing are then lost, especially for traffic which occurs in bursts separated by long intervals of silence. Therefore, it is expected that sources will be allowed into the transmission link even though the total peak rate of all sources may exceed the link capacity [57]. In this case, cell loss may be caused by buffer overflow. The cell loss will arise mainly from combinations of bursts of cells which exceed the capacity of the link. This will result in the loss of cells over a short time period. As link utilization increases, short time periods

of buffer overflow will occur frequently and cause significant disruption of all services using the link at that time.

At the burst level, control action attempts to ensure that cell loss due to buffer overflow is constrained. A source policing control monitors the user's traffic flow of each connection with respect to its negotiated traffic parameters. If a violation is detected, an appropriate action has to be taken: immediate dropping or marking of excessive cells to become candidates for loss in case of congestion. Leaky bucket average rate limiting [58] is one of the techniques currently being considered for this purpose. The leaky bucket consists of a cell counter which counts up at a fixed rate to a given maximum value, and counts down for every cell transmitted. When it reaches zero, further cell transmissions are inhibited. The finite size of the counter can be seen as the maximum allowable burst length. Some have argued that marking is ineffective and may degrade performance for unmarked cells [80]. Another disadvantage of marking is that it has no correlation with user level data priority. The user has better understanding of which cells are more important in delivery, whereas the marking system determines priority regardless of the user's needs. The output rate of a leaky bucket is selected according to the network's average admission rate which is related to resource allocation. This results in some difficulties with the policing of the peak rate and average rate at the same time. Another technique currently considered for the congestion control at burst level is sliding or jumping window flow control [13], which operate on cell counts within time windows. This control mechanism may not be suited to real-time services due to variable delay and loss [79]. An auto-repeat-request (ARQ) system is also required to support the window control mechanism.

Control at the cell level

At the cell level, the control deals with short term cell loss.

The common technique for dealing with short term cell loss is to provide buffers of sufficient size in the multiplexers and switches. These buffers will absorb sudden simultaneous cell arrivals, and store cells for transmission when link is free. As the total rate of traffic arrival increases toward the link capacity, the buffer may occasionally overflow. This situation may be overcome by increasing the size of the buffer. However, larger buffers result in an increase in both cell delay and the variation in cell delay. This can cause

difficulties for real time services which require synchronization between the transmitter and receiver and operate under strict time constraints. It should be noted that a first-in-first-out (FIFO) queueing model is commonly used to analyze the performance at cell level [57][75].

Cell discard priority has been suggested to separate services with different delay and loss requirements [59][63]. Two discard priority levels are recommended by CCITT [22]. In case of buffer overflow, cells with low-priority will be dropped first. Typical values of cell loss rate which have been recommended by CCITT are 10^{-10} for high priority and 10^{-6} for low priority [59][63], but these have been used for switching nodes only. Priority mechanisms can guarantee the quality for the services with high priority, but must pay a high price of loss and delay for the low priority services. However, the use of the discard priority mechanism to guarantee a certain level of overall service quality for any end-to-end service needs further study.

The mechanisms of congestion control operating at various levels are certainly helpful for avoiding cell loss. However, an ATM network is intended to be flexible enough to efficiently support all current and foreseeable services such as voice, data, video, and image. Thus an ATM network is required to handle different types of traffic such as fixed rate traffic and highly variable rate traffic. The latter may consists of traffic for which the bit rate cannot be controlled during a call, or peak rate cannot be predicted before call setup. In fact, the control problem has not been completely solved yet, because of the high link speed, diverse service requirements, and diverse characteristics of the traffic that ATM is expected to support at the same time [56].

1.4 Error control

Error control in an ATM network involves two different categories: i) detect and correct the transmission errors contained in the cell header or cell information field, ii) detect and recover the lost cells due to misrouting, switching error, priority discarding, and network congestion at the cell level or burst level.

Since ATM networks process cells by their header, bit errors in the cell header may cause

switching error and misrouting, and result in cell loss. However, referring to CCITT standard of ATM layer [66], cell header of 5 octets contains 8 redundant bits for one-bit header error forward correction using a CRC-8 [38] code with a minimum Hamming distance of three. Therefore, the cell loss due to uncorrectable header error in an random bit error environment can be greatly reduced. For a optical fiber with the bit error rate of 10^{-9}, the probability of cell loss due to random bit errors in the cell header is less than 4.0×10^{-17}, and is negligible. Burst errors, on the other hand, can be a major cause of cell delineation failure. Though the probability of cell delineation failure is small, once it happens, consecutive cells are then lost. According to the CCITT standard [66], cell delineation procedure recognizes delineation failure after 7 successive lost cells, and it takes 6 cells to recover the cell delineation. Therefore, at least 13 cells are consecutively lost when a cell delineation occurs. We will demonstrate that, in the latter chapters, this burst cell loss caused by cell delineation failure can be recovered using a two-dimensional coding scheme [8].

In order to protect the whole information field against the transmission bit errors, and to detect the lost cells due to buffer overflow, misrouting, and switching errors, the CCITT has recommended setting up an ATM adaptation layer (AAL) in the payload field [66] which contains two sublayers: convergence sublayer (CS) and segmentation and reassembly sublayer (SAR). All ATM cells are divided into 4 different types: AAL type 1 for fixed bit rate services, AAL type 2 for variable bit rate service, and AAL type 3 and 4 for data services. In the SAR sublayer of 4 octets of each ALL types 1 to 4, a cell sequence number of 4 bits is set up for cell loss detection [66]. In the AAL type 1 and 2, first bit to fourth bit of the information field form a cell sequence number. In the AAL type 3 and 4, third bit to sixth bit from a cell sequence number. In the AAL, 10 bits in the SAR sublayer are reserved for error protection in the information field. However, the CCITT has not decided yet what type of error protection coding schemes should be used. Therefore, the considerations of forward error recovery coding scheme is timely and important.

The ATM network is intended to be flexible enough to efficiently support all current and foreseeable services such as voice, data, images, and video. There are several conflicting transmission requirements for these services. Uncompressed voice services are tolerant

of occasional errors [45]. The inherent redundancy characteristics of voice allow small amounts of lost or corrupted information to be reconstructed or even omitted, without a severe degradation in voice quality. However, burst cell loss due to network congestion or buffer overflow may seriously affect the service quality.

Coded video involves bit rates from 2 Mbits/s (e.g., videophone) to 140 Mbits/s (e.g., HDTV) [46][47][48]. Because of the high bit rates involved, compared to 64 kbits/s telephony, cell loss occurs more frequently for video services. For example, for a cell loss probability of 10^{-8} and a cell length of 53 octets [1], the average period between two lost cells for a HDTV video service encoded at 140 Mbit/s is 5 minutes, while for 64 kbit/s telephony, under the same conditions, this average period is more than 8 days. For a video service, the primary frame has a duration of 40 ms, and thus will be transmitted over multiple cells. If a cell is lost in the middle of a primary frame, the phase alignment will be lost for the rest of the frame. Depending on the synchronization scheme used, a cell loss may corrupt a large part of a frame. For the DPCM video coding algorithm [48], cell loss will indeed result in loss of the rest of the video frame if no extra countermeasures are taken. For a variable length coding (VLC) algorithm [47], if a cell is lost, synchronization of the VLC code is lost, and the consecutive cells are useless until the VLC codes are reset. For the algorithm used in [48], the VLC codes are reset every 8-lines. Therefore, if a cell is lost, up to 8 consecutive lines may be affected. If such errors are visibly significant, it will be necessary to reset the VLC codes more frequently, say every line. This will limit the propagation of errors, but will increase the associated overhead.

Data applications demand essentially error-free transmission, and existing protocol standards, such as CCITT recommendation X.25 [2] reflect these requirements. Auto-repeat-request (ARQ) techniques such as go-back-N ARQ [32] is often used in data networks. This can provide reliable transmission, but it also introduces extra traffic into the network due to retransmission, and additional delay and variation in delay [43][44]. Compared with forward error control techniques, an ARQ system also requires: i) a return path for transmission of positive acknowledgement (ACK) and negative acknowledgement (NAK) signals; ii) buffers to store the transmitted data packets. Miller [7] discussed the possibility of using selective repeat ARQ or Hybrid I ARQ schemes for both data and packet voice

transmission in the packet-switched networks under the condition of without congestion. Compared to a go-back-N ARQ scheme, the selective repeat ARQ or Hybrid I ARQ has higher throughput and less amount of extra traffic. However, a pair of larger buffers are required to be set up at the both transmitter end and receiver end, respectively. Thus in an ATM network with very high transmission and switch throughput, an ARQ technique may require an enormous buffer.

On the other hand, both voice and video are real-time services and sensitive to transmission delay. The voice services require the delay to be small and uniform. Otherwise, the received signal will be distorted. Interactive video services also require delay to be uniform. Therefore, the ARQ technique may not be suitable for voice and video services as it may introduce variable delay. We will discuss the details in chapter 4.

The author has proposed a concept of forward error recovery [8] in 1989 for the purpose: i) detect and correct the transmission errors contained in cell information field, ii) detect and recover the cell loss due to misrouting, switching error, and network congestion at the cell level. This is done by a two-dimensional code, in which error detection for a single cell is represented by the x-axis, while the cell error correction and lost cell recovery among a block of consecutive cells is represented by the y-axis. In this two-dimensional coding scheme, the x-axis parity redundant bits used for error detection and the cell sequence number used for cell loss detection are incorporated in to the SAR sublayer located in the AAL in the payload field. The details will be described in the later chapters. We should note that the cell losses caused by buffer overflow, switching errors, or misrouting, and the cell errors caused by transmission errors may appear with different statistics. However, they can share the same error recovery schemes. In this scheme, our attention is focussed on using forward error recovery coding techniques against the cell loss caused by network congestion at the cell level rather than using large size queues. The proposed scheme operates between the network edges at the virtual path connection level or end-to-end source-destination pairs at the virtual channel connection level [66] rather than at every switching node along the route. It therefore allows greater control over cell delay, as well as providing an extremely powerful mechanism for reducing information loss.

Ohta et al. recently presented a forward loss recovery coding scheme [84] for data transfer

application in an ATM network. A block consists of Γ cells arranged in Θ rows. Each row of Γ cells contains one redundant cell for loss detection, and each column of Θ cells contains one redundant cells for loss recovery. Therefore, $\Gamma + \Theta - 1$ cells are overhead in each block, while a single burst loss of $\Gamma - 1$ cells can be recovered in each block. A pair of buffer of size $\Gamma \times \Theta$ is also required to be set up at the transmitter end and receiver end, respectively, to associate the encoding and decoding. By comparison with our more elegant two-dimensional coding scheme, we shall show in a later chapter that this scheme [84] has more overhead and less capacity for recovering from long burst loss.

1.5 Outline of thesis

The purpose of the study in this thesis is to examine the application of forward error recovery coding techniques to reduce cell loss probability. Our approach is to select an efficient two-dimensional code for loss recovery at the cell level. The cell sequence number needed for cell loss detection and the parity check bits generated by encoding the information field of a cell, are incorporated into ATM adaptation layer (AAL). The approaches presented in this thesis extend the range of network utilization and conditions over which an ATM network can be safely exploited, and can be a useful supplement to the currently proposed set of congestion control techniques.

In chapter 2, we study a statistical description of cell loss resulting from network congestion at the cell level. This is important to obtain the evaluation of performance with coding. A virtual path (VP) is modelled as a tandem queueing system consisting of a number of queues in series. Each queue along the path is considered as a bulk arrival queue with a finite buffer size. The term "bulk" is used to indicate that a random-size group of cells arrive at a switching node from different traffic streams within the same time interval. For cell arrival traffic, the mixtures of discrete-time constant rate deterministic arrival processes are considered.

Firstly, the performance of a virtual path is evaluated for a designated traffic stream as it goes through the queues along the tandem queueing network. In the analysis, an independence assumption between the adjacent queues is used [61][65]. The cell loss

probability and queueing delay are derived.

Secondly, two Markov models are used to estimate the higher order statistical distributions of bursty cell loss patterns, such as cell loss-free run distribution and cell loss pattern distribution. These results are subsequently used as a basis for evaluating the performance of traffic streams with forward error recovery coding schemes in chapters 4. The results also provide useful information needed for code design.

In Chapter 3, a simulation study of statistics of cell loss caused by congestion is undertaken. The simulation is arranged to represent a VP as a tandem queueing network. Each queue along the path is represented as a bulk arrival queue with finite buffer. In the simulation, an actual tandem queueing network is used to obtain the performance measures for the designated traffic. All software implementations are written using FORTRAN and run on a VAX system. The statistics of cell loss due to buffer overflow are obtained for both random and bursty traffic using a Monte Carlo method.

Firstly, the simulation results based on the mixtures of constant rate deterministic arrival processes are compared with the analytical approximations obtained in chapter 2. Although our analytical approach in chapter 2 is approximate, it is shown to be quite accurate by comparing it to the simulation results.

Secondly, since an exact queueing analysis for a bulk arrival queue with bursty cell arrivals is too complex due to queue transient effects, the statistics of cell loss with bursty cell arrival traffic are studied by simulation. The mixtures of homogeneous burst-silence processes [73][83] are used in the simulation as the bursty cell arrival traffic.

In chapter 4, the forward error protection scheme proposed for cell error correction and cell loss recovery at the cell level is investigated. This scheme employs a two-dimensional code, in which error detection for single cell is performed by the x-axis code, while lost or errored cell recovery among a block of consecutive cells is performed by the y-axis code. A class of high rate codes, referred to *small distance* codes, suitable for the x-axis code are investigated. The erasure correction coding is used for the y-axis code. The erasure correction decoding algorithm and its implementation using binary cyclic codes or non-binary Reed-Solomon codes is also investigated.

The performance with coding is evaluated in the presence of transmission errors and cell loss due to network congestion. The bandwidth efficiency, delay, cell loss/error probability, and the capability of recovering from long cell loss bursts are evaluated. Several specific illustrative schemes are considered using a *small-distance* code as the x-axis code and an efficient binary cyclic code or non-binary Reed-Solomon code as the y-axis code. We show that if the x-axis code and the y-axis code are chosen properly, high recovery rates can be obtained even under high link utilization conditions.

The statistics of burst cell loss obtained from chapters 2 and 3 may be used to assist in the design of coding schemes. The methods for using these statistics for code design, including choice of code word length, code rate, and queueing buffer size, are also discussed in chapter 4.

Finally, in chapter 5 we consider the strategies of forward loss control against the cell loss caused by congestion at the burst level. A powerful method of forward error recovery along a virtual path or virtual channel based on a joint usage of threshold controlled discarding of cells, two-dimensional coding, and interleaving is described and evaluated. This method allows one to decrease significantly the number of lost cells during periods of congestion at burst level due to traffic fluctuations during a call. It is shown that coding techniques can be a useful supplement to the conventional use of congestion control techniques.

Chapter 2

STATISTICS OF CELL LOSS

Summary

In this chapter, we study the statistical description of cell loss caused by congestion at the cell level [56][72][79]. This will be needed for the evaluation of performance with forward error recovery coding in the later chapters. We also define a number of terms and symbols which will be used in the later chapters.

A virtual path (VP) is modelled as a tandem queueing network consisting of a number of single server queues in series which operate on a first-in-first-out (FIFO) basis. The switches associated with the queues will split up incoming streams with virtual paths (VPs) going to different destinations, and combine streams with VPs going to the same outgoing link at the switch output, to enter the output queue [60][61][63]. Each queue along the VP is considered as a bulk arrival queue with finite buffer size. The term "bulk" is used to indicate that a random-size group of cells arrive at a queue from different traffic streams within the same time slot. In this chapter, cell arrival streams which come from the different VPs and join the queues are considered to be constant rate deterministic arrival processes. We shall consider bursty traffic by simulation in chapter 3, since an exact queueing analysis of an ATM multiplexer with bursty cell arrival traffic leads to very complex queueing models due to queue transient effects.

Firstly, we derive an expression of the equilibrium buffer occupancy distribution and probability of cell loss for a single server queue with finite buffer using a discrete time

Markov model [25][64][71]. The conditions of the steady-state queueing approximation used in the analysis are discussed.

Secondly, the cell loss probability is obtained for a designated traffic stream as it goes through the tandem queueing network. An independence assumption between the adjacent queues is used in the analysis. The cell loss probability is evaluated in terms of link utilization for a given set of the queueing network parameters.

Finally, we estimate the statistical distributions of burst cell loss. Two Markov models are used as the generative model for bursty cell loss. We have estimated some higher order statistical distributions of cell loss such as cell loss free-run distribution and cell loss pattern distribution. These results are subsequently used as a basis for evaluating the performance with error recovery coding in chapter 4.

Since our analytical approach is approximate, we will check the accuracy by comparing the analytical results to the results obtained from simulation in chapter 3. In the simulation, an actual tandem queueing network without the "independence assumption" will be used to obtain the performance measures for the designated traffic stream.

2.1 Introduction

Queueing arises naturally in a study of packet switched networks [14]. In an ATM network based on fast packet switching techniques, cells arriving either at an entry point to the network or at an intermediate switching node on the way to the destination, are buffered and processed to determine the appropriate outgoing transmission link connected to the next switching node along the path. The time spent in the buffer waiting for transmission, and the probability of cell loss due to buffer overflow, become the measures of network performance. The waiting time and loss rate depend on switch processing time and cell length. They also depend on the transmission link capacity, on the traffic statistics, on the buffer size, and on the service discipline being used to handle the cells. In this chapter, an idealized switch structure with output queues will be considered [60][61][63]. The details will be discussed in later sections.

Many papers [15 - 21] have studied the performance of a multiplexer with packetized voice, video, and data using queueing theory. Their aim is to properly understand the characteristics of packet-switched network with different types of traffic and to evaluate the statistics of loss or delay. Kuczura [17] proposed an interrupted Poisson process (IPP) model to assess data traffic performance for a single output queue with finite buffer. The probability of buffer overflow versus the buffer size was evaluated for a given set of parameters such as traffic load and statistics of the arrival traffic. Heffes et al. [19] proposed a Markov modulated Poisson process (MMPP) model for the packetized data and voice traffic. The performance of a single MMPP/G/1 queue with infinite buffer size was analyzed. The average queueing delay versus the offered traffic load were obtained. Maglaris et al. [20] [21], using a discrete-state, continuous-time correlated Markov process model, modelled multiplexed video traffic as one which changes among different fixed-rate levels. The performance with a single M/G/1 queue was analyzed. However, most studies of this type relate to a multiplexer in the access network.

In an ATM network, congestion may be decomposed into 3 different levels [56]. An overload situation at the connection level leads to call blocking whereas congestion at the lower levels leads to losses of cells. In connection with recovery of cell loss, only the lower level congestion must be taken into consideration. It is appropriate to consider cell loss phenomena which occur at burst and cell levels. In particular, the cell arrival process exhibits different types of correlation at both levels [56][57][72][79]:

- at the burst level, congestion occurs when the total arrival rate exceeds the capacity of the output link for averaged over a period greater than an inter-cell time [72][79]. The instantaneous cell arrival rate varies relative slowly with burst arrivals and departures, tending to produce a positive correlation between the numbers of cell arrivals in successive periods;
- at the cell level, congestion is caused by simultaneous cell arrivals occurring in a time span equivalent to a source inter-cell time. For a given burst composition, interarrival times are negatively correlated due to the periodic emission of cells within each VP.

Over the past few years, many studies [69 - 73] [81 - 83] have been focussed on the loss of an

ATM multiplexer with mixed traffic sources. Sriram and Whitt [73] considered an ATM multiplexer. They indicated that if a large number of independent sources are connected to an ATM multiplexer, and each source contributes a small fraction to the total load, then within the inter-cell time of a reference source, the arrivals from the other sources are locally Poisson distributed, and short-term behavior of a statistical multiplexer can be characterized using this simple model [73].

For bursty arrival traffic, more complex models based on the MMPP, such as mixtures of burst-silence sources have been used to study an ATM multiplexer [64][72][75][82]. The studies have shown that for different levels, the loss in an ATM queue has the following properties [83]:

1. for long queues and long burst lengths, loss behavior is determined by the average arrival rate into the queue, exceeding the output rate of the queue. When this happens, albeit infrequently in a well designed network, the loss rate will be temporarily large.

2. for long queues and short burst lengths, transient queue effects must be taken into account.

3. for short queues and long burst lengths, loss behavior is determined by the fluctuations within a frame period. This is readily taken into account by the assumption that cells from different streams are uniformly distributed over the frame period. In this condition, the transient queue effects can be ignored.

Norros et al. [72], Eckberg et al. [75], and Roberts et al. [64] have considered the performances of an ATM multiplexer with mixtures of homogeneous burst-silence traffic under the condition 3, i.e., assuming that when the burst lengths are much greater than the buffer size, the transient effects can be ignored. Kröner et al. [82] have compared a number of approximations for cell loss at both burst and cell level. However, an exact queueing analysis of an ATM multiplexer with cell arrivals in short bursts leads to very complex queueing models due to the transient effects involved, and seems not tractable [59]. Therefore, the performance evaluation of the statistics of cell loss at the cell level for an ATM multiplexer with cell arrivals in short bursts has to be done by simulation or

approximation.

In this chapter, we shall study the statistics of cell loss due to congestion at the cell level using mixtures of constant rate deterministic arrival processes. The statistics of cell loss for bursty arrivals will be studied in chapter 3 using simulation. Our attention is focussed on a designated traffic stream as it goes through an ATM network along a VP with small size queues under heavy traffic conditions. This is the region of interested to us for the forward error recovery coding techniques which will be discussed in chapters 4 and 5. For the traffic streams carried on each VP, our approach is to use a discrete-time model rather than continuous time, reflecting more closely the slotted nature of fixed length cell based switching techniques in ATM networks. We assume that the traffic carried on each VP is approximated by a constant rate deterministic arrival process [59]. The justification of our defined models will be discussed in detail in the later sections.

2.2 Modelling of a virtual path

In an ATM network using virtual path (VP) techniques, the switches will be required to operate at very high throughput (possibly from 150 Mbit/s to 600 Mbit/s per input [1]) and they may be interconnected by several different input and output optical fiber links. Each link consists of a number of VPs. We assume that cells carried on a VP all have the same priority level.

We shall model a switching node as an ideal nonblocking switch with cell buffering at the output ports [60][61][63]. The switch is considered to be attached to a number of input and output links, each of which carries a number of VPs. A typical switch structure is illustrated in Figure 2.1.

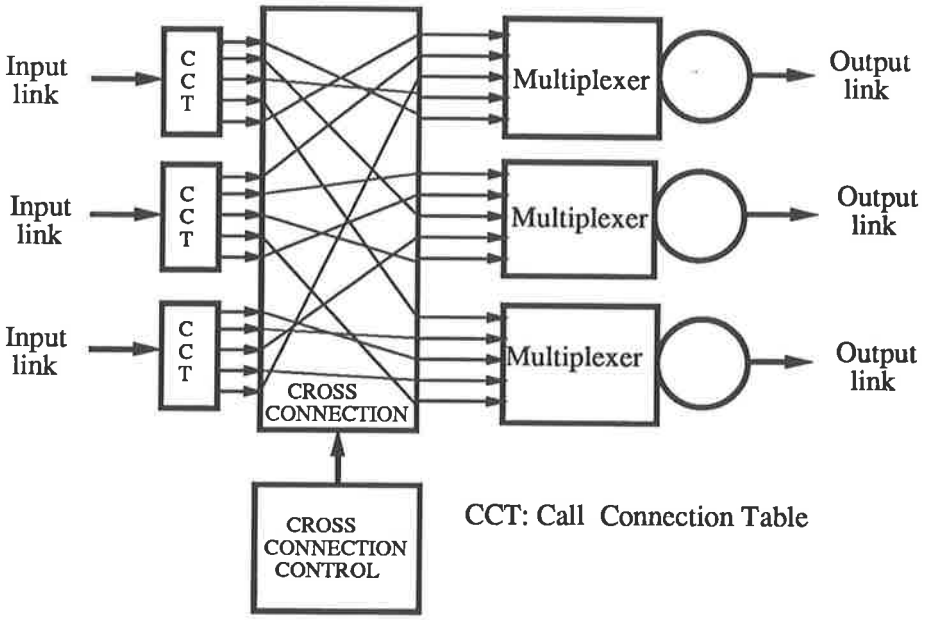


Figure 2.1 A diagram of switch node.

In this figure, the call connect table (CCT) routes the cells into different paths in the cross connection by processing the virtual path identification (VPI). Each VP has its own individual path through the cross connection. This path can be set up by the VP controller according to the requests for transmission establishment.

Each multiplexer is modelled as a single server output queue operating on a first-in-first-out (FIFO) basis. In the transmission link, channel time is slotted with slot size being equal to a cell transmission time. Cells are assumed to occur at slot boundaries. One cell is served per slot, if any exists in the queue. We also assume that a cell arriving at the beginning of the slot is immediately served if it finds the queue empty. The number of cells waiting in the queue is called *queueing length*. Bulk cell arrivals from different traffic streams result in significant variations of the output queueing length, which may also result in buffer overflow. As mentioned in the introduction to this chapter, the overflow could happen in two levels: burst level and cell level. In this chapter, our attention is focussed on the loss at the cell level.

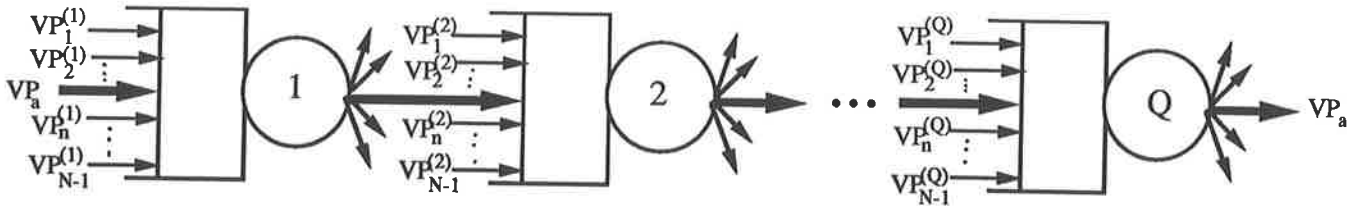


Figure 2.2 A virtual path is modeled as a tandem queueing network.

As shown in Figure 2.2, a particular virtual path through the network may now be modelled as a tandem queueing network. In the following sections, we will consider the performance of a particular virtual path VP_a through an ATM network using the following descriptions.

1. VP_a is modelled by Q single server queues in series. These queues operate on a first-in-first-out (FIFO) basis.

2. A queue I (I = 1, 2, ..., Q) along the path VP_a is modeled as a bulk arrival queue with a finite size K.

3. The queue I has N ($N >> K$) independent input VPs, including VP_a. We let $VP_n^{(I)}$ ($n = 1, 2, .., N-1$) denote one of the $N-1$ input VPs, excluding VP_a, which join the queue I. This is shown in Figure 2.3.

4. The switching functions result in splitting of traffic streams with VPs going to different destinations and mixing of traffic streams going to the same output links to enter the output queues [61][63].

5. A traffic stream, denoted as A-stream, is designated to be carried on the virtual path VP_a through the tandem queueing network, while the other virtual paths $VP_n^{(I)}$ carry traffic streams, denoted collectively as the D-stream.

6. The designated A-stream carried on VP_a departing the current queue goes to the next queue while the D-streams carried on other VPs are assumed to leave the tandem queueing network immediately.

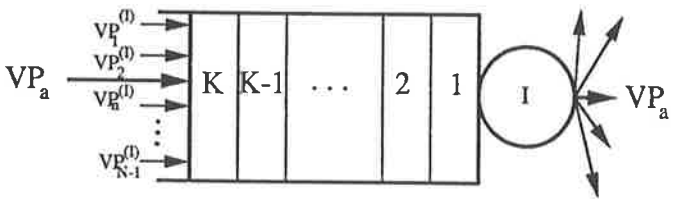


Figure 2.3 A bulk arrival queue with the finite buffer.

2.3 Traffic models

In this chapter, each traffic stream carried on each VP including the A-stream and the D-streams, is considered to be a discrete-time constant rate deterministic arrival process [25]. This will be justified in the next section. We describe this traffic model as follows.

1. We use the numbers T, $2T$, $3T$, ..., to designate points in time associated with the beginning of time slots.
2. The time interval $[kT, (k+1)T)$ represents the kth time slot, which includes the time instant $t = kT$ but excludes the time instant $t = (k+1)T$. T is the length of a time slot unit. The time instant $t = (k+1)T$ belongs to time interval $[(k+1)T, (k+2))$, i.e. $(k+1)$th time slot.
3. No more than one cell may occur within a time slot in a single traffic stream.
4. The arrival of a cell in a given time slot is statistically independent of the arrival of cells within any previous time slots.

5. The probability that a cell comes from the designated A-stream and arrive at the first queue of the tandem queueing network is $p_a^{(1)}$ per slot, where subscript a represents the designated A-stream, and superscript 1 represents the traffic stream joining the first queue ($I = 1$). The probability that a cell comes from the D-stream is p per slot.

Note that the conditions 3, 4, and 5 means that for a single D-stream, the interarrival time between two successive arrivals is geometrically distributed [25], that is,

$$P_{rob}\{\text{interarrival time} = \Omega \text{ slots}\} = (1-p)^{\Omega-1}p, \tag{2.1}$$

with a mean value of $1/p$. Likewise, for the A-stream, we have

$$P_{rob}\{\text{interarrival time} = \Omega \text{ slots}\} = \left(1 - p_1^{(1)}\right)^{\Omega-1} p_a^{(1)}, \tag{2.2}$$

with a mean value of $1/p_a^{(1)}$.

Since our attention is focussed on the designated A-stream stream going through a tandem queueing network, the traffic characteristics of the designated stream may be changed as it passes through several queues. Here, we assume that the interdeparture process of the designated A-stream which departs the queue I ($1 \leq I \leq Q$) and goes to join the queue $I+1$, is also a constant rate deterministic process, but with a traffic rate of

$$p_a^{(I+1)} = (1 - P_{ls}^{(I)})p_a^{(I)}, \tag{2.3}$$

where $p_a^{(I)}$ is the cell rate of the designated A-stream arriving at the queue I, and $P_{ls}^{(I)}$ is the cell loss probability for A-stream at the queue I. Since here we have introduced an independence assumption between the adjacent queues in the tandem queueing network, we will check its accuracy by comparing it to simulation results in chapter 3. In the simulation, an actual tandem queueing network without the independence assumption is used to obtain the performance measures for the designated traffic stream as it goes through the tandem queueing network.

Since N independent traffic streams join the queue I, then more than one cell may arrive at the queue I in the same time slot. Let $g^{(i)}$ represent the probability that i cells arrive

at the queue I in a slot, $g^{(i)}$ can be computed from a binomial distribution:

$$g^{(i)} = \begin{cases} (1 - p_a^{(I)})(1-p)^{N-1} & \text{for } i = 0 \\ g_{a1}^{(i)} + g_{a2}^{(i)} & \text{for } 1 \leq i \leq N \end{cases} \tag{2.4}$$

where

$$g_{a1}^{(i)} = p_a^{(I)} \begin{pmatrix} N-1 \\ i-1 \end{pmatrix} p^{i-1}(1-p)^{N-i} \tag{2.5}$$

represents probability that the i cell arrivals do contain a cell from A-stream, and

$$g_{a2}^{(i)} = (1 - p_a^{(I)}) \begin{pmatrix} N-1 \\ i \end{pmatrix} p^{i}(1-p)^{N-i-1} \tag{2.6}$$

represents probability that the i cell arrivals do not contain a cell from A-stream. The average rate of overall cell arrivals is given by

$$\eta = \sum_{i=0}^{N} i g^{(i)} \tag{2.7}$$

and the peak rate of overall cell arrivals is given by

$$\eta_p = p_a^{(1)} + (N-1)p. \tag{2.8}$$

2.4 Justification of defined models

As shown in Figure 2.1, we use a discrete time first-in-first-out single server queue with bulk arrivals to represent an ATM outgoing link buffer. A similar model was used in [61] [63]. We do not model the internals of a switch as we only need to represent the splitting and mixing of the traffic streams in the switch. There are many implementations using such models where buffer overflow occurs only at the outputs [63][64][65]. In our analysis, a short queue is appropriate because our attention is focussed on effects of loss recovery schemes which will be discussed in chapter 4.

In this chapter, since we will analyze the statistics of cell loss due to congestion at the

cell level using mixtures of constant rate deterministic arrival processes, we will attempt to justify our choice of traffic models as follows.

Mixtures of constant rate deterministic arrival processes:

- If an ATM multiplexer is connected with a large number N of input streams, and each input stream only contributes a small fraction to the total traffic load, then the interarrival distribution between two consecutive cell arrivals which come from the same traffic stream is locally Poisson [59][73]. A Poisson arrival process will be accurate if the size of the queue is smaller than the inter-cell time of the highest rate stream. The correlated cells will rarely appear in the queue at the same time. For a bulk arrival queue with a short buffer of K slots and a large number N of input streams, the equivalent condition of $N >> K$ is possible to be satisfied. Since we are considering a discrete-time system, and for each individual stream the interarrival time between two successive arrivals is geometrically distributed as given by equations (2.1) and (2.2), then the short-term behavior of the mixed traffic stream can be characterized using a discrete-time Bernoulli arrival process which is a discrete-time form of the Poisson process [59].

- Likewise, for $N >> K$, the interdeparture distribution between two consecutive cells of the designated A-stream is also a locally discrete-time Bernoulli process. Note that here we have introduced an independence assumption between the adjacent queues in the tandem queueing system.

- In the network core links, cells coming from the same stream will become less correlated due to mixing of traffic in the queues. The gaps between consecutive cells of the designated A-stream will also increase as it passes through several queues. Hence, the traffic model may also suit some higher rate streams as well.

- When $N \leq K$ and the average rate of overall cell arrivals is high, the inter-cell time may be less than the queueing buffer size K, and cells from the same stream will often appear in the queue at the same time. In this case, the queue transient time effects must be taken into account, and analysis becomes very complex. In this chapter, we do not consider the situation of $N \leq K$. We will consider this in chapter 3 by means

of simulation.

Mixtures of bursty traffic with long burst length:

- A mixture of N bursty traffic streams with long burst length can be described as a succession of constant rate bursts [73]. In a fixed burst composition whose length is assumed to be much greater than buffer size K, the cell arrivals can also be considered locally as a mixture of N constant rate deterministic traffic streams. If $N >> K$, then the inter-cell time of a single stream is much greater than the buffer size, and the interarrival distribution between two consecutive cell arrivals which come from the same traffic stream is locally Poisson.
- Cells from the same stream will become less correlated due to mixing of traffic at the queue. In this case, the traffic model may also suit some higher rate streams as well.
- The condition of above assumptions are: i) the burst length must be much greater than the buffer size K, ii) $N >> K$. In the long run, the cumulated queue transient time will be smaller by comparison with the cumulated time of steady-state operation.

Therefore, for the purpose of analysis of short-term behavior for an ATM multiplexer with long burst arrivals, and where the peak rate of all arrivals is less than the capacity of the transmission link, the steady-state queueing model with mixtures of constant rate deterministic traffic can be used as an approximation. Note that when the peak rate of all arrivals is greater than the output link capacity, the loss belongs to the burst level [57].

When medium size or short bursts are involved, the situation is different. Both transient effects and cell level fluctuation must be taken into account. In this case, an exact queueing analysis leads to very complex queueing models, and seems not tractable. The performance measures of interest to us at the cell level cannot obtained exactly by analysis, and these measures have not been studied very much so far. We shall consider this using simulation method in chapter 3.

2.5 Performance of a virtual path

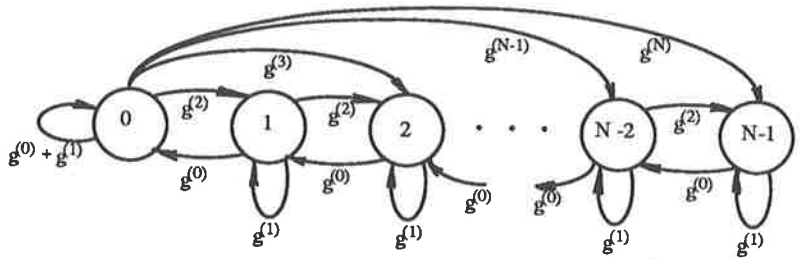
Now we consider the cell loss probability for a particular virtual path VP_a. As shown in Figure 2.2, a queue I ($I = 1, 2, ..., Q$) along the path VP_a can be described as a bulk arrival queue with a finite buffer of size K, and has one A-stream (the designated stream) and $N - 1$ D-streams to input the queue. Here we will first solve for the equilibrium buffer occupancy distribution and probability of cell loss for the queue I. Much of the following analysis of the bulk arrival queue involves well-known results for discrete-time queueing systems [23][25]. Note that the solution for the well known **M/D/1/K** queue using mixed Markov chain is very similar to the idealized case of $N \rightarrow \infty$.

Recall the definition of $g^{(i)}$ which gives the probability that i cells arrive at the queue I in the same time slot, as given by equation (2.4). Now we use a Markov chain to represent this bulk arrival queue. The Markov state transition probability diagram is illustrated in Figure 2.4. The Markov diagram contains a set of $K + 1$ states, labelled

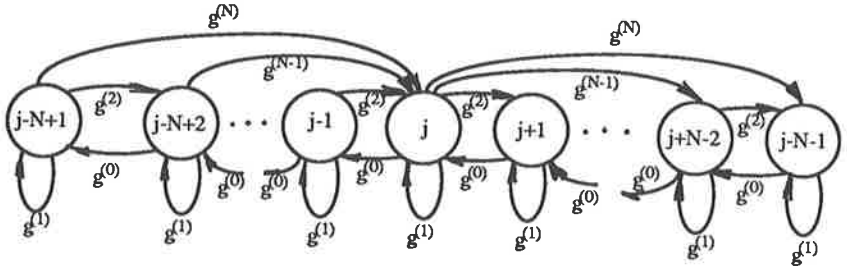
$$j = \{0, 1, 2, ..., K\},$$

where j represents the number of cells waiting in the queue. The transition in a given time slot will depend on the number of cells which arrive at the queue and whether a cell departs from the queue in that time slot.

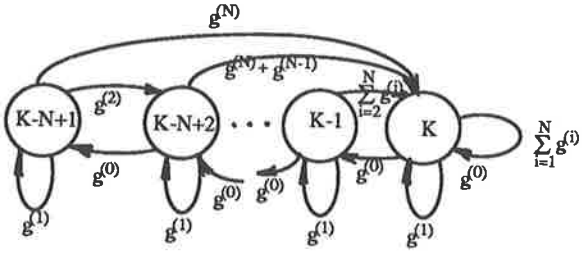
Figure 2.4 (a) shows the transition diagram of state $j = 0$. Likewise, Figure 2.4 (b) shows the transition diagram of state $0 < j < K$, and 2.4 (c) for state $j = K$, where K is the buffer size.



(a) j = 0



(b) 0 < j < K



(c) j = K

Figure 2.4 Markov state-transition-probability diagram for a bulk arrival queue with a finite buffer size of K.

Let $P(j)$ represent the probability of j cells waiting in the queue. Then we have

$$P(0) = (g^{(0)} + g^{(1)})P(0) + g^{(0)}P(1) \tag{2.9}$$

$$P(j) = [\sum_{i=0}^{N} g^{(i)}P(j+1-i)] \qquad 0 < j < K, \tag{2.10}$$

$$P(K) = \sum_{i=1}^{N}(\sum_{l=i}^{N} g^{(l)})P(K+1-i). \tag{2.11}$$

Note that, when $j < 0$ or $j > K$, $P(j) = 0$.

To eliminate $P(0)$ in equation (2.9), we define

$$x_j = \frac{P(j)}{P(0)} \qquad \text{for } j = 0, 1, 2, ..., K, \tag{2.12}$$

$$\beta = \sum_{j=0}^{K} x_j. \tag{2.13}$$

From equations (2.9)(2.10)(2.11), we can obtain x_j for $j = 0, 1, 2, ..., K$, that is

$$x_0 = 1,$$

$$x_1 = \frac{1 - (g^{(0)} + g^{(1)})}{g^{(0)}}, \tag{2.14}$$

$$x_j = [\sum_{i=0}^{N} g^{(i)} x_{j+1-i}] \qquad 0 < j < K, j + 1 \geq i, \tag{2.15}$$

$$x_K = \sum_{i=0}^{N} (\sum_{l=i}^{N} g^{(l)}) x_{K+1-i} \qquad K + 1 \geq i. \tag{2.16}$$

From equation (2.13), we can obtain

$$\begin{aligned} \beta &= \sum_{j=0}^{K} x_j \\ &= \frac{P(0) + P(1) + ..., + P(K)}{P(0)}. \end{aligned} \tag{2.17}$$

Since

$$\sum_{j=0}^{K} P(j) = 1,$$

then

$$\beta = \frac{1}{P(0)}.$$

Thus $P(0)$ is given by

$$P(0) = \frac{1}{\beta}, \tag{2.18}$$

and $P(j)$ is given by

$$P(j) = \frac{x_j}{\beta} \qquad j = 1, 2, ..., K. \tag{2.19}$$

The queueing length j is a random variable, and $P(j)$ gives its equilibrium distribution. The mean length of the queue is

$$\overline{W} = \sum_{j=0}^{K} jP(j) \tag{2.20}$$

Then the average queueing delay is given by

$$\overline{\tau} = \frac{\overline{W}T}{1 - g^{(0)}} \tag{2.21}$$

where $g^{(0)}$ can be given by equation (2.4).

Suppose the number of cells arriving at the queue in a given time slot is i $(1 \leq i \leq N)$, and j cells are waiting in the queue at the time instant of the beginning of that time slot. If $(j + i) > K$ and one cell is served within that time slot, then

$$l = j + i - 1 - K$$

cells will be lost. In this case, l could be equal to 1, 2, ..., $N - 1$. Thus, the probability that l cells are lost in a time slot can be given by

$$P_{lsq}(l) = \sum_{i=l+1}^{N-1} g^{(i)} P(K + l + 1 - i) \qquad l = 1, 2, ..., N - 1. \tag{2.22}$$

Overall, the cell loss probability for a single bulk arrival queue is given by

$$\begin{aligned} P_{lss} &= \frac{\text{total number of lost cells}}{\text{total number of cell arrivals}} \\ &= \frac{\sum_{l=1}^{N-1} P_{lsq}(l)}{(1 - g^{(0)})} \end{aligned} \tag{2.23}$$

Note that in equation (2.23), if $p_a^{(1)} = p$ (i.e., the N input streams all have the same arrival probability) and $N >> 1$, then the bulk arrival queue with finite buffer can be approximated as a well known **M/D/1/K** queue [23][25].

If random service is used among the cells arriving in the same slot, then the average probability that cell arrivals contain a cell from the designated A-stream, but that cell

cannot successfully pass the queue I is

$$P_{ls}^{(I)} = \sum_{l=1}^{N-1} \sum_{i=l+1}^{N} \frac{l}{i} g_{a1}^{(i)} P(K + l + 1 - i). \tag{2.24}$$

Obviously, a cell from the designated A-stream must pass each queue successively along the VP_a, otherwise it will be lost. Overall, cell loss probability for the designated A-stream will be given by

$$P_{loss} = \frac{\text{total number of lost cells in } A\text{-stream}}{\text{total number of the cells carried in } A\text{-stream}} \tag{2.25}$$

$$= [1 - \prod_{I=1}^{Q} (1 - P_{ls}^{(I)})]/p_a^{(1)}. \tag{2.26}$$

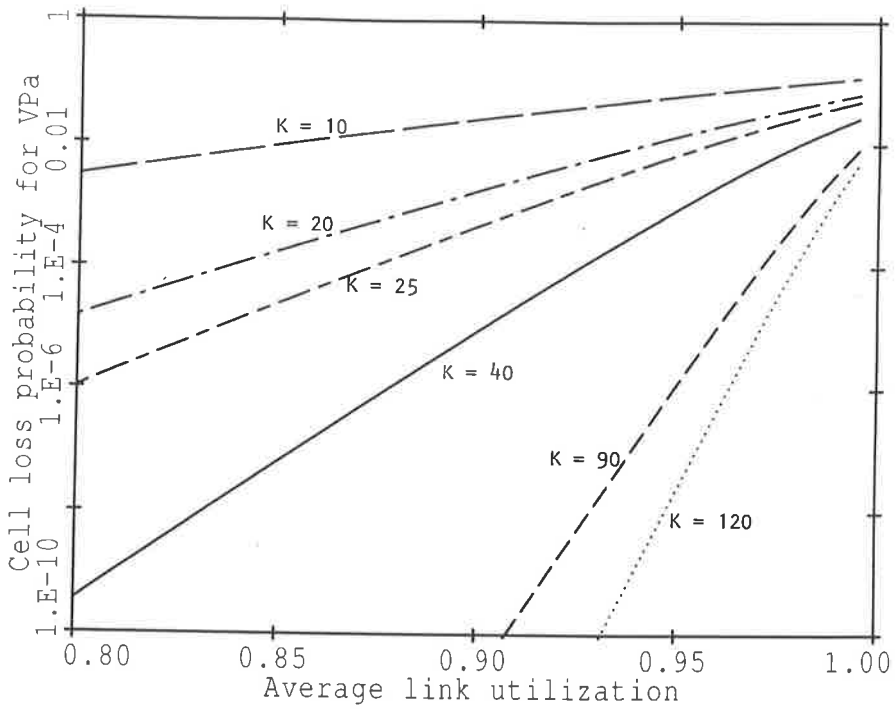


Figure 2.5 Cell loss probability versus the average link utilization for various buffer sizes ($Q = 5$, $N = 50$, $p = 0.018$).

Figure 2.5 illustrates the cell loss probability P_{loss} for VP_a versus the average link utilization for various buffer sizes. The cell arrival traffic streams are constant rate deterministic arrival processes. The computed P_{loss} is conditioned on a given tandem queueing network, described by its parameters Q, K, N, p, and $p_a^{(1)}$. Here VP_a is a virtual path consisting of 5 queues. Each queue is fed by $N = 50$ constant rate deterministic traffic streams,

including the designated A-stream. The average link utilization η is given by equation (2.7).

The results shown in Figure 2.5 indicate that extending the queueing buffer size can reduce the cell loss probability for an individual virtual path. For example, when $\eta = 0.8$, if buffer size increases from $K = 10$ to $K = 20$, the cell loss probability will be reduced from 3.03×10^{-3} to 1.48×10^{-5}. However, when heavy traffic is involved, say $\eta = 0.95$, in the same conditions, the cell loss probability is only reduced from 6.37×10^{-2} to 1.43×10^{-2}. The improvement becomes smaller. This situation may be improved by using larger buffers. For example, as shown in figure 2.5, when $\eta = 0.95$, the cell loss probability will be reduced to 2.37×10^{-8} by a large buffer of size $K = 120$. However, larger buffers increase both cell delay and the variations in cell delay. The buffer delays will also increase proportionally to the length of the virtual path cross the network. This may cause difficulties for real time service which require synchronization between the transmitter and receiver. Note that in this thesis, our consideration is focussed on the use of error control coding techniques combined with small size queue to improve the loss performance under high link utilization conditions rather than the use of large size queues.

2.6 Estimation of the statistics of bursty cell loss

2.6.1 Markov models for analysis of bursty cell loss

The cell loss due to buffer overflow may occur in bursts, particularly under high link utilization conditions. It is important to understand the statistical dependence that occur in the cell loss patterns when evaluating alternative error control coding schemes. In this section, we will consider the statistical dependence of bursty loss patterns due to congestion at the cell level.

Markov chain models are often used to represent the statistics of burst errors in digital transmission systems [27] [28] [29]. We consider these Markov chain models for our network application. Unfortunately the associated statistical dependence introduced by

tandem queueing network complicates mathematical solutions, the degree of difficulty being dependent upon the manner in which the statistical dependence is represented. Models must be chosen to be simple enough that meaningful performance estimates can be calculated in a straightforward manner either directly or by computer simulation. For the purpose of estimating the statistics of bursty cell loss, in this section, VP_a is approximated in terms of a single bulk arrival queue. However, the performance of the tandem queueing network is worse than a single bulk arrival queue. In chapter 3, we will check the accuracy by comparing the analytical results to the results obtained from the simulation in which an actual tandem queueing network without "independence assumption" is taken into consideration.

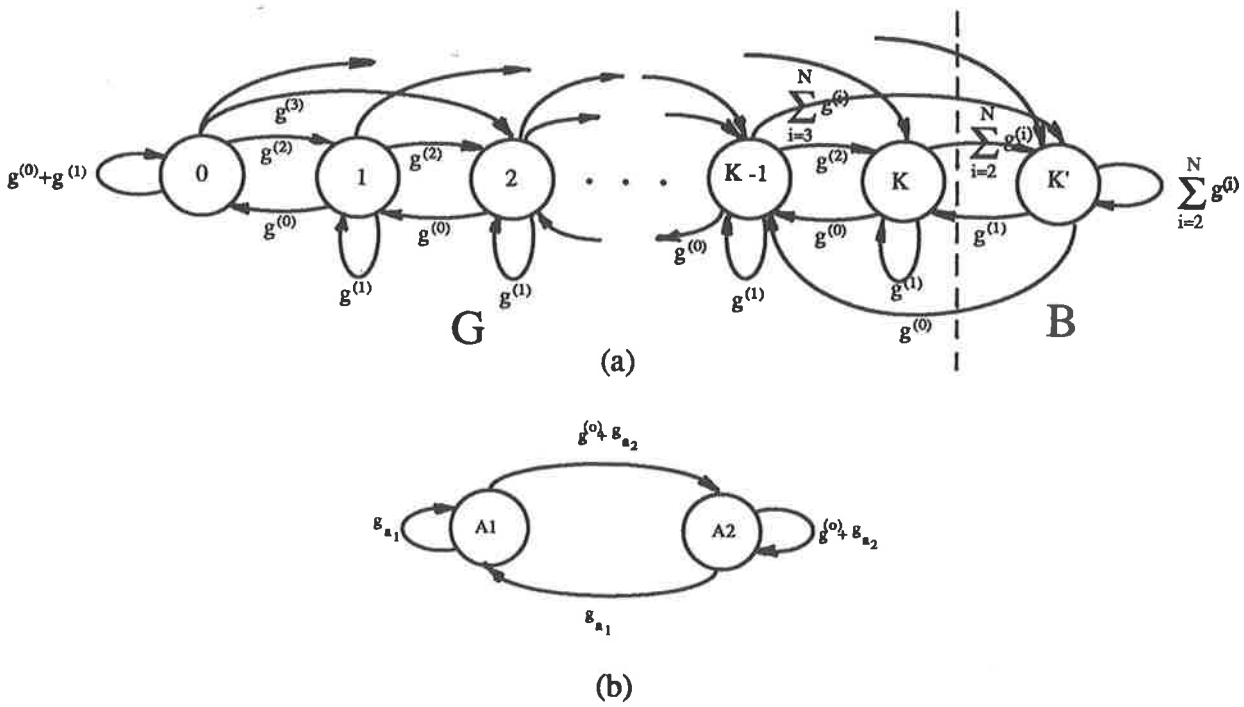


Figure 2.6 Markov models for analysis of bursty cell loss.

Figure 2.6 illustrates two Markov models proposed as a means of representing cell loss statistics. Model (a) is a Markov chain used to represent the status of the bulk arrival

queue in a given time slot. Model (a) contains a set of $K+2$ states, labelled

$$\Phi = \{\underbrace{0,\ 1,\ 2,\ ...,\ K}_{G},\ \underbrace{K'}_{B}\}.$$

As indicated, this is partitioned into two sets of states, designated G and B. The set $G = \{0, 1, 2, ..., K\}$ is a set of *good-transmission* states representing the fact that the bulk arrival queue is not overflow in a given time slot. The designation ξ of a state in G (ξ = 0, 1, ..., K) represents the case where ξ cells are waiting in the queue. $B = K'$ is single *bad-transmission* state representing that in a given time slot, the bulk arrival queue has overflowed, and $\xi = K$ cells are waiting in the queue. Let $[p_{\xi,\theta}]$ represent the set of transition probabilities for transition from state ξ to θ. Then

$$[p_{\xi,\theta}] =$$

$$\begin{bmatrix}
g^{(0)}+g^{(1)} & g^{(2)} & g^{(3)} & \cdots & g^{(N)} & 0 & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\
g^{(0)} & g^{(1)} & g^{(2)} & \cdots & g^{(N-1)} & g^{(N)} & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\
0 & g^{(0)} & g^{(1)} & \cdots & g^{(N-2)} & g^{(N-1)} & g^{(N)} & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & g^{(0)} & g^{(1)} & \cdots & \cdots & \cdots & \cdots & g^{(N)} \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 0 & g^{(0)} & \cdots & \cdots & \cdots & \cdots & g^{(N-1)}+g^{(N)} \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
0 & 0 & 0 & \cdots & 0 & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & g^{(0)} & g^{(1)} & g^{(2)} & \sum_{\xi=3}^{N} g^{(\xi)} \\
0 & 0 & 0 & \cdots & 0 & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & 0 & g^{(0)} & g^{(1)} & \sum_{\xi=2}^{N} g^{(\xi)} \\
0 & 0 & 0 & \cdots & 0 & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & 0 & g^{(0)} & g^{(1)} & \sum_{\xi=2}^{N} g^{(\xi)}
\end{bmatrix} \quad (2.27)$$

Model (b) is a two-state Markov chain used to represent the statistics of the cells arriving at the bulk arrival queue. In model (b), there two states A_1 and A_2. The state A_1 represents that in a given time slot, the i cell arrivals ($i \geq 1$) in the model (a) contains a cell from the designated A-stream. The state A_2 represents that in a given time slot, the i cell arrivals ($i \geq 0$) in the model (a) do not contain a cell from the designated A-stream. As indicated, the probability of leaving states A_1 and A_2 are $g^{(0)} + g_{a2}$ and g_{a1}, respectively.

$$g_{a1} = \sum_{i=1}^{N-1} g_{a1}^{(i)}$$

and

$$g_{a2} = \sum_{i=1}^{N-1} g_{a2}^{(i)},$$

where $g^{(0)}$, $g_{a1}^{(i)}$, and $g_{a2}^{(i)}$ are given by equations (2.4), (2.5), and (2.6). From model (b), we can obtain that

$$P(A_1) = g_{a1}$$

and

$$P(A_2) = g^{(0)} + g_{a2}.$$

The cell loss pattern for the designated A-stream carried on VP_a is not only related to the number of cell arrivals from N input VPs in the same time slot but is also related to the queueing length in that time slot. However, the queueing length must be a function involving the traffic of previous time slots including those in which there is no cell arrival from the designated A-stream.

To estimate the cell loss pattern for the designated traffic stream, in the following sections, we first use the Markov model (a) to evaluate the bulk arrival queue overflow patterns in term of time slots. Each time slot may or may not contain a cell arrival from the designated A-stream. Secondly, we use the statistics of cell arrival provided by model (b) to estimate the cell loss pattern for the designated stream by means of the bulk arrival queue overflow patterns which are obtained from the model (a).

2.6.2 Cell loss-free run distribution

Let $Z = \{z_i : i = 1, 2, ...\}$ represent a sequence of states in which transitions occur synchronously with the cell sequence $X = \{x_i : i = 1, 2, ...\}$ which is carried on the path VP_a. Let z_i take values

$$z_i = \begin{cases} 0, 1, 2, ..., K & \text{for } z_i \in G \\ K' & \text{for } z_i \in B \end{cases} \tag{2.28}$$

The *cell loss-free run distribution* of length r or greater is defined to be a conditional probability

$$P(0^r|1) = P(z_i = 0, 1, 2, ..., K : 1 \leq i \leq r | z_o = K').$$

This is a conditional probability that given that a cell loss has occurred, it will be followed by r or more consecutive cells which are not lost.

As mentioned in last section, to estimate $P(0^r|1)$ our procedures are first to evaluate the bulk arrival queue overflow-free run distribution using the Markov chain model (a).

Let us define $P'(0^{r'}|1)$ to be the bulk arrival queue overflow-free run distribution. That is, $P'(0^{r'}|1)$ is a conditional probability that given that the bulk arrival queue has overflowed in a time slot, it will be followed by r' or more consecutive time slots in which the queue has not overflowed. Let $P'(10^{r'})$ be the probability that after leaving the state B, the system does not return to the state B at least in the following $r'-1$ steps. Let $P'(1)$ be the probability that the system transit into state B. Using the matrix of transition probabilities $[p_{\xi,\theta}]$ in equation (2.27), we have

$$P'(1) = p_{K'K'} + \sum_{i=0}^{K} p_{iK'} \tag{2.29}$$

$$= \sum_{i=2}^{N} g^{(i)} + \sum_{j=1}^{N-1} \sum_{i=j+1}^{N} g^{(i)} \tag{2.30}$$

$$\tag{2.31}$$

$$P'(10^{r'}) = P'(10)P'(0^{r'-1}) \tag{2.32}$$

where

$$P'(10) = g^{(o)} + g^{(1)}$$

$$P'(0^{r'-1}) = \prod_{j=1}^{r'-1} (\sum_{i=0}^{j+1} g^{(i)}) \quad 0 \le i \le N, \text{ if } i > N, \text{ then } g^{(i)} = 0.$$

Then we can obtain $P'(0^{r'}|1)$, that is

$$P'(0^{r'}|1) = \frac{P'(10^{r'})}{P'(1)} \tag{2.33}$$

This conditional probability $P'(0^{r'}|1)$ gives the bulk arrival queue overflow-free run distribution.

Secondly, we use the Markov chain model (b) to estimate the cell loss-free run distribution by means of $P'(0^{r'}|1)$. From the model (b), we know that $P(A_1) = g_{a1}$ is the probability

of at least one cell arrival from the designated A-stream in a time slot. The length r of cell loss-free runs in A-stream can be approximated as

$$r \approx g_{a1} r'. \tag{2.34}$$

The reason that it is not exactly equal is that there will be a small probability that cell loss-free runs in A-stream may be extended as follows:

(1) When the bulk arrival queue has overflowed, there is no cell arrival from the designated A-stream. In this case, a cell loss-free run may not be broken into two or more separate runs despite the occurrence of queue overflow.

(2) When the bulk arrival queue has overflowed, cell arrivals from designated A-stream are not lost. Even though a queue overflows, one cell is always transmitted, and there is, therefore, a small probability that the cell from A-stream is accepted by an overflowing queue (see Section 2.5). In this case, the cell loss-free run in the designated A-stream will again be extended.

When link utilization is high, the probability that above two situations occur will be smaller than the case of having cell arrivals from the A-stream being lost. Because the queueing buffer has memory, then the probability that the queue overflows in groups of consecutive time slots will be much greater the probability that the queue overflows in a single isolated time slot, especially when we consider the high link utilization conditions. From the view of long term performance, we can approximate

$$P(0^r|1) \approx P'(0^{r'}|1), \tag{2.35}$$

where $r = \lfloor g_{a1} r' \rfloor$, with $\lfloor w \rfloor$ being the largest integer which is less than w.

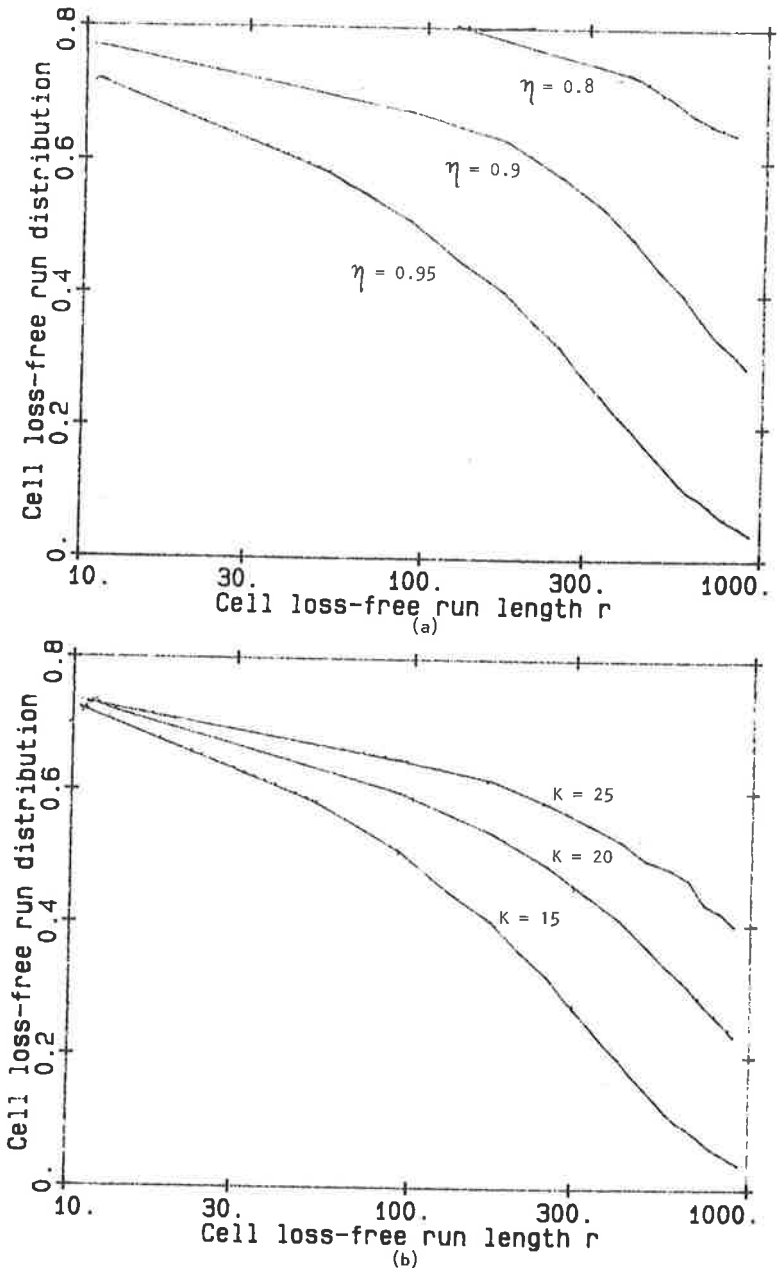


Figure 2.7 Cell loss-free run distribution.

(a) Cell loss-free run distribution for different average link utilization. ($K = 15$)

(b) Cell loss-free run distribution for different queueing buffer sizes. ($\eta = 0.95$)

Figure 2.7 illustrates how the cell loss-free run distribution varies with the average link utilization. The parameters used in this example are the same as those used in Figure 2.5.

The results in Figure 2.7 (a) show that the cell loss-free distribution $P(0^r|1)$ is sensitive to the traffic load. However, the result in Figure 2.7 (b) shows that extending the buffer size can significantly increase the cell loss-free interval among the bursts even for high link utilization, say $\eta = 0.95$. For example, in Figure 2.7 (b), when the average link utilization is $\eta = 0.95$, the buffer size increases from $K = 15$ to $K = 25$, $P(0^{300}|1)$ also increases from 0.281 to 0.596. This means that for $\eta = 0.95$, if using a buffer size of $K = 15$, the 28.1% of the cell loss-free intervals between the losses are greater than 300 consecutive cells. If using a buffer of $K = 25$, then 59.6% of the cell loss-free intervals between the lost cells are greater than 300 consecutive cells. This may provide more opportunities for using error recovery coding techniques to improving the performance, because the codes used for burst loss recovery must have enough cell loss-free guard space between two burst losses. This will be one of the topics raised in chapter 4. Note that when buffer size increases, the burst loss lengths may be increased. This requires us to consider another high order distribution: cell loss pattern distribution.

2.6.3 Cell loss pattern distribution

Let $P(u, v)$ [27] represent the probability that a block of v cells which are transmitted through VP_a, contain exactly u lost cells due to buffer overflow. Let $P(u', v')$ represent the probability that an interval of v' time slots consists of u' time slots in which the bulk arrival queue has overflowed. For model (a), we define $f_i(u', v')$ as the probability of exactly u' transitions into the B state in v' steps, given that the initial state was i, where

$$u = \lfloor g_{a1} u' \rfloor,$$

$$v = \lfloor g_{a1} v' \rfloor.$$

Then the following recurrence relation can be derived.

$$f_i(u', v') = \sum_{j=0}^{K} p_{ij} f_j(u', v' - 1) + p_{iK'} f_{K'}(u' - 1, v' - 1) \tag{2.36}$$

Note that the equation (2.36) can be computed interactively using the following initial conditions:

1. $f_i(u', v') = 0$, for $u' > v'$,

2. $f_i(u', v') = 0$, for $v' < 0$ or $u' < 0$,

3. $f_i(0, 0) = 1$,

the last condition merely indicates that zero losses are always made in zero transitions. Since the initial state i can be any one of states $\{0, 1, ..., K, K'\}$, thus we have

$$P(u', v') = \sum_{i=0}^{K'} p_i f_i(u', v') \tag{2.37}$$

where p_i are the state probabilities. Then the cell loss distribution $P(u, v)$ is approximated by

$$P(u, v) \approx P(\lfloor g_{a1} u' \rfloor, \lfloor g_{a1} v' \rfloor), \tag{2.38}$$

where

$$u = \lfloor g_{a1} u' \rfloor,$$

$$v = \lfloor g_{a1} v' \rfloor.$$

Figure 2.8 shows $P(u, v)$ for the designated traffic stream with various values of average link utilization. The parameters are the same as those used in Figure 2.5.

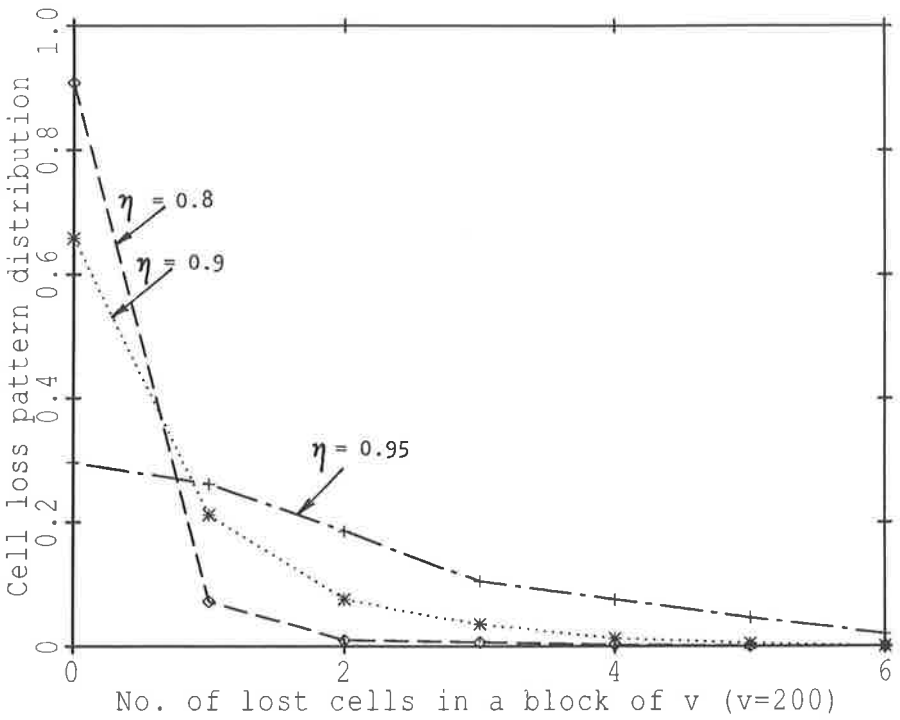


Figure 2.8 Cell loss pattern distribution ($K = 15$).

2.7 Discussion

A virtual path through an ATM network is modelled as a tandem queueing network. The arrival processes allowed to enter the tandem queueing network are mixtures of constant rate deterministic arrival processes. Our studies have been mainly considered on a designated A-stream as it goes through a tandem queueing system with small size queues. This allows us to study the statistics of cell loss due to congestion at the cell level with mixtures of constant rate deterministic arrival processes and long bursts within the link capacity. Since our analytical approach is approximate, we will check its accuracy by comparing it to the results obtained from simulation in chapter 3. In the simulation, an actual tandem queueing system has been used to obtain the performance measures for

the designated traffic stream. That is,

- Interdeparture process has been used to observe the designated traffic stream as it goes through the tandem queueing network.
- We have introduced no "independence assumption" between the adjacent nodes.

The results of statistics of cell loss can be summarized as follows. Extending the queueing buffer size can reduce the cell loss probability, but when traffic load is heavy, say $\eta > 0.95$, using small size queues, this improvement may be insignificant. By contrast, under the same conditions, extending the queueing buffer size can significantly increase the cell loss-free interval between the burst losses. This indicates the possibility of using the error control coding techniques combined with extending the queueing buffer size to improve the network performance. To properly design the coding schemes for recovering long bursts of cell loss, we have to very carefully examine the higher order distributions of bursty cell loss patterns. This will be discussed in following chapters.

The high order distributions of cell loss pattern can provide more information beyond the average cell loss rate, and this is able to estimate the end-to-end overall quality of services (QOS) for an individual service over the ATM network, especially for some real-time services. For example, for an end-to-end video transmission over an ATM network, the cell loss pattern distribution provides the distribution of the number of lost cells in an interrupted frame, and the cell-free run distribution gives information concerning the interval between two interrupted frames.

Chapter 3

SIMULATION STUDY OF STATISTICS OF CELL LOSS

Summary

Simulation is arranged to represent a virtual path (VP) as an actual tandem queueing network consisting of a number of queues in series. At the switches, the switching functions result in splitting of traffic streams going to different destinations and mixing of traffic streams going to the same output link to enter the output queues [61]. Each queue along the VP is implemented as a discrete-time bulk arrival queue with a finite buffer.

Two types of cell arrival processes are considered in the simulation: i) mixtures of constant rate deterministic arrival processes and ii) mixtures of homogeneous discrete-time burst-silence processes [72][73][83]. Our studies have been mainly considered on a designated traffic stream as it goes through the tandem queueing network, especially for bursty traffic. This allows us to study the statistics of cell loss due to congestion at both burst level and cell level subject to cell loss recovery in ATM networks.

The software is written using FORTRAN and runs on a VAX system. The simulation using the Monte Carlo method is focussed on a VP with high link utilization as this region has higher loss probability. This is the region of most interest to us for the application of forward error control coding techniques discussed in chapters 4 and 5. For low loss rate, the simulations require an excessive amount of computing resources, and we have

not addressed that region.

Firstly, we simulate the actual tandem queueing network using mixtures of constant rate deterministic arrival processes as the random arrival traffic. The results obtained from the simulation including cell loss probability and high order distributions of burst cell loss are compared with the results obtained from analytical studies in chapter 2. The simulation results demonstrate that although the analytical results are approximate, they are quite accurate over the region of interest to our studies.

Secondly, we simulate the actual tandem queueing network using mixtures of homogeneous burst-silence processes as the bursty arrival traffic [73]. The aim is to study the statistics of cell loss for bursty arrival traffic at both burst and cell levels, as the exact queueing analysis for an ATM multiplexer with bursty traffic leads to very complex queueing model due to queue transient effects, and seems not tractable. Therefore its performance measures cannot be obtained exactly by analysis, and these measure have not been studied very much so far. The results obtained from the simulation are subsequently used as a basis for the evaluation of the performance of error recovery coding in chapter 4.

3.1 Simulation model

A virtual path through an ATM network may pass through a number of switching nodes in series. Each switching node may be connected to several different input and output optical fiber links. Each link may carry a number of VPs. A virtual path VP_a under consideration is modelled as a tandem queueing network consisting of Q single server queues in series, which operate on a first-in-first-out (FIFO) basis. The switching functions result in splitting of traffic streams going to different destinations and combining of different streams going to the same output link to enter the output queues. A queue I (I = 1, 2, ..., Q) along VP_a has a finite buffer of length K, and is attached by N independent input VPs, including VP_a and $VP_n^{(I)}$ (n = 1, 2, ..., $N-1$). A designated A-stream is carried on the virtual path VP_a through the tandem queueing network, while the other virtual paths $VP_n^{(I)}$ carry a D-stream.

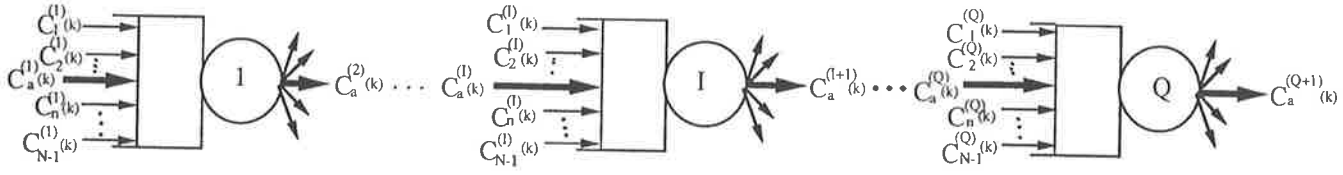


Figure 3.1 Simulation model for a virtual path.

In Figure 3.1, $\{C_a^{(1)}(k)\}$ represents the cell sequence of the designated A-stream arriving at the first queue of the tandem queueing network. $\{C_a^{(I)}(k)\}$ represents the interdeparture process of the designated A-stream as it departs the queue $I-1$ and goes to join the queue I $(2 \leq I \leq Q)$. Likewise, $\{C_n^{(I)}(k)\}$ $(n = 1, 2, ..., N-1)$ represents the cell sequence of a D-stream which is carried on the virtual path $VP_n^{(I)}$, and joins the queue I. k is a time slot index.

In the simulation, the actual tandem queueing network is used to obtain the performance measures for the designated traffic stream. In this model

- We have introduced no "independence assumption" between the adjacent queues.
- The interdeparture process is used to observe the designated traffic stream as it goes through the tandem queueing network, so that the queue transient effects is taken into account.

3.2 Generation of the cell traffic

3.2.1 Random traffic

In the simulation, we first consider both the A-stream and the D-streams to be discrete-time constant rate deterministic arrival processes. According to the justification made in

section 2.4, if a large number N of traffic sources are mixed in each short queue of size K along the tandem queueing network, and each source contributes a small fraction to the total load so that $N >> K$, then the short-term behavior of the mixture of streams can be characterized using discrete-time Bernoulli process [73][82]. The definition of the constant rate deterministic arrival process is given in section 2.3. The cell sequence from a constant rate deterministic arrival process source is generated using a Monte Carlo method in the following manner.

1. Generate a random number Ψ $(0 \leq \Psi \leq 1)$.

2. For a given cell arrival probability p, the cell sequence $\{C_n^{(I)}(k)\}$ is generated as follows:

$$C_n^{(I)}(k) = \begin{cases} 1 & \text{if } \Psi \geq p \\ 0 & \text{if } \Psi < p \end{cases},$$

   where 1 represents a cell arrival in a slot, and 0 represents no cell arrival in that slot.

Likewise, the cell sequence $\{C_a^{(1)}(k)\}$, which comes from the designated A-stream and arrives at the first queue of the tandem queueing network, can be generated in the same manner by replacing p with $p_a^{(1)}$.

3.2.2 Mixture of homogeneous burst-silence processes

A burst-silence process is an ATM traffic source in which cells are spaced by fixed intervals at the peak rate which is allocated to the call, and occur in bursts of variable length, separated by silence intervals of variable length [72]. A voice source is a well-known example of a burst-silence process if the source coding scheme employs speech activity detection and silence suppression. Further, a data source in interactive data communication can be classified as a burst-silence source [72][73][83].

This traffic model has been used in [64][72][73][75][83] to consider the performance of an ATM multiplexer at the burst level with an assumption of the burst length is much greater than the queueing buffer size, and an approximation of ignoring the queueing

transient time effects. In this chapter, we will use the mixtures of homogeneous discrete-time burst-silence processes as the bursty cell arrival traffic model in the simulation of an actual tandem queueing network. Our attention will be focussed on a designated traffic stream passes through the tandem queueing network to study the statistics of cell loss due to congestion at both the cell level and burst level. In the simulation, an iterdeparture process has been used to observe the designated traffic stream as it goes through the tandem queueing network. We have introduced no "independence assumption" between the adjacent queues. The queue transient effects have been taken into account in the simulation process.

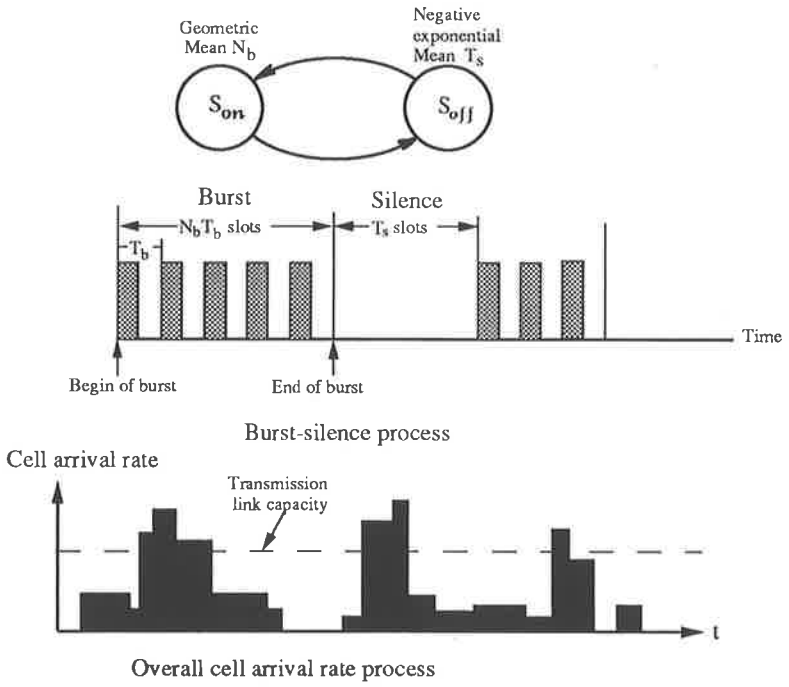


Figure 3.2 Mixture of homogeneous discrete-time burst-silence traffic.

As shown in Figure 3.2, a burst-silence source has two states S_{on} and S_{off}. S_{on} is a burst state. S_{off} is a silence state. Such a source emits cells with a constant cell interval time T_b slots only when it is in the state S_{on}, and emits idle cells if the source is in the state S_{off}. Let the random variable M_b represent the number of cells within a burst. M_b is assumed to be geometrically distributed with mean value of N_b cells [72]. The probability

density function is given by

$$P_B(M_b) = \left(1 - \frac{1}{N_b}\right)^{M_b - 1} \frac{1}{N_b}. \tag{3.1}$$

Let the random variable M_s represent the number of slots in a silence. M_s in our model is assumed to be negative exponentially distributed with mean value of T_s slots [72]. Its probability density function is given by

$$P_T(M_s) = \frac{1}{T_s} e^{-\frac{M_s}{T_s}}. \tag{3.2}$$

We adopt this burst-silence source, because it can present the bursty nature of the traffic carried in ATM network [64][72][73]. A burst-silence source with above distribution can be characterized by following set of parameters:

$$\text{The peak (burst) rate} \qquad \lambda = 1/T_b \tag{3.3}$$

$$\begin{aligned} &\text{The mean burst duration} && T_{on} = N_b T_b \\ &\text{The mean silence duration} && T_{off} = T_s \end{aligned} \tag{3.4}$$

$$\begin{aligned} &\text{The source burstiness} && (T_{on} + T_{off})/T_{on} = (N_b T_b + T_s)/N_b T_b \\ &\text{The mean cell rate} && \lambda T_{on}/(T_{on} + T_{off}) = N_b/(N_b T_b + T_s) \\ &\text{The mean cycle duration} && T_{on} + T_{off} = N_b T_b + T_s \end{aligned} \tag{3.5}$$

A burst-silence source is in a burst state with probability

$$q = T_{on}/(T_{on} + T_{off}), \tag{3.6}$$

and silent with probability $1 - q$. Then the average rate of the designated A-stream arrives at the first queue of the tandem queueing network is given by

$$p_a^{(1)} = \lambda q. \tag{3.7}$$

The overall cell arrival rate process of the multiplexed traffic streams which join each queue along the tandem queueing network can be described as a succession of constant rate bursts, as shown in Figure 3.2. That is, the overall cell arrival rate changes by jumps

at the instants of burst arrival and termination [72]. Then the average rate of the overall arrival traffic at the first queue of the tandem queueing network is given by

$$\eta = N\lambda q, \tag{3.8}$$

and the peak rate is given by

$$\eta_p = N\lambda. \tag{3.9}$$

3.2.3 Some characteristics of bursty traffic model

As discussed in chapter 2, the cell loss caused by congestion can be decomposed into two different levels - burst level and cell level. Now we recall the definition of both burst level and cell level congestion which are given in chapter 2:
i) at the burst level, congestion occurs when the total arrival rate exceeds the capacity of the output link for an average period greater than the inter-cell time of the reference source [72][79].
ii) at the cell level, congestion is caused by simultaneous cell arrivals occurring in a time span equivalent to the inter-cell time of a reference stream.

Now we discuss some characteristics the mixture of homogeneous burst-silence traffic as follows:

1. According to bursty arrival traffic behavior [59], here we only consider that the number N of burst-silence sources which is attached to a queue along the tandem queueing network, is less than the queueing buffer size K, so that the inter-cell time of a bursty source is less than the queueing buffer size K.

2. If the peak rate of the mixture of homogeneous burst-silence sources is less than the output link capacity, (i.e. without overload), then cell loss belongs to the cell level. In this case, if the burst length is much greater than the queueing buffer size, then the queue transient effects could be ignored. Otherwise, the queue transient effects have to be taken into account.

3. If the peak rate of the mixture of homogeneous burst-silence sources exceeds the output link capacity (i.e., overload is involved), then two different situations must be taken into consideration.

   - A burst-silence source with the average burst length less than the queueing buffer of K slots is referred to as a short burst source. In the mixed traffic, the average length of a fixed burst composition in which the peak rate exceeds the link capacity, will be less than the average length of a single burst-silence source due to mixing of traffic in the queue. In this case, the cell loss is likely to be caused by simultaneous cell arrivals occurring in a short time span, and cell loss could be approximated to be the cell level. However, the queue transient effects must be taken into account.
   - A burst-silence source with the average burst length greater than queueing buffer of K slots is referred to as a long burst source. In the mixed traffic, the average length of a fixed burst composition in which the peak rate exceeds the link capacity, will be greater than the inter-cell time of a single burst-silence source, and loss belongs to the burst level.

3.2.4 Generation of a burst-silence source

In the simulation, the cell sequence from each burst-silence source is generated using the well-known Monte Carlo sampling technique. Now we first give an introduction to the Monte Carlo sampling method to produce random variables with a given distribution. Let us suppose that the random variable t to be generated has a probability distribution function $F_c(t)$ which is the right tail cumulate. If the random variable is continuous, then

$$F_c(t) = \int_t^{\infty} f(x)dx \tag{3.10}$$

where $f(x)$ is the density function. If the random variable is discrete, then

$$F_c(t) = \sum_{x=t}^{\infty} f(x) \tag{3.11}$$

Now we designate $F_c^{-1}(\Psi) = t$ as the inverse function of $F_c(t)$. As $f(x) \geq 0$, $F_c(t)$ is monotonously increasing with t, so that $F_c^{-1}(\Psi)$ is unique. We note that the domain of $F_c^{-1}(\Psi)$ is $0 \leq \Psi \leq 1$, and the range corresponds to the domain of $F_c(t)$, which for t = time, would be $F_c^{-1}(\Psi) \geq 0$. We evaluate $t = F_c^{-1}(\Psi)$, where successive random numbers are assigned to Ψ. Each random number Ψ taken from a uniform distribution will give a random number t from the designed distribution $F_c(t)$. Hence, the Monte Carlo sampling simply reduces to using random numbers to enter the ordinate scale, and reading values from the scale designated as t. However, the simulation using Monte Carlo techniques is only feasible for relatively high cell loss probability, because of the simulation of lower cell loss probability needs a huge amount of computer resources, and this is also limited by the random number source built-in the computer system.

Secondly, we consider how to generate the cell sequence $\{C_a^{(1)}(k)\}$ as a burst-silence process, where k is the time slot index. Now we let $C_a^{(1)}(k) = C(\rho, \alpha)$, where ρ represents the ρth uninterrupted state period, and α represents the αth time slot in the ρth uninterrupted state period. The term"uninterrupted state period" represents an uninterrupted interval of time slots in which the burst-silence process is in one of S_{on} and S_{off} states. If during the ρth uninterrupted state period, the burst-silence source is in the state S_{on}, then

$$C(\rho, \alpha) = \begin{cases} 1 & \text{a cell arrival in the } \alpha\text{th time slot of the } \rho\text{th uninterrupted state period} \\ 0 & \text{no cell arrival in the } \alpha\text{th time slot of the } \rho\text{th uninterrupted state period} \end{cases}$$

If during the ρth uninterrupted state period, the burst-silence source is in the state S_{off}, then

$$C(\rho, \alpha) = 0 \qquad \text{no cell arrival in the } \alpha\text{th time slot of the } \rho\text{th uninterrupted state period.}$$

The procedures are:

- *Step 1:* Generate a random number M_b which represents the number of cells within a burst, and obeys a geometric distribution. Using the probability density function given by equation (3.1), then we have

$$F_c(M_b) = \sum_{x=M_b-1}^{\infty} (1 - 1/N_b)^x (1/N_b) \tag{3.12}$$

$$= \frac{1}{N_b}(1-1/N_b)^{M_b-1}[1+(1-1/N_b)+(1-1/N_b)^2+...]$$
$$= (1-1/N_b)^{M_b-1}$$

If Ψ_1 is a uniformly distributed random number between 0 and 1, then M_b can be obtained by

$$M_b = \lfloor \frac{ln(\Psi_1)}{ln(1-1/N_b)} \rfloor + 1, \tag{3.13}$$

where Ψ_1 represents $F_c(M_b)$, and $\lfloor w \rfloor$ represents the largest integer which is less than w.

- *Step 2:* Generate the cell sequence $C(\rho, \alpha)$. Since when the burst-silence process is in the state S_{on}, cells are generated with a constant interval of every T_b slots. For a given T_b, $C(\rho, \alpha)$ is generated by

$$C(\rho,\alpha) = \begin{cases} 1 & \text{if } \alpha = mT_b \quad m = 0, 1, 2, \ldots M_b \\ 0 & \text{if } \alpha \neq mT_b \end{cases}, \alpha \leq M_bT_b \tag{3.14}$$

- *Step 3:* When α reaches M_bT_b, then generate a new random number M_s which represents the length of uninterrupted time for a burst-silence process in the state S_{off}, and obeys a negative exponential distribution with a density function given by equation (3.2). Then the distribution function is given by

$$F_c(M_s) = \int_{M_s}^{\infty} P_T(M_s)dM_s \tag{3.15}$$
$$= e^{-\frac{M_s}{T_s}}$$

with average of T_s slots. If Ψ_2 is a uniformly distributed random number between 0 and 1, then M_s will be obtained by

$$M_s = \lfloor -T_s ln(\Psi_2) \rfloor, \tag{3.16}$$

where Ψ_2 represents $F_c(M_s)$.

- *Step 4:* Generate the cell sequence $C(\rho+1, \alpha)$. Since when a burst-silence process is in state S_{off}, the burst-silence source emits idle cells, i.e.,

$$C(\rho+1, \alpha) = 0 \quad \alpha \leq M_s.$$

- *Step 5:* When α reaches M_s, then repeat *Step 1* to generate a new random number M_b which represents the number of cells in a new burst period.

Likewise, we can implement a cell sequence $\{C_n^{(I)}(k)\}$ as a burst-silence process to represent the traffic stream which comes from $VP_n^{(I)}$ and joins the queue I ($I = 0, 1, ..., Q$).

3.3 Simulation procedures

As shown in Figure 3.1, in the simulation we use:

$$C_n^{(I)}(k) = \begin{cases} 1 & \text{one cell arrival from } VP_n \text{ in the } k\text{th slot} \\ 0 & \text{no cell arrival from } VP_n \text{ in the } k\text{th slot} \end{cases}$$

to represent the D-stream which come from $VP_n^{(I)}$ ($1 \leq n \leq N-1$) and join the queue I ($0 \leq I \leq Q$), excluding VP_a.

$$C_a^{(1)}(k) = \begin{cases} 1 & \text{one cell arrival from } VP_a \text{ in the } k\text{th slot} \\ 0 & \text{no cell arrival from } VP_a \text{ in the } k\text{th slot} \end{cases}$$

represents the cell sequence of the designated A-stream which arrives at the first queue of the tandem queueing network.

$$C_a^{(I+1)}(k) = \begin{cases} 1 & \text{one cell being carried on } VP_a \text{ in the } k\text{th slot} \\ 0 & \text{no cell being carried on } VP_a \text{ in the } k\text{th slot} \\ -1 & \text{one cell being lost from } VP_a \text{ in the } k\text{th slot} \end{cases},$$

represents the interdeparture process of the designated A-stream which departs the queue I and goes to join the queue $I+1$.

The simulation procedures are:

- *Step 1:* Generate the cell arrival sequence $\{C_n^{(I)}(k)\}$ for $I = 1, 2, ..., Q$, and $\{C_a^{(1)}(k)\}$ according to designated distributions.

- *Step 2:* Simulate the bulk arrival queue $I = 1, 2, ..., Q$ and determine the outgoing cell sequence $\{C_a^{(I+1)}(k)\}$. Since cells may be delayed or lost as they are multiplexed and buffered in the queue I, then the status of the output cell sequence $\{C_a^{(I+1)}(k)\}$ is not only related to the number of cells arriving at the queue in the same slot, but also related to the queueing length in that time slot. The queueing length must be a function involving the traffic of previous slots including those in which there is no cell arrival from the designated A-stream. We use $\{C_a^{(I+1)}(k)\}$ to record the status of the designated A-stream as it passes the queue I, and goes to the next queue $I+1$. This ensures that the independence assumption is not needed, and the queue transit effects have been taken into account. Cells arriving at a queue in the same time slot are randomly multiplexed and served.

- *Step 3:* Evaluate the statistics of cell loss using the obtained $\{C_a^{(Q+1)}(k)\}$, which is the cell sequence at the output of the last queue Q.

Since the simulation process starts with an empty queue, a warm up period of 10^3 slots has been setup to ensure that the results are estimated on the basis of steady simulation process.

3.4 Performance measures

The cell loss due to buffer overflow may occur in bursts, especially for high link utilization, or bursty traffic. It is important to understand the cell loss probability and the statistical dependencies that occur in cell loss patterns when evaluating error protection procedures.

3.4.1 Cell loss probability

The cell loss probability is estimated as

$$P_{loss} = \frac{\text{Total number of lost cells in } A\text{-stream}}{\text{Total number of the cells carried in } A\text{-stream}} \tag{3.17}$$

The total number of the cells carried in A-stream, denoted as N_I can be obtained from the output cell sequence $\{C_a^{(Q+1)}(k)\}$. This is

$$N_I = \sum_{k=1}^{\omega} |C_a^{(Q+1)}(k)| \tag{3.18}$$

where ω ($\omega >> Q$) is the total number of the simulation running cycles. In the output cell sequence $\{C_a^{(Q+1)}(k)\}$, we have denoted

$$C_a^{(Q+1)}(k) = \begin{cases} 1, & \text{a cell arrival from the } A\text{-stream is successfully transferred through } VP_a \text{ in the } k\text{th time slot} \\ 0, & \text{no cell arrival from the } A\text{-stream in the } k\text{th time slot} \\ -1 & \text{a cell arrival from the } A\text{-stream is lost in the } k\text{th time slot} \end{cases}$$

Then the total number of lost cells in the A-stream, denoted as N_L, is the sum of the negative number in the output cell sequence $\{C_a^{(Q+1)}(k)\}$. That is,

$$N_L = \sum_{k=1}^{\omega} L_a(k) \tag{3.19}$$

where

$$L_a(k) = \begin{cases} 1, & \text{if } C_a^{(Q+1)}(k) = -1 \\ 0, & \text{Otherwise} \end{cases}$$

Overall,

$$P_{loss} = \frac{N_L}{N_I}. \tag{3.20}$$

3.4.2 Cell loss-free run distribution

In chapter 2, we have estimated the cell loss-free run distribution $P(0^r|1)$ using analytical methods. Now we consider the estimation of performance measures for $P(0^r|1)$ using simulation methods. Both random and bursty traffic will be considered. Recall the definition of the cell loss-free run distribution $P(0^r|1)$ given in section 2.6, that is, $P(0^r|1)$ is defined as a conditional probability that given a cell loss has occurred, it will be followed by r or more consecutive loss-free cells.

To estimate $P(0^r|1)$ we define a loss-free interval of length h in $\{C_a^{(Q+1)}(k)\}$ to be a range between two lost cells contains exact h consecutive cells which are not lost. Let

$N_g(h)$ be the cumulated number of loss-free intervals of length h in $\{C_a^{(Q+1)}(k)\}$, N_g be the total number of loss-free intervals in $\{C_a^{(Q+1)}(k)\}$, and

$$N_g = \sum_h N_g(h).$$

N_g and $N_g(h)$ can be obtained from the output cell sequence $\{C_a^{(Q+1)}(k)\}$. Recall the definition of $P(0^r|1)$, then $P(0^1|1)$ represents the conditional probability that a loss-free interval of length $h \geq 1$ follows a lost cell. Then, the $P(0^1|1)$ can be estimated by

$$P(0^1|1) = \frac{N_g}{N_L}.$$

Likewise, $P(0^2|1)$ is defined as the conditional probability that a loss-free interval of length $h \geq 2$ follows a lost cell. Obviously, $P(0^r|1)$ will include $P(0^2|1)$. Then we have

$$P(0^2|1) = P(0^1|1) - N_g(1)/N_g.$$

Similarly, we have that

$$\begin{aligned} P(0^r|1) &= P(0^{r-1}|1) - N_g(r-1)/N_g, \\ &= P(0^1|1) - \frac{N_g(1) + N_g(2) + ... + N_g(r-1)}{N_g}. \end{aligned} \tag{3.21}$$

3.4.3 Cell loss pattern distribution

In chapter 2, we have also estimated the cell loss pattern distribution $P(u,v)$ using analytical method. Now we consider the estimation of performance measures for $P(u,v)$ using simulation method. Both random and bursty traffic will be considered. Recall the definition of the cell loss pattern distribution $P(u,v)$ given in section 2.6, that is, the cell loss pattern distribution $P(u,v)$ is the probability that a block of v cells transmitted through VP_a contains exact u cells lost due to buffer overflow.

We divide the output cell sequence $\{C_a^{(Q+1)}(k)\}$ into blocks of v cells. Let N_B be the total number of the blocks, and $N_B(u)$ be the cumulated number of the blocks in which u cells are lost $(u = 0, 1, 2, ..., v)$. Then the cell loss pattern distribution can be estimated by

$$P(u,v) = \frac{\text{Cumulated number of blocks in which } u \text{ cells are lost}}{\text{Total number of received blocks of } v \text{ cells}} \tag{3.22}$$

$$= \frac{N_B(u)}{N_B}$$

Let $\overline{N}_A$ be the average number of lost cells in a given block of v cells. Then we have

$$\overline{N}_A = \sum_{u=0}^{v} uP(u,v) \tag{3.23}$$

3.5 Case Studies

The simulation results given below present the network performance for two different cases. In case I, the mixtures of constant rate deterministic arrival processes are used as the random cell arrival traffic. In case II, the mixtures of homogeneous burst-silence traffic are considered to be the bursty arrival traffic.

3.5.1 Statistics of cell loss with random arrival traffic

Firstly, we consider a single bulk arrival queue with $N = 50$ constant rate deterministic arrival sources. Each source generates cells with a probability p per slot. Figure 3.3 shows the simulation results of cell loss probability versus the average link utilization for different buffer sizes, where the average link utilization is given by

$$\eta = \sum_{i=0}^{N} i \begin{pmatrix} N \\ i \end{pmatrix} p^i (1-p)^{N-i}, \tag{3.24}$$

and the peak rate of overall cell arrivals is given by

$$\eta_p = Np. \tag{3.25}$$

To determine whether the simulation model predicts the system performance well, we compare the simulation results to a well known M/D/1/K queue [23] in Figure 3.3 and Table 3.1. As discussed in section 2.5, when $N >> K$, the simulation model should be equivalent to an M/D/1/K queue.

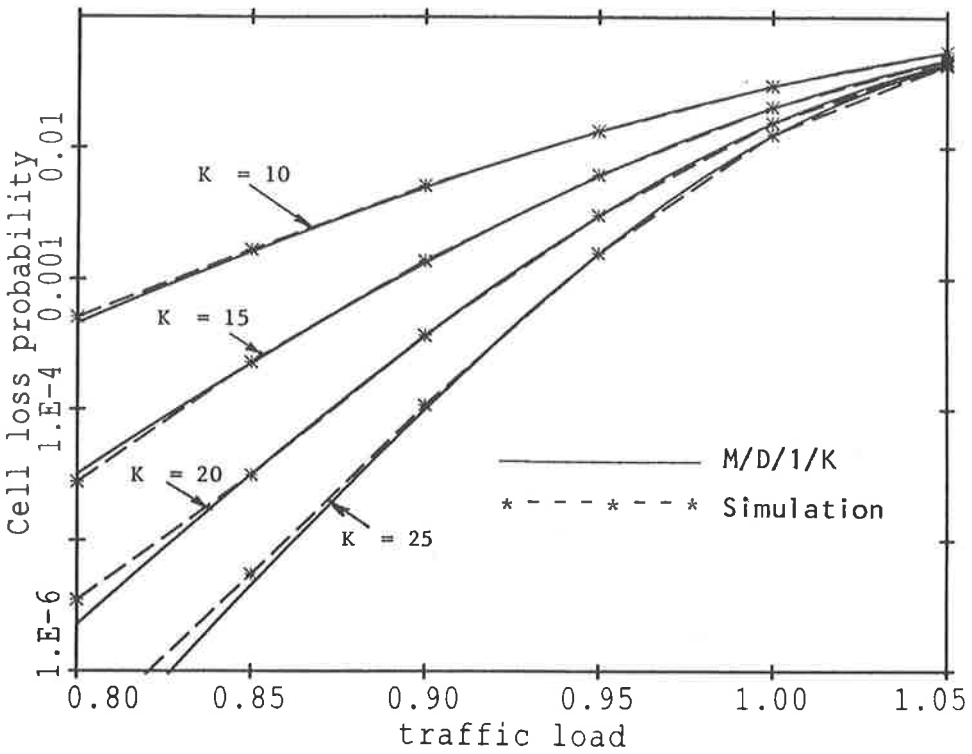


Figure 3.3 Cell loss probability for a single bulk arrival queue.

Average link utilization η	Cell loss probability			
	$K = 15$		$K = 25$	
	Simulation	M/D/1/K	Simulation	M/D/1/K
0.80	2.80E-5±4.1E-6	3.22E-5	3.01E-7±5.6E-8	1.61E-7
0.85	2.32E-4±5.2E-6	2.27E-4	5.50E-6±9.1E-7	4.53E-6
0.90	1.39E-3±5.1E-5	1.33E-3	1.09E-4±6.7E-6	1.01E-4
0.95	6.27E-3±1.2E-4	6.12E-3	1.58E-3±1.8E-5	1.56E-3

Table 3.1 Comparison of simulation of a single bulk arrival queue and the M/D/1/K queue.

Comparison of M/D/1/K queue and simulation results of the bulk arrival queue enables a mean error bars of the simulation in the range from 5% to 10 % of a calculated value of 1.6×10^{-7}, and from 0.5% to 1 % of a calculated value of 1.2×10^{-4}. The main reason is that the simulation runs are limited by the computer resources due to using the Monte Carlo techniques. In this illustrated example, 10^7 simulation cycles were used, while the

maximum repetition period of the computer's random number generator is 10^8. Thus the simulation may be only feasible for relatively high cell loss probabilities.

Secondly, we simulate a tandem queueing network consisting of $Q = 5$ queues. Each queue has $N = 50$ input VPs including VP_a. The constant rate deterministic arrival processes are considered for all VPs with a cell arrival probability $p_a^{(1)}$ for the designated A-stream, and p for the other D-streams. The average link utilization η is given by equation (2.7), and the peak rate η_p of overall cell arrivals is given by equation (2.8). p is kept fixed at 0.018, and $p_a^{(1)}$ is varied, so η varies from 0.75 to 1.

Figure 3.4 illustrates the cell loss probability for the designated A-stream carried on VP_a versus the average link utilization for various buffer sizes. Figure 3.5 presents the simulation results of cell loss-free run distribution $P(0^r|1)$. As a comparison, in Figures 3.4 and 3.5, we also plot the analytic results obtained from chapter 2.

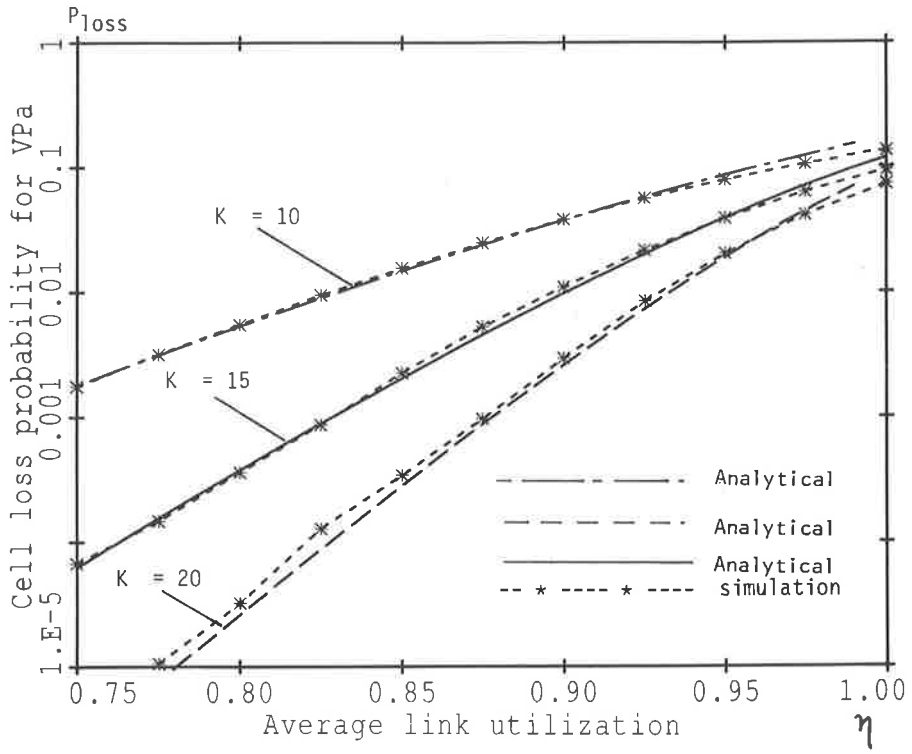


Figure 3.4 Cell loss probability versus the average link utilization for various queueing buffer sizes ($N = 50$ $p = 0.018$).

Average Link Utilization	$K = 10$		$K = 15$		$K = 20$	
η	Analysis	Simulation	Analysis	Simulation	Analysis	Simulation
0.775	3.1E-3	3.2E-3±3.4E-5	1.6E-4	1.5E-4±6.7E-6	7.6E-6	1.1E-5±2.6E-6
0.800	5.4E-3	5.5E-3±1.7E-4	3.8E-4	3.6E-4±1.2E-5	2.6E-5	3.3E-5±6.7E-6
0.825	9.0E-3	9.6E-3±3.8E-4	8.9E-4	8.7E-4±8.7E-5	8.8E-5	1.3E-4±3.6E-5
0.850	1.5E-2	1.6E-2±5.2E-4	2.0E-3	2.3E-3±1.4E-4	2.8E-4	3.4E-4±5.4E-5
0.875	2.4E-2	2.5E-2±7.5E-4	4.6E-3	5.3E-3±7.6E-4	8.7E-4	9.6E-4±7.8E-5
0.900	3.8E-2	3.8E-2±3.1E-4	9.8E-3	1.1E-2±1.2E-3	2.6E-3	2.9E-3±2.4E-4
0.925	5.8E-2	5.7E-2±1.2E-3	2.0E-2	2.2E-2±1.0E-3	7.3E-3	8.4E-3±1.1E-3
0.950	8.6E-2	7.9E-2±4.7E-3	6.9E-2	6.4E-2±5.5E-3	1.9E-2	2.0E-2±2.0E-3
0.975	0.12	0.11±5.6E-3	7.1E-2	6.9E-2±1.7E-3	4.5E-2	4.2E-2±3.5E-3

Table 3.2 Comparison of analysis and simulation.

The simulation results shown in Figure 3.4 indicate that extending the queueing buffer size can reduce the cell loss probability for an individual virtual path. For example, when $\eta = 0.8$, if the buffer size increases from $K = 10$ to $K = 20$, then the cell loss probability will be reduced from 5.52×10^{-3} to 3.25×10^{-5}. However, when traffic load increases, the improvement becomes smaller. For example, when $\eta = 0.95$, increase of buffer size from $K = 10$ to $K = 20$, the cell loss probability is only reduced from 7.9×10^{-2} to 2.04×10^{-2}.

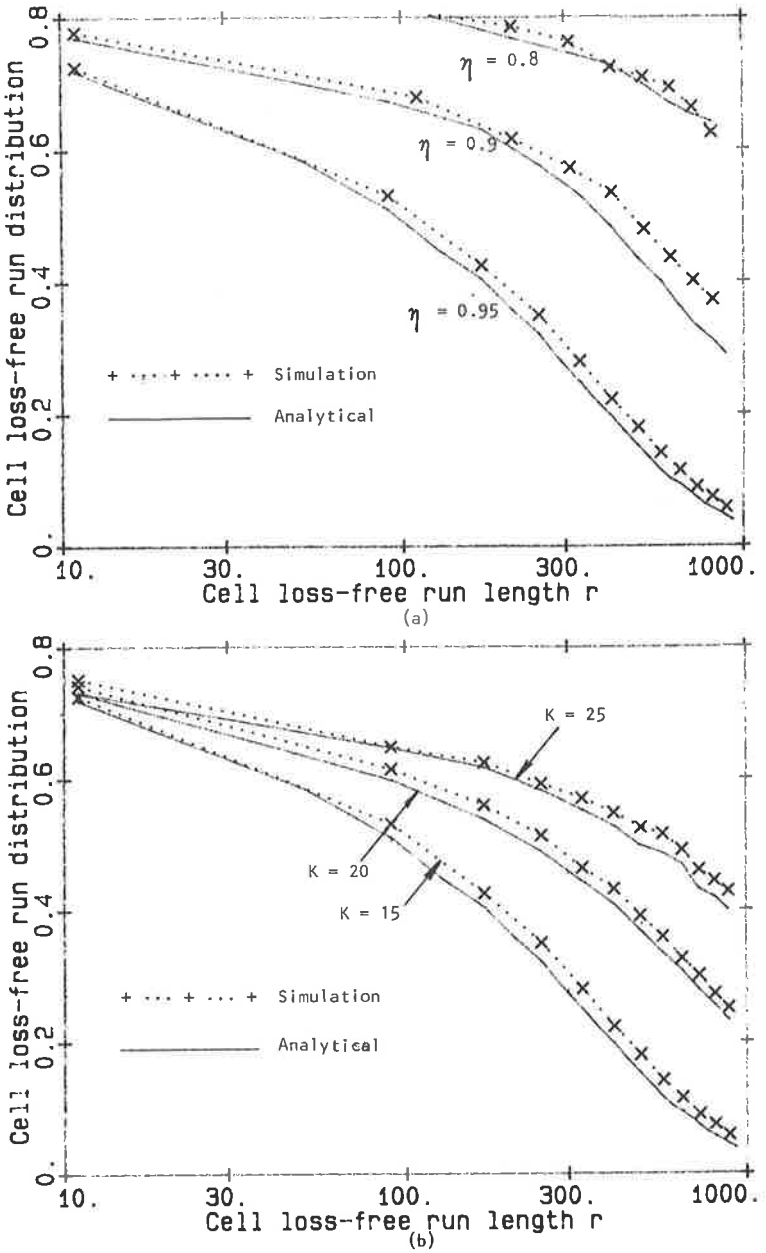


Figure 3.5 Cell loss-free run distribution (N=50).

(a) Cell loss-free run distribution for different average link utilization. ($K = 15$)

(b) Cell loss-free run distribution for different queueing buffer sizes. ($\eta = 0.95$)

Loss-free run distance r	$K = 15$		$K = 20$	
	Analysis	Simulation	Analysis	Simulation
10	0.718	0.723±0.094	0.733	0.738±0.013
90	0.508	0.534±0.027	0.595	0.608±0.025
180	0.405	0.423±0.019	0.547	0.565±0.019
270	0.326	0.355±0.028	0.481	0.506±0.036
540	0.155	0.183±0.03	0.374	0.388±0.015

Table 3.3 Comparison of analysis and simulation for $P(0^r|1)$

The results shown in Figure 3.5 indicate that extending the buffer size can significantly increase the loss-free runs between the burst losses even for the high link utilization. For example, when $\eta = 0.95$, and using a buffer of size $K = 15$, we have $P(0^{300}|1) = 0.213$, i.e., 21.3% of loss-free intervals between the losses are greater than 300 consecutive cells. However, in the same situation but using a buffer of $K = 25$, 63.4% of loss-free intervals between the losses are greater than 300 consecutive cells. This may provide more opportunities for using error correction coding techniques to improve the performance. We note that although increase of buffer size can significantly increase the length of loss-free ranges between the losses, but the burst loss length is also increased. Thus we must carefully study the other important higher order distribution function such as cell loss pattern distribution.

Figure 3.6 shows the simulation results of cell loss pattern distribution $P(u, v)$. As an approximation, in Figures 3.6, we also plot the results obtained from chapter 2.

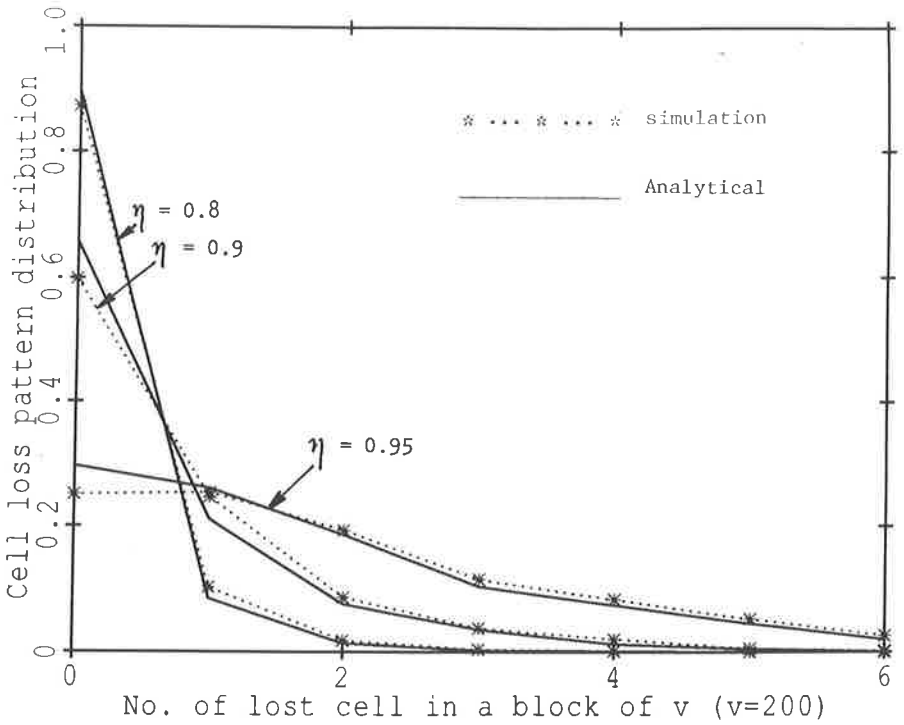


Figure 3.6 Cell loss pattern distribution (N=50, K = 15).

3.5.2 Statistics of cell loss with bursty arrival traffic

In this section, our studies are mainly devoted to the performance for a VP with bursty traffic and the relative statistics of cell loss due to congestion. We simulate a tandem queueing network consisting of $Q = 5$ queues, where each queue has $N = 5$ input VPs including VP_a. The homogeneous burst-silence sources with different burst length as described in section 3.2 are considered for all VPs. In the simulation, since the interdeparture process has been used to observe the designated A-stream carried on VP_a as it goes through the network, the queue transient effects have been taken into account.

Recall the characteristics of the mixture of homogeneous burst-silence traffic discussed in section 3.2.3 as follows:

- If the peak rate of total arrivals is less than the output link capacity, (i.e. without overload), then loss belongs to the cell level. In this case, if the burst length is much greater than the queueing buffer size, then the queue transient effects could be

ignored. Otherwise, the queue transient effects must be taken into account.

- A burst-silence source with the average burst length greater than the inter-cell time T_b slots but less than the queueing buffer of K slots is referred to as a short burst source. In this case, the cell loss is approximated as the loss at the cell level. However, the queue transient effects must be taken into account.

- A burst-silence source with the average burst length greater than queueing buffer of K slots is referred to as a long burst source. In this case, if the peak rate of total arrivals exceeds the link capacity, then loss belongs to the burst level.

Firstly, we consider the cell loss probability and the relative effects of the burst lengths and queueing buffer sizes.

Figure 3.7 shows the cell loss probability for VP_a versus the average link utilization η for various queueing buffer sizes with short burst traffic. A comparison of the cell loss probability versus the average link utilization for random traffic, short bursts, and long bursts with different queueing buffer sizes is illustrated in Figure 3.8. In Figures 3.7 and 3.8, the short burst traffic is the mixtures of homogeneous burst-silence sources with a mean burst duration of $T_{on} = 10$ slots, and a mean silence duration of $T_{off} = 2$ slots. For the long burst traffic, we consider the mixtures of homogeneous burst-silence source with a mean burst duration of $T_{on} = 100$ slots, and a mean silence duration of $T_{off} = 20$ slots. Both the short burst and the long burst sources have the same source burstiness 1.2. The average link utilization η is given by equation (3.8), and peak rate of overall cell arrivals η_p is given by equation (3.9). Both the short burst source and the long burst source have the same peak rate λ which varies from 0.18 to 0.24, so η varies from 0.75 to 1. For the random traffic, we consider the simulation of the same tandem queueing network consisting of $Q = 5$ queues, and each queue with $N = 5$ constant rate deterministic arrival sources which have the same arrival rate. The average link utilization η for the random traffic is given by equation (3.24), and the peak rate of overall cell arrivals is given by equation (3.25).

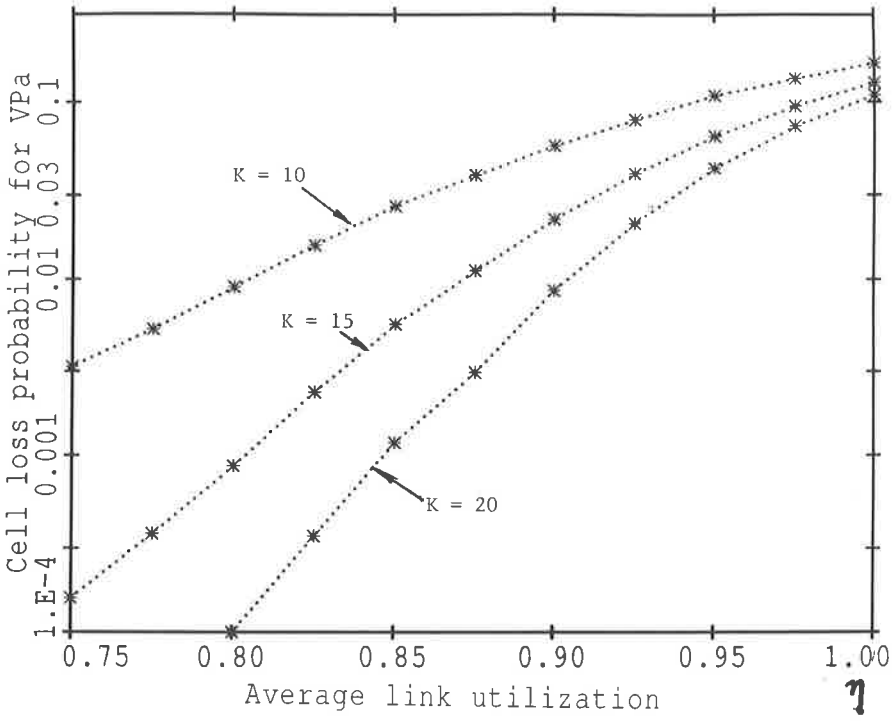


Figure 3.7 Cell loss probability for VP_a with short bursts ($N = 5$, $Q = 5$, $T_{on} = 10$ slots, $T_{off} = 2$ slots).

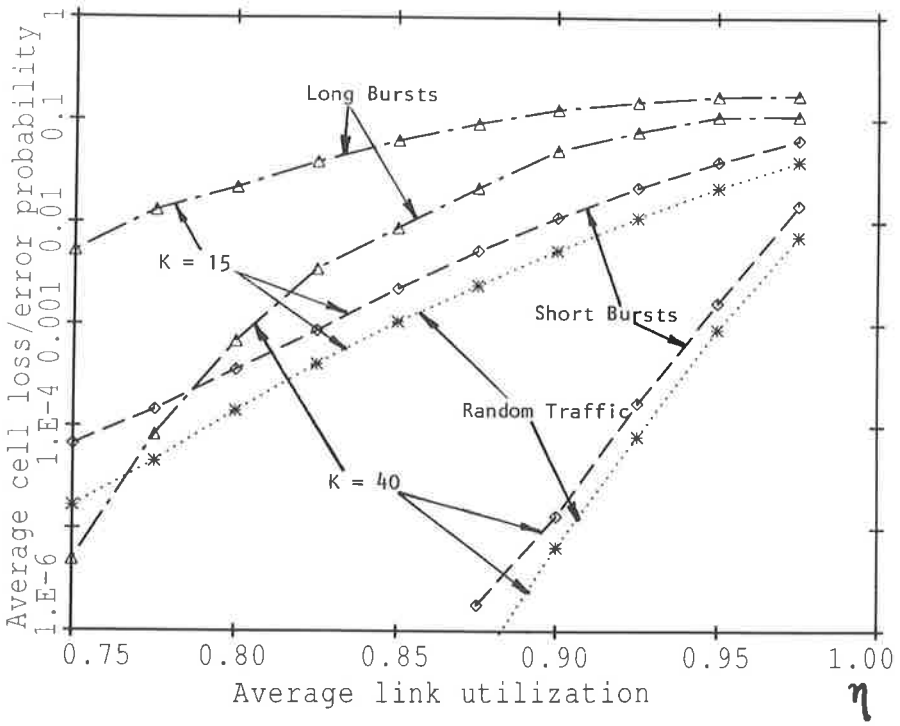


Figure 3.8 Cell loss probability for VP_a with long burst traffic, short burst traffic, and random traffic.

From equations (3.8) and (3.9) we know that, in Figure 3.8, when $\eta \leq 0.83$ the corresponding peak rate of overall cell arrivals is $\eta_p \leq 1$, (i.e., without overload). In this case, the cell loss for random traffic, short bursts, and long bursts belong to the cell level, and extending the queueing buffer size can significantly reduce the cell loss probability. For example, when $\eta = 0.75$, if buffer size increases from $K = 15$ to $K = 40$, then for the long bursts, cell loss probability is reduced from 5.29×10^{-3} to 4.99×10^{-6}. Under the same conditions, obviously, for the short bursts or random traffic, the improvement will be more significantly.

However, when $\eta > 0.83$, the corresponded peak rate of overall arrivals will exceed the link capacity. In this case, for both the random traffic and the short bursts, the cell loss still belongs to the cell level, but for the long bursts it belongs to the burst level. Then extending the queueing buffer size can still significantly reduce the cell loss probability for the random traffic or the short bursts. For example, when $\eta = 0.9$ and the queueing buffer is extended from $K = 15$ to $K = 40$, for the random traffic, the cell loss probability is reduced from 5.49×10^{-3} to 6.66×10^{-6}, and for the short bursts, the improvement is from 1.11×10^{-2} to 1.33×10^{-5}. Because the queueing buffer is much greater than the burst length of both short bursts and random traffic (a fixed rate traffic is equivalent to a bursts-silence source with the mean burst length in cells equal to 1, see section 3.2.3), then queues can absorb short bursts, and store cells for transmission when output link is free. However, when the long bursts is involved, the improvement would be very limited. For example, when $\eta = 0.9$ and queueing buffer size increases from $K = 15$ to $K = 40$, for the long bursts, the improvement is only from 9.27×10^{-2} to 5.03×10^{-2}. On the other hand, the cell loss is also sensitive to link utilization. When the average link utilization increases, the effects of extending queueing buffer size becomes smaller.

Secondly, we consider the cell loss-free distribution $P(0^r|1)$. Figure 3.9 illustrates the cell loss-free run distribution with short bursts for different average link utilization and buffer sizes. The parameters of the short burst traffic are the same as those considered in Figure 3.8.

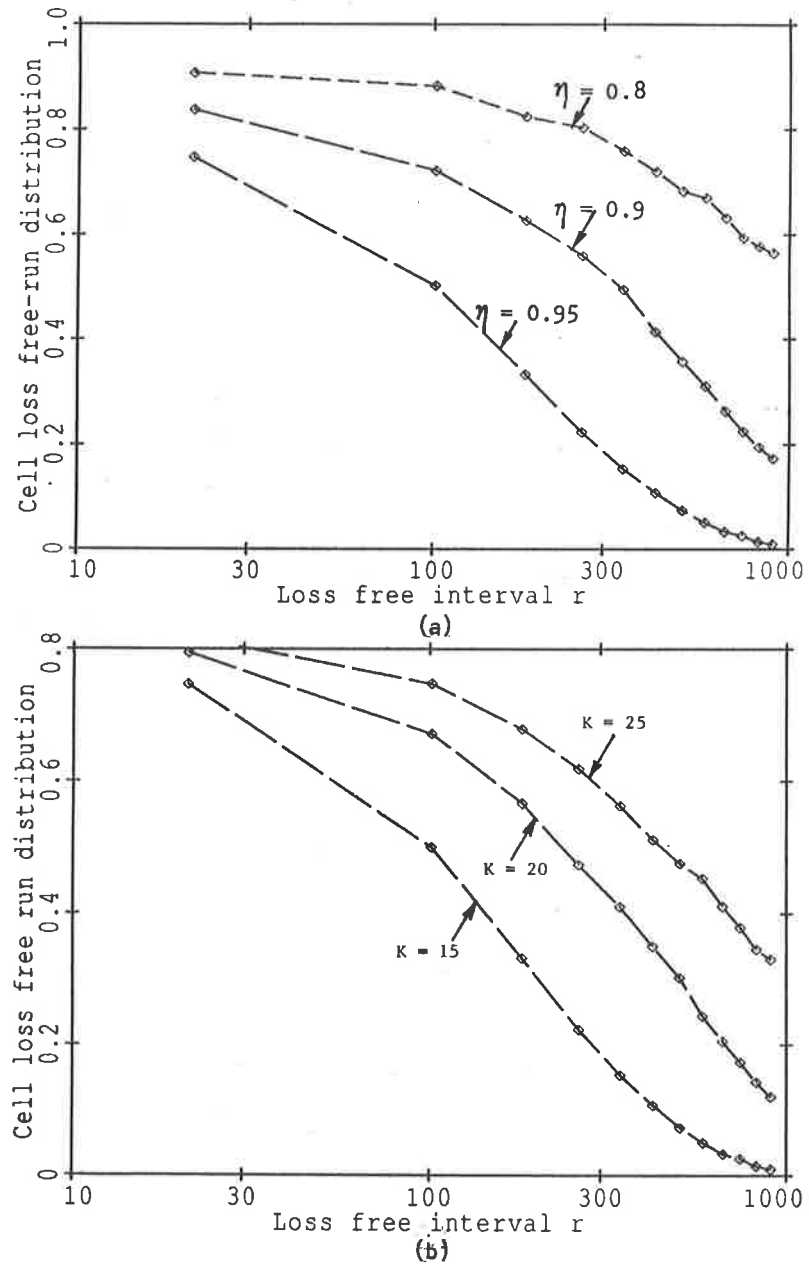


Figure 3.9 Cell loss free run distribution with short bursts (N=5, Q=5, T_{on} = 10 slots, T_{off} = 2 slots).

(a) Cell loss-free run distribution for different average link utilization. (K = 15)

(b) Cell loss-free run distribution for different queueing buffer sizes. (η = 0.95)

The results in Figures 3.9 show that $P(0^r|1)$ is sensitive to traffic load. However, extending buffer size may also significantly increase the loss- free ranges between the losses. For

example, when $\eta = 0.95$ and queueing buffer size increases from $K = 15$ to $K = 25$, The probability that cell loss-free run of length greater than 300 consecutive cells, $P(0^{300}|1)$, increases from 0.18 to 0.54. By contrast, for random traffic as shown in Figure 3.5, this probability is slightly higher and shows a similar trend.

Figure 3.10 illustrates the cell loss-free run distribution for long bursts with different average link utilization. The parameters of the long burst traffic are the same as those used in Figure 3.8.

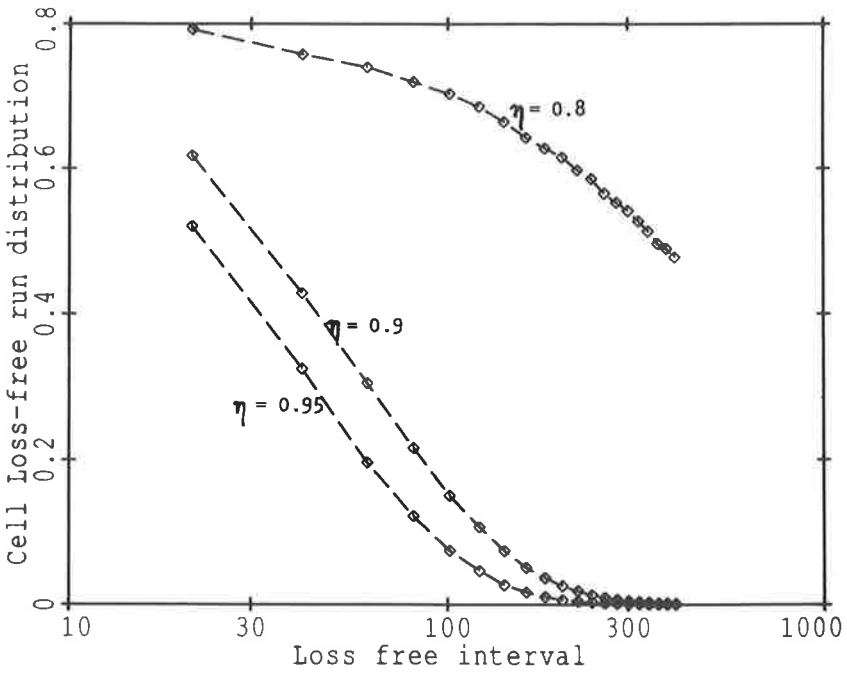


Figure 3.10 Cell loss-free run distribution for different average link utilization ($K = 15$, $T_{on} = 100$ slots, $T_{off} = 20$ slots).

As shown in Figure 3.10, when $\eta = 0.8$, the corresponding peak rate of overall cell arrivals is less than the link capacity. In this case, the cell loss belong to the cell level, and 70.3% of loss-free ranges between the losses are greater than 100 consecutive cells, i.e. $P(0^{100}|1) = 0.703$. However, when $\eta = 0.9$ or $\eta = 0.95$, the corresponding peak rate of overall cell arrivals exceeds the link capacity, and the cell loss belongs to the burst level. Then the probability of the loss-free runs of length greater than 100 consecutive cells decreases to 0.15 for $\eta = 0.9$, and 7.34×10^{-2} for $\eta = 0.95$. Therefore, for the long bursts,

the cell loss-free run distribution is also sensitive to the traffic load.

A comparison of the cell loss-free distribution for random traffic, short bursts, and long bursts is illustrated in Figure 3.11. The parameters of short burst traffic and random traffic are the same as those used in Figure 3.8. For the long burst traffic, we consider two different situations: i) the burst length is much greater than the queueing buffer size, that is, the burst-silence source with the mean burst duration of $T_{on} = 100$ slots, and the mean silence duration of $T_{off} = 20$ slots. ii) burst length is just greater than the queueing buffer size, that is, the burst-silence source with the mean burst duration of $T_{on} = 30$ slots, and the mean silence duration of $T_{off} = 6$ slots (In this example, the queueing buffer is 25). Both the short bursts and the long bursts have the same source burstiness 1.2. The average link utilization $\eta = 0.95$ which is given by equation (3.8).

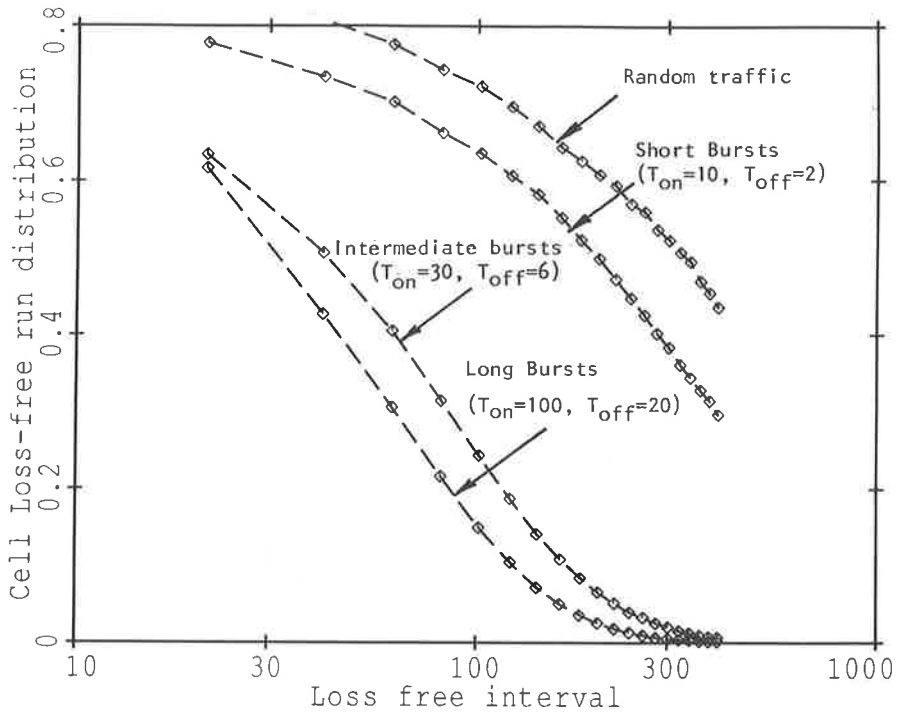


Figure 3.11 Cell loss free run distribution for different burst lengths ($K = 25$, $\eta = 0.95$).

In Figure 3.11, for $\eta = 0.95$ the corresponded peak rate of overall cell arrivals is $\eta_p = 1.05$ which exceeds the link capacity. In this case, the cell loss-free run distribution $P(0^r|1)$ is sensitive to the burst length. According to the justification made at the beginning of this section, for both random traffic and short bursts, the cell loss belongs to the cell level, and

$P(0^{100}|1)$ is 0.72 and 0.64 for the random traffic and short bursts, respectively. However, for a long bursts with $T_{on} = 30$ slots which is just greater than queueing buffer size and $T_{off} = 6$ slots , the cell loss free run decreases to $P(0^{100}|1) = 0.243$. When the burst length increases to $T_{on} = 100$ slots and $T_{off} = 20$ slots, the cell loss-free run distribution becomes worse with a value of $P(0^{100}|1) = 0.15$. Because when the burst length is greater than the queueing buffer size, the loss belongs to the burst level.

Finally, we consider the cell loss pattern distribution $P(u, v)$. Figure 3.12 shows the cell loss pattern distribution $P(u, v)$ in a block of $v = 200$ cells for short bursts with mean burst length of $T_{on} = 10$ slots and mean silence length of $T_{off} = 2$ slots. The average link utilization is given by equation (3.8).

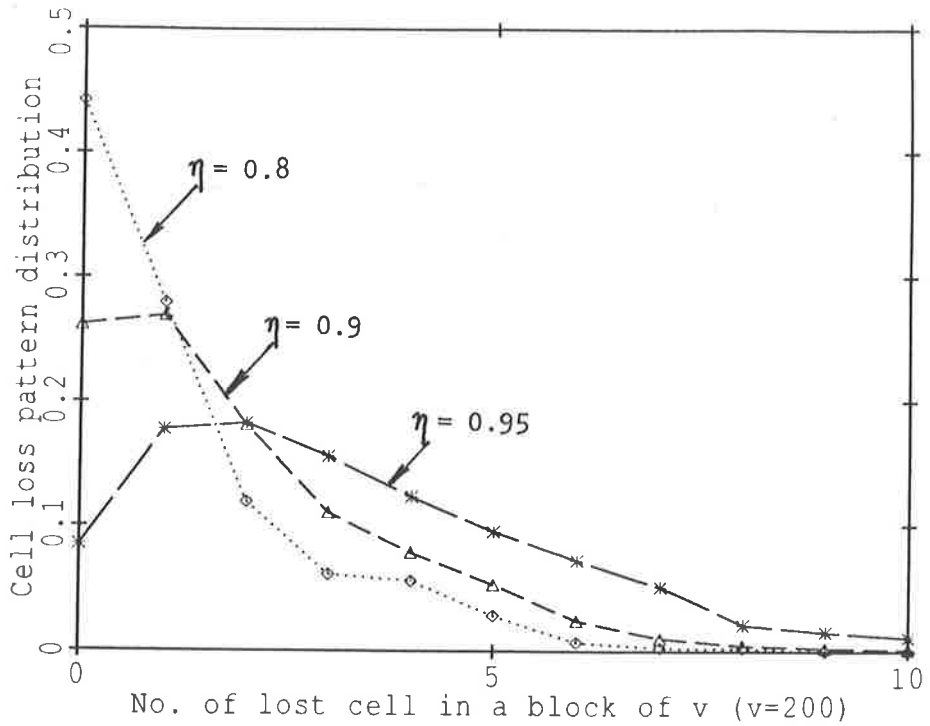


Figure 3.12 Cell loss pattern distribution $P(u, v)$ with short burst traffic for different link utilization (N=5, Q=5, $T_{on} = 10$ slots, $T_{off} = 2$ slots, $K = 15$, and $v = 200$ cells).

Figure 3.13 shows the cell loss pattern distribution $P(u, v)$ in a block of $v = 200$ cells for long bursts with mean burst length of $T_{on} = 100$ slots and mean silence length of $T_{off} = 20$ slots.

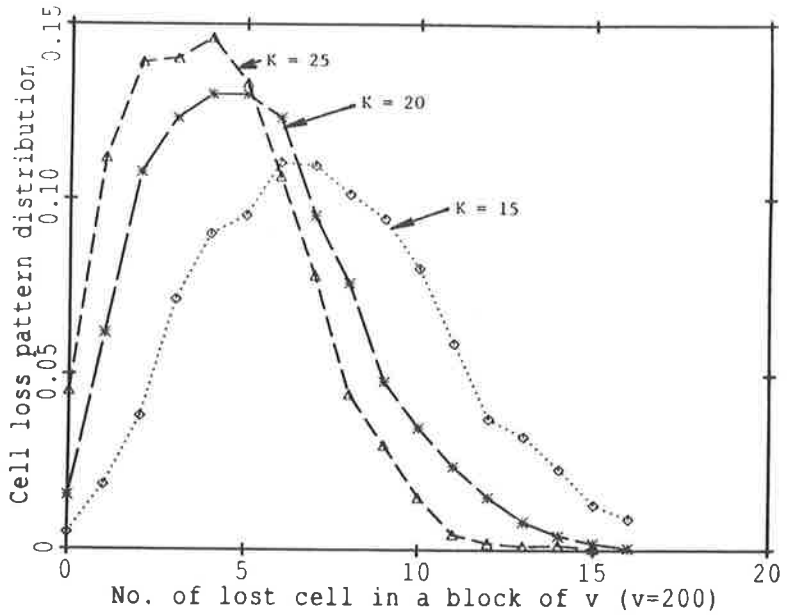


Figure 3.13 Cell loss pattern distribution with long bursts for different buffer size ($N = 5$, $Q = 5$, $T_{on} = 100$ slots, $T_{off} = 20$ slots, $v = 200$ cells, and $\eta = 0.95$).

The results in Figure 3.13 show that the cell loss pattern distributions $P(u, v)$ are affected by changing of buffer sizes, but the average cell loss probability is only slightly affected. For example, for a given average link utilization $\eta = 0.95$, the average cell loss rate versus the buffer size is listed as follows:

Buffer size K	$\overline{N}_A$ in a block of $v = 200$ cells	Average cell loss probability
15	6.29	3.1×10^{-2}
20	5.23	2.6×10^{-2}
25	4.38	2.2×10^{-2}

This presents that at the burst level the bursty loss patterns may affected by changing of buffer size but not the cell loss rates.

3.5.3 Discussion

In this section, a virtual path through an ATM network is modelled as a tandem queueing network. Two types of cell arrival processes are allowed to enter the network: i) mixtures of discrete-time constant rate deterministic arrival processes, and ii) mixtures of homogeneous discrete-time burst-silence processes. Our studies have been mainly devoted to the later to study the statistics of cell loss for a VP with bursty arrival traffic subject to information loss recovery in ATM networks, since the exact queueing analysis of a tandem queueing network with burst arrivals is difficult.

In the simulation, we consider the random traffic to be a fixed rate random arrival process, the short burst traffic to be a burst-silence source with mean burst length less than the queueing buffer size, and long burst traffic to be a burst-silence source with mean burst length greater than the queueing buffer size.

Cell loss caused by congestion can be decomposed into two different levels: cell level and burst level. The performance of a VP with random traffic, or short bursts, or long bursts without overload belong to the cell level. The performance of long bursts with overload belongs to the burst level.

The results of cell loss statistics are summarized as follows.

1. At the cell level, extending the queueing buffer size can reduce the cell loss probability for an individual VP, but the improvement becomes smaller as the average link utilization toward to the link capacity.

2. At the cell level, extending queueing buffer size can significantly increase the cell loss-free runs between the burst losses, even for a high average link utilization. This may provide a possibility to use error correction techniques combined with suitable queueing buffer size to improve the network performance at the cell level. We will discuss this in chapter 4.

3. At the burst level, using larger buffer will not significantly to reduce the cell loss rate because of the long bursts involved.

4. At the burst level, extending the queueing buffer size may also increase the loss-free

intervals between the burst losses, but this improvement is very limited. On the other hand, the cell loss pattern distribution has shown that the cell loss occur in bursts. For the purpose of cell loss recovery at the burst level, some other countermeasures such as interleaving/deinterleaving techniques may be used to associate the forward error recovery coding techniques. This is to break up and distribute the burst loss patterns so that to the decoder they appear randomly distributed [32]. However, the capacity of the interleaver/deinterleaver is limited by the hardware implementation and the relative delay. If long burst length leads to cell loss in very long bursts, then the forward error coding scheme will probably be no longer work. However, the burst length can be controlled using the cell policing mechanism. We will consider this in chapter 5.

5. The cell loss is sensitive to both traffic load and burst length. When the peak rate of overall cell arrivals is less than the link capacity, the cell loss belongs to the cell level. In this case, extending the queueing buffer size can significantly improve the performance for random, short burst, and long burst traffic. However, when the peak rate of overall cell arrivals exceeds the link capacity, the cell loss becomes sensitive to the burst lengths. For both the random traffic and the short bursts, their burst length is less than the queueing buffer sizes (the random traffic can be equivalent to a burst with the average burst length equal to 1 slot), and the cell loss belongs to the cell level. In this case, extending the queueing buffer size can still significantly reduce the cell loss rate. However, as the burst length increases, the improvement becomes smaller. Under the same condition, for the long bursts, the cell loss belongs to the burst level, and the effects of extending the queueing buffer size to reduce the cell loss rate the will be very limited. To avoid the burst level congestion, the cell policing mechanism should control the burst length to be less than the queueing buffer size.

Chapter 4

RECOVERY OF CELL LOSS AT THE CELL LEVEL

Summary

In this chapter, we consider the recovery of cell loss at the cell level using a forward error control coding scheme. This scheme is implemented as a two-dimensional code, in which error control for a single cell is referred to as being along the x-axis, while error control among a block of consecutive cells is performed in direction of the y-axis. In the x-axis direction, a class of high rate codes is considered for cell error detection. In the y-axis direction, the erasure correction decoding is used for lost and errored cell recovery. This forward error recovery coding scheme can be used to control the cell loss for either Virtual Path Connections (VPCs) or Virtual Channel Connections (VCCs) [66]. In order for this scheme to be possible, a cell sequence number is required for detection of cell loss. This cell sequence number can be incorporated into the *Segmentation and Reassembly* (SAR) sublayer located in the ATM adaptation layer (AAL) in the payload field [66]. We note that this two-dimensional coding scheme can be also used for recovery of errored cells. In this chapter, we will also take this into our consideration.

The performance with coding is analyzed in the presence of transmission errors and cell loss due to network congestion at the cell level. As new results, we will show that if the x-axis code and y-axis code are chosen properly, high reliability can be obtained even for

high link utilization. We illustrate several specific coding schemes, in which *small distance* codes [9] with $d_{min1} < 3$ are used for the x-axis error detection code, while some efficient high rate binary cyclic codes, or non-binary Reed-Solomon codes are considered for the y-axis code. The bandwidth efficiency, delay, cell loss/error probability, and capability of recovering from long cell loss bursts are evaluated. We will show that forward error correction coding techniques can be used as a supplement to the conventional use of congestion control techniques at the cell level to significantly improve the performance of ATM network suffering from short-term congestion at the cell level. This extends the range and condition over which the ATM network can be safely exploited. Finally the methods for design of suitable coding schemes, utilizing the cell loss statistics, are discussed.

4.1 Introduction

A two-dimensional forward error protection scheme for lost cell recovery and cell error correction at the cell level has been proposed by the author in 1989 [8]. This scheme can be used with end-to-end virtual channel connections (VCCs). It can also be used to protect multiple services by means of edge-to-edge error protection for virtual path connections (VPCs). We assume that the cells carried on a virtual path or virtual channel all have the same priority level. It is expected that cells with different priority will be carried by different virtual paths or virtual channels [22][72].

In this chapter, the term "forward error recovery" means forward recovery of cells with transmission error and cells lost due to congestion at the cell level. For an optical fiber link with a bit error rate of 10^{-9}, the probability of a cell information field of 384 bits contains transmission errors is 4.24×10^{-7} (without error protection). However, the situation under our consideration is the performance of a VP modelled as a tandem queueing network with small size queues and high link utilization. In this case, the results shown in Figures 3.4 and 3.7 have shown that the cell loss probability due to congestion at the cell level is much greater by comparing it to the probability that cells contain transmission errors. Therefore, our forward error protection coding scheme is mainly focussed on the

recovery of cell loss gainst the congestion at the cell level.

As discussed in chapters 2 and 3, cell loss due to buffer overflow often occurs in bursts, even for random traffic. We will show that the forward error recovery coding schemes work best for recovery of burst cell loss at the cell level. Our attention is focussed on a designated traffic stream with a forward error protection coding scheme as it passes through the network along a virtual path. The application of coding for cell loss control at the burst level will be considered in chapter 5.

In section 4.2, the two-dimensional coding scheme is first described. Then in sections 4.3 to 4.6, the implementation of the two-dimensional coding scheme is studied. The performance of the scheme is analyzed in the presence of transmission error and cell loss in section 4.7. The bandwidth efficiency, delay, cell loss/error probability and capability of recovering from long cell loss bursts are evaluated. In section 4.8, several specific illustrative schemes are considered. The coding gains are evaluated and compared with the uncoded cases for both random and short burst traffic. We show that if the two-dimensional code is chosen carefully, satisfactory results can be obtained even for a high link utilization and long cell loss bursts. In section 4.9, we describe the methods of choosing suitable coding schemes utilizing cell loss statistics. Finally, in section 4.10, ARQ scheme as an approach of cell loss control at the cell level is studied. A comparison of the two-dimensional coding scheme and ARQ scheme is also discussed.

At the cell level, the common technique for dealing with short term cell congestion is to provide buffers of sufficient size in the access multiplexers and switches. These buffers will absorb sudden simultaneous arrivals, and store cells for transmission when the link is free. As the total rate of traffic arrival increases toward the link capacity, buffers will often overflow. This situation may be partly overcome by increasing the size of the buffer. However, large buffers increase both cell delay and the variation in cell delay. The delays will also increase proportionally to the length of virtual path through the network. This can cause difficulties for real time services which require synchronization between the transmitter and receiver. Therefore, a large elastic buffer may be needed at the receiver end to remove the cell delay jitter.

In this chapter, we consider reduction of cell loss rate by the two-dimensional cod-

ing scheme combined with small buffers rather than increasing buffer size. The two-dimensional coding scheme also introduces delays, mainly caused by the y-axis decoding, but the delay is fixed. The detail will be discussed in the later sections. The effects of using forward recovery coding for cell loss control are compared with the effects of using large buffering. It shows that two-dimensional coding schemes provide an extremely powerful mechanism for reducing information loss, and extend the range of loads and conditions over which ATM networks can be safely exploited.

As an alternative to a two-dimensional scheme, block codes combined with interleaving may also be used for lost cell recovery [49]. The cell header cannot be interleaved/deinterleaved since it contains virtual path identification (VPI) which must be processed along the way through the network. It would therefore be necessary to code and interleave the information field before the cell header is incorporated into the cell at the transmission end. This would need to be implemented with end-to-end virtual channel connection (VCC). For a cell with an information field of 48 octets (384 bits) [1], the interleaving could be done by arranging a block of 384 bits $\times$ 384 bits into 384 rows of a rectangular array and then transmitting them column by column (384 bits) as a cell. At the receiving end, it would be necessary to perform deinterleaving after the cell header is removed from the cell. If the transmitted cells are interleaved, and after transmission, the received cells are deinterleaved, then each single lost cell would appear as a single error bit in a deinterleaved cell, and it could be corrected by a simple forward error correction code with parity redundant bits located in the AAL layer in the payload field. Since the number of parity redundant bits in the AAL layer reserved for information field error control is limited (10 bits are recommended by CCITT) [66], the capacity of recovering from lost cells would be also limited. For example, using an CRC-10 code [37] only can recover one error bit in the information field of a received cell. That is, for an interleaving/deinterleaving block of 384 cells, it only can recover a single lost cell. To increase the capacity of recovering from long burst loss, interleaving/deinterleaving size would be much greater than a block of 384 cells. This may involve a large delay at both the transmission end and the receiver end.

Ohta et al. recently presented a forward loss recovery coding scheme [84] for data transfer

application with virtual path connection in ATM networks. A block consists of Γ cells arranged in Θ rows. Each row of Γ cells contains one redundant cell for loss detection, and each column of Θ cells contains one redundant cells for loss recovery. Therefore, $\Gamma + \Theta - 1$ cells are overhead for each block. A single burst loss of $\Gamma - 1$ cells in each block may be recovered by this scheme. A buffer of $\Gamma \times \Theta$ cells is also required to be set up at the receiver end to assist the decoding. As mentioned in [84], this forward loss recovery coding scheme may be only suitable for low link utilization due to the heavy overhead. For example, as mentioned in [84], if using a coded block of 16×20 cells, then 15% overhead is introduced by the coding scheme. In this case, when the link utilization is greater 0.76 (based on a M/D/1 queue), the cell loss rate of the coded case is worse than that of uncoded case.

By contrast, our two-dimensional scheme will be shown to have significant performance advantages compared to the use of conventional block codes with interleaving as proposed in [49][84]. For example, the high transmission rates expected in ATM networks will require the encoding and decoding to operate at very high speed. In this respect, the two-dimensional scheme will have advantages, because encoding and decoding can be done in parallel, resulting in encoders and decoders which can be fast enough.

4.2 Two-dimensional coding scheme for forward error recovery

A diagram of the two-dimensional forward error protection scheme is shown in Figure 4.1. Figure 4.1(a) shows the scheme used as an end-to-end function at the virtual channel connection (VCC) level. This approach would be more suitable for services which require high throughput, such as bursty data, video, and image. Figure 4.1(b) shows the forward error protection scheme used as an edge-to-edge approach (between concentrators) at the virtual path connection (VPC) level [1][22][66]. The error protection includes the ATM adaptation layer in the payload field but not the cell headers, because the virtual path identifier (VPI) contained in the cell header will be modified in each switching node along the virtual path. However, according to current CCITT standards, the cell header will contain its own error protection coding schemes in the ATM layer [66].

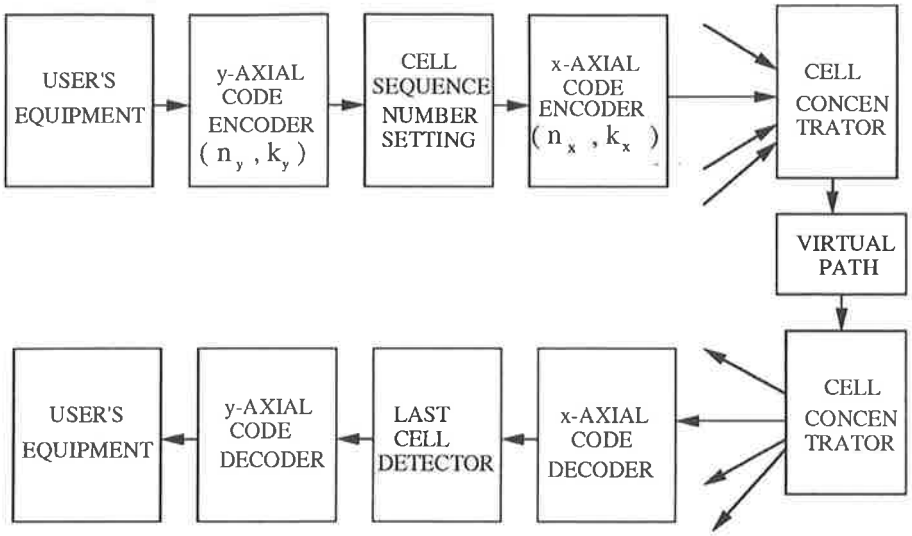


Figure 4.1 (a) End-to-end forward error protection for the virtual channel connections (VCCs).

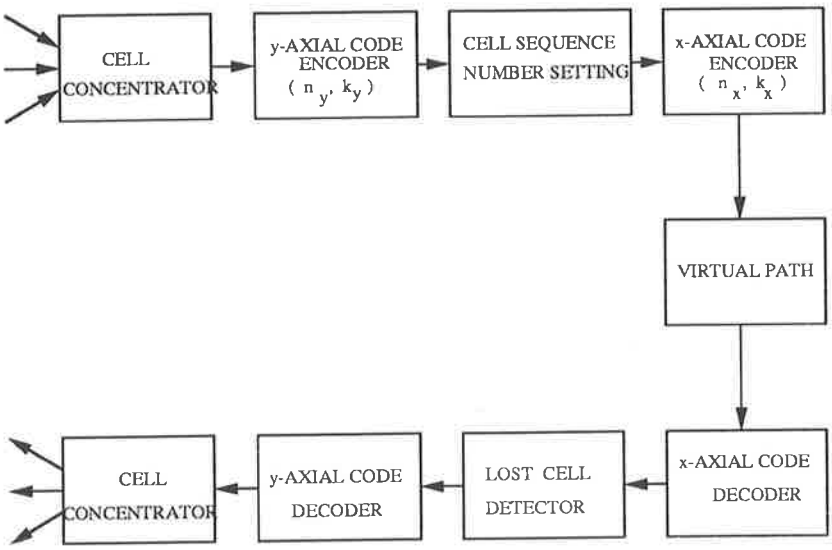


Figure 4.1 (b) Edge-to-edge forward error protection for virtual path connections (VPCs).

If the two-dimensional error protection coding scheme is used as an end-to-end approach for an individual service at the virtual channel connection (VCC) level, then

the coder/decoder will operate at the pair of customer's terminal equipment. The error protection could cover the whole information field including the ATM adaptation layer in the payload field. If error protection is provided for individual virtual paths, the encoding will be done at port controllers at the edges of the network, before the VPI is added into the cell header. The error protection could again cover the whole information field including the ATM adaptation layer (AAL). The error recovery decoding will be done after cell header decoding is done, and the VPI is removed from the cell header, i.e., cells are sorted into individual VP's (or VC's). We will demonstrate later that, within limits (i.e., given sufficient hardware), no processing bottlenecks will occur with this scheme, although some delay will be introduced due to the need for receiver buffering of cells before decoding. This delay however will be fixed. We should note that in an ATM network, cell losses caused by buffer overflow, switching errors, priority discarding, or misrouting, and cell errors caused by transmission errors may appear with different statistics. However, they can share the same error recovery schemes using two-dimensional codes. In this chapter, our attention is focussed on the recovery of cells lost due to congestion at the cell level. The correction of transmission errors contained in the cell information field is also under our consideration.

In order for this two-dimensional scheme to be possible, sufficient bits for cell sequence number and parity check bits for cell error detection must be set aside in the ATM adaptation layer (AAL). According to current versions of CCITT standards, a 4-bit cell sequence number is setup in the *Segmentation and Reassembly* (SAR) sublayer which is located in the AAL in the payload field to detect the cell loss or misinserted cells for an end-to-end connection. Therefore, when the two-dimensional coding scheme as an end-to-end approach for the virtual channel connections, the 4-bit cell sequence number can be directly used for cell loss detection. However, when the two-dimensional coding scheme is used as an edge-to-edge approach for virtual path connections, we must set up a separate cell sequence number in the SAR sublayer. According to the CCITT standard, 10 parity redundant bits are reserved in the SAR sublayer for the protection of transmission errors in the information field [66]. However, in the following section, we will show that in our two-dimensional coding scheme, using a *small distance* code as the x-axis code with only 4 - 6 parity check bits in the SAR sublayer can sufficiently provide satisfactory error

detection mechanism for a standard cell information field. Therefore, at least 4 out of 10 bits reserved in the SAR for error protection are not used in this two-dimensional coding scheme. We will consider to use these unused bits to set up a separate cell sequence number for edge-to-edge connection. The details will be discussed in section 4.5.

Encoding/decoding procedures

Figure 4.2 illustrates a diagram of the encoder. Encoding is performed in three stages as shown in Figure 4.3. First, k_{x1} information bits for a cell (excluding cell header, cell sequence number, and the parity check bits in the AAL) is divided into k_{x1} units of 1 bit each; each unit being simultaneously fed into k_{x1} parallel encoders of the y-axis code C_y. A total of k_y consecutive cells will be encoded in this way to form k_{x1} y-axis code words, each of length n_y units, denoted as $\overline{r}_y$. After parallel to series conversion they form n_y cells altogether as illustrated in Figure 4.5. This consists of k_y information cells and $n_y - k_y$ parity redundancy cells.

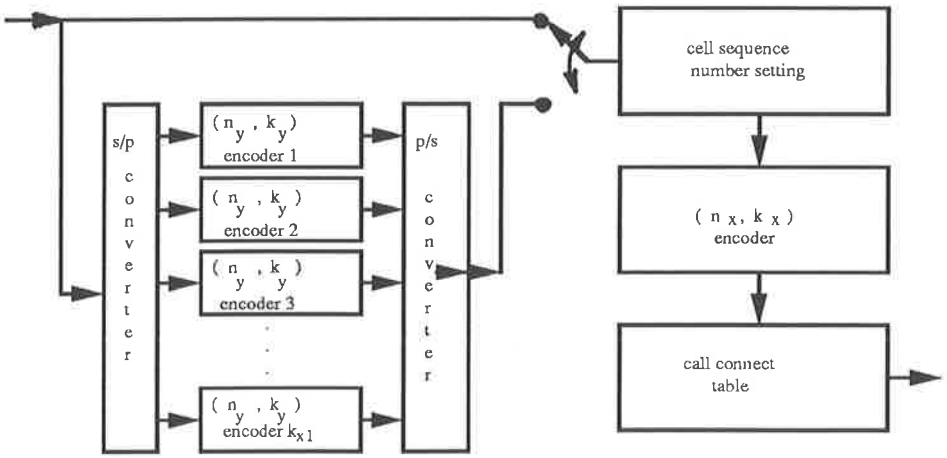


Figure 4.2 A diagram of the encoder.

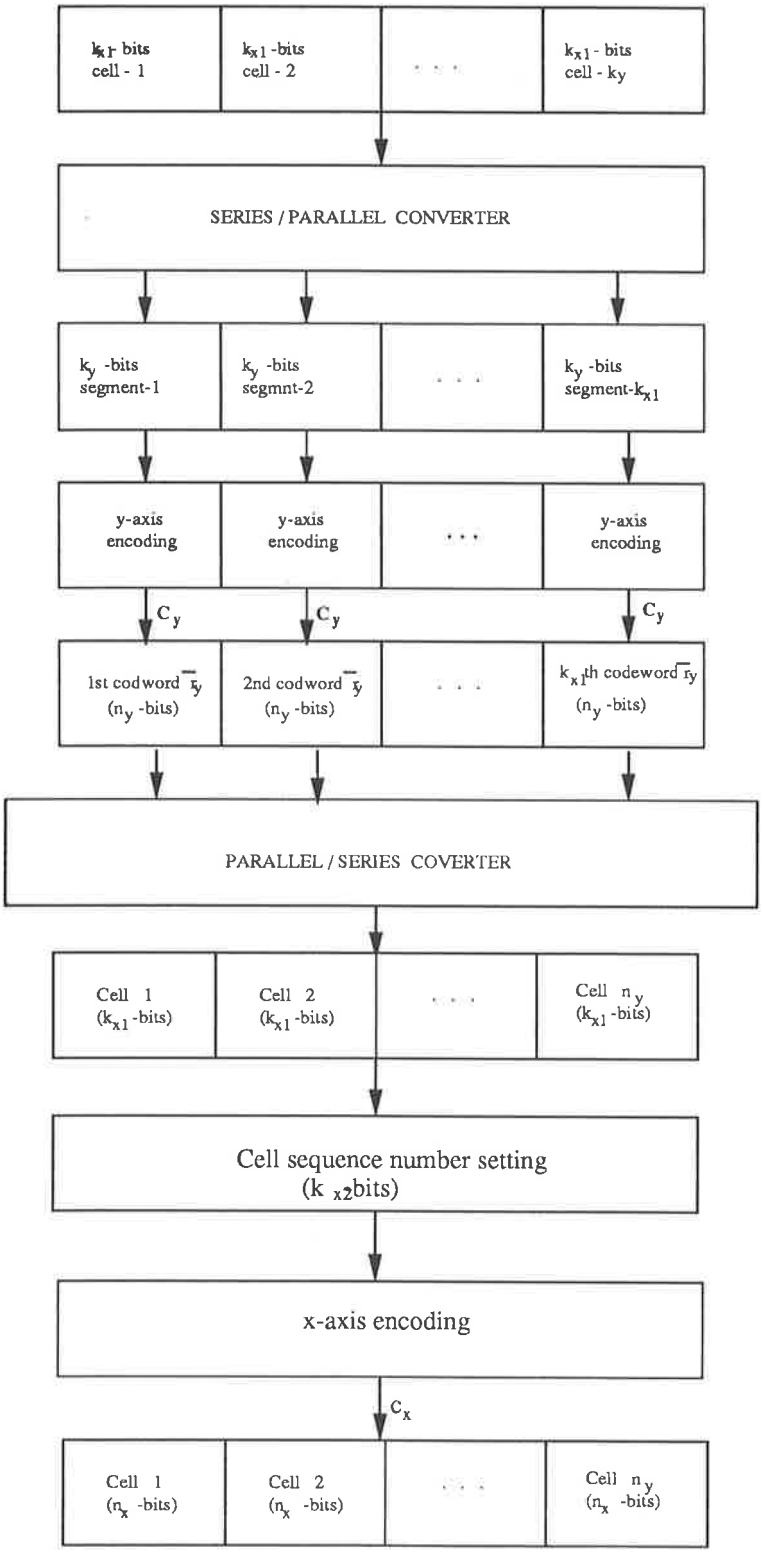


Figure 4.3 Encoding procedures

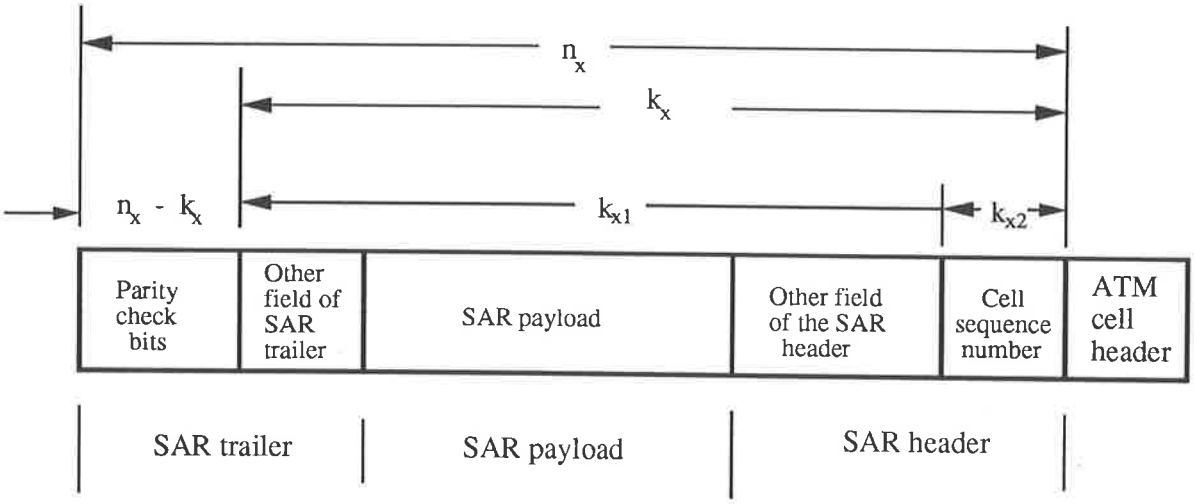


Figure 4.4 The cell with forward error protection coding.

Next, a cell sequence number of k_{x2} bits is incorporated into each of the n_y cells at the output of the y-axis code encoders. Thus the information field of each cell becomes $k_x = k_{x2} + k_{x1}$ bits long.

At the third stage of encoding, each cell of k_x bits is encoded in terms of the x-axis code C_x to form an n_x-bit code word, denoted $\overline{r}_x$, which now includes $n_x - k_x$ parity check bits. Note that both the cell sequence number of k_{x2} bits and $n_x - k_x$ parity check bits are incorporated into the SAR sublayer located in the AAL in the payload field. Thus from the k_y cells of k_{x1} bits at the input of the y-axis code encoders, a sequence of n_y cells of n_x bits will appear at the output of the x-axis code encoder. It should be noted that, after k_y information cells are delivered onto a virtual path (or virtual channel), $n_y - k_y$ redundant cells are automatically generated by the encoder. There is no buffering needed in the encoder.

Figure 4.4 illustrates the format of a coded cell. We will refer to a sequence of n_y cells as a *block*. The entire encoding operation results in a binary $(n_x n_y, k_x k_y)$ linear code referred to as a *two-dimensional code* and illustrated in Figure 4.5. The code has a rate (ration of information size to code size) of

$$R = \frac{k_x k_y}{n_x n_y}. \tag{4.1}$$

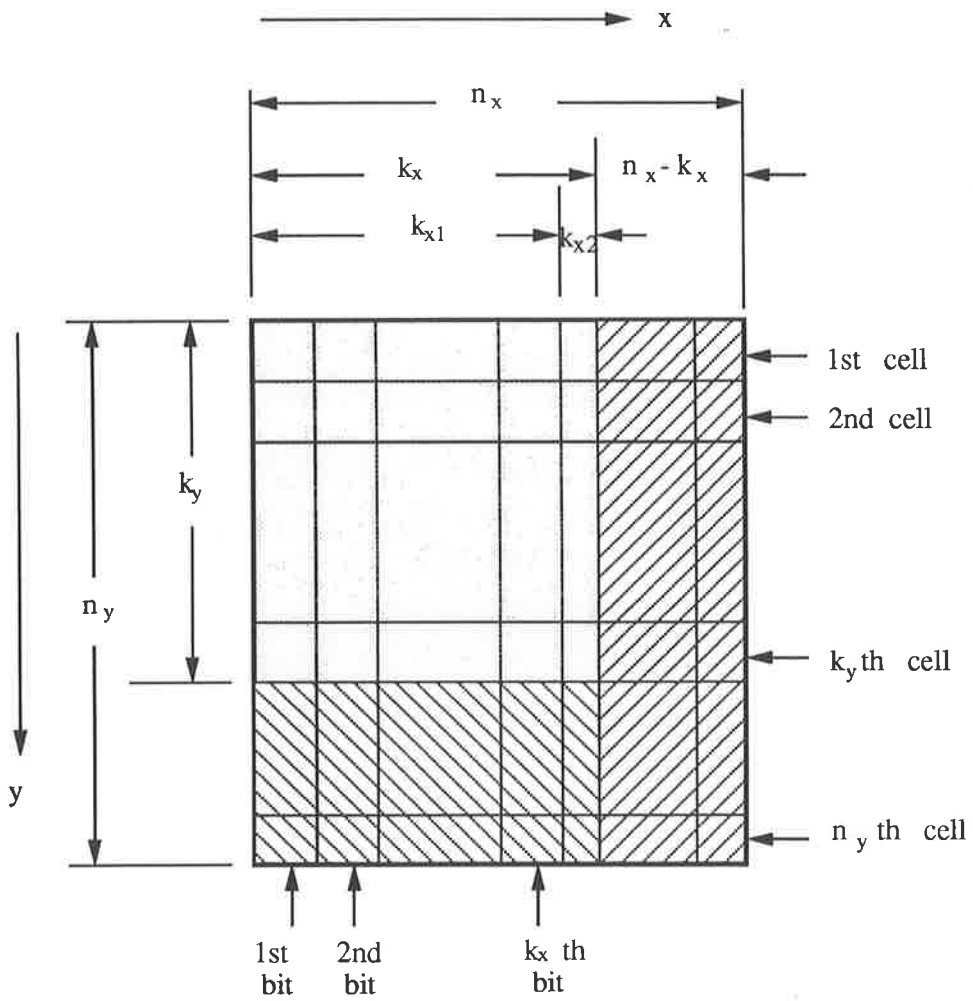


Figure 4.5 Two - dimensional block code.

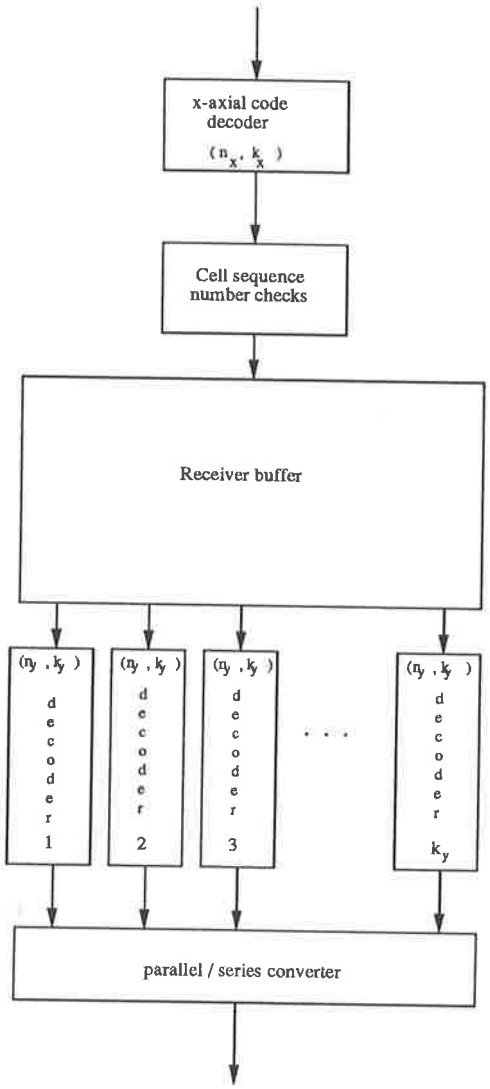


Figure 4.6 A diagram of the decoder.

The decoding for this scheme also consists of three stages as shown in Figure 4.6. The first stage is the x-axis code decoding performed to detect bit errors in the cells. When a code word $\overline{r'}_x$ in a block is received, its syndrome s is computed based on the x-axis code

C_x, that is

$$s = \overline{r'}_x \cdot \mathbf{H}_x^T \tag{4.2}$$

where $\mathbf{H}_x$ is the parity check matrix of C_x [32]. If the syndrome corresponds to an error pattern ($s \neq 0$), the $n_x - k_x$ parity bits are removed from the received code word $\overline{r'}_x$ and the cell of k_x bits is given an *error mark*. If the syndrome corresponds to an error free pattern ($s = 0$), the decoded cell is without a mark.

The second stage of decoding is to detect an entire cell lost due to buffer overflow or misrouting. If a lost cell is detected, then a dummy cell (all bits are 0's) with error mark is stored in the receiver buffer. After the cell sequence number check is done, the k_{x2} bit cell sequence numbers are then removed from the received cells, and the decoded cells of k_{x1} bits, with or without error marks, are stored in the receiver buffer.

As soon as n_y received cells in a received block have been processed, the third stage of decoding begins. If there are no any cells with error mark in a received block, $n_y - k_y$ redundant cells are then removed from the received block and k_y information cells are delivered directly to the users. In this case, it is not necessary for the y-axis decoding. If there is one or more cells with error marks appearing in a received block, then the y-axis code decoder starts to decode the n_y received cells which are stored in the buffer. A lost cell or a cell with bit errors appears as a single bit erasure in the received code word $\overline{r'}_y$ which make up the block. Note that if some error patterns in code word $\overline{r'}_x$ are undetected, these errors will be stored in the buffer without an error mark, and thus cannot be corrected by the y-axis code decoder.

Referring to CCITT standard of ATM [66], a cell header of 40 bits contains 8 redundant bits for a one-bit header error forward correction. Therefore, the occurrence of bit errors in the cell header caused by transmission errors is of very low probability, and can be ignored. In this chapter, we do not address the problem of error correction in cell header. However, cell loss caused by errors in the cell header can also be detected by the cell sequence number and is also recoverable using the y-axis code.

4.3 *Small distance* codes for cell error detection

To protect the cell information field against transmission errors, the CCITT has recommended to set-up an ATM adaptation layer (AAL) containing two sublayers, the convergence sublayer (CS) and the segmentation and reassembly sublayer (SAR) in the payload field [66]. In the SAR sublayer, 10 bits are recommended for protection of transmission errors in the cell information field. However, the CCITT has not decided yet what type of error protection coding schemes should be used in the ATM adaptation layer (AAL) [66] to protect the errors contained in cell information field. The consideration of these x-axis codes are timely and important.

The use of optical fiber transmission links in B-ISDN will result in a very low bit error rate (the optical fiber involves a typical bit error rate of the order of 1 in 10^9). Therefore codes that have traditionally been used for error detection may be unnecessarily powerful, and hence would involve an excessive number of redundant bits. A class of codes, called *small-distance codes*, with $d_{min1} < 3$, has been conceived by Miller and developed further in conjunction with the author [9].

In this section, we will show that the small distance codes only with 4 - 6 parity check bits incorporated into the SAR can be sufficient to provide a satisfactory error detection mechanism for a standard cell information field of 384 bits. Then at least 4 out 10 bits reserved in the SAR for error protection are not used, and these unused bits can be used to set up a separate cell sequence number for cell loss detection, when the two-dimensional coding scheme is used as an edge-to-edge approach for a VP connection. This is one of the important reasons for us to consider the small distance codes in this chapter.

Consider an (n_x, k_x) linear block code C_x for which the number of parity bits $n_x - k_x$ is chosen such that

$$2^{n_x - k_x} - 1 < n_x. \tag{4.3}$$

The performance of an error detection code, when used on any given channel, is measured by the undetected-error probability P_{ue}, which is the probability of the event that channel noise transforms the transmitted code word into another valid code word so that parity

checking at the receiving end fails to detect the errors in the transmitted code word.

It is well known [32] that a code C_x for which (4.3) holds will have minimum distance $d_{min1} < 3$. The probability that the error detecting code will fail to detect an error in a received block is given for a binary symmetrical channel (BSC) by

$$P_{ue} = \sum_{i=1}^{n_x} A_i \, p_b^i (1 - p_b)^{n_x - i}, \tag{4.4}$$

where p_b is the channel error probability. The values A_i for $i = 1, 2, ...$ are the weight spectrum values for the code (that is, the number of code words of weight i).

For $p_b << 1$, a *small-distance* (n_x, k_x) code C_x is defined [9] as one for which $A_1 = 0$ and A_2 is minimized.

Lemma: An (n_x, k_x) linear code C_x with $2^{n_x - k_x} - 1 < n_x$, and for which $A_1 = 0$ and A_2 is a minimum, must have a parity check matrix $\mathbf{H}_x$ chosen so that

(1) no column of $\mathbf{H}_x$ contains all zeroes,
(2) no row of $\mathbf{H}_x$ contains all ones, and
(3) the number of identical columns of $\mathbf{H}_x$ should be minimized.

Proof: The validity of (1) and (3) follows from the fact [32] that any weight spectrum value A_i for an (n_x, k_x) linear block code with parity check matrix $\mathbf{H}_x$ is equal to the number of columns of $\mathbf{H}_x$ whose sum is a vector of weight i, for $i = 1, 2, ..., n_x$. Also it is easy to see that (2) follows by considering the relationship between the weight spectrum values A_i of code C_x and the spectrum values B_i of the dual code C_d. These are related by [32]

$$A(z) = 2^{-(n_x - k_x)} (1 + z)^{n_x} \, B(\frac{1 - z}{1 + z}), \tag{4.5}$$

from which it follows that if $A_1 = 0$, then

$$\sum_{i=1}^{n_x} i \, B_i = n_x \, 2^{n_x - k_x - 1}. \tag{4.6}$$

Consider the case where $\mathbf{H}_x$ contains an all-ones row vector, denoted here by $\underline{1}$. Then for every n_x-tuple v in C_x which has weight $w(v) = i$, there exists another n_x-tuple $\underline{u}$ say, in

C_x which has weight $n_x - i$. This is the vector $\underline{u} = \underline{v} + \underline{1}$. For this case, the dual code weights satisfy $B_i = B_{n-i}$ for $0 \leq i \leq n_x$. Then

$$\sum_{i=1}^{n_x} i\, B_i = n_x\{2^{n_x-k_x} - 1\}, \tag{4.7}$$

which is a contradiction to equation (4.6).

Summarizing, it follows that the *small-distance codes* C_x can be constructed using the following theorem.

Theorem: The (n_x, k_x) linear code C_x with $2^{n_x-k_x} - 1 < n_x$, $A_1 = 0$, and A_2 a minimum, has a parity check matrix $\mathbf{H}_x$ whose columns consist of all the non-zero $(n_x - k_x)$-tuple repeated φ_x times, and in addition, ϑ_x other non-zero vectors, where

$$\varphi_x = \lfloor \frac{n_x}{2^{n_x-k_x} - 1} \rfloor, \tag{4.8}$$

and

$$\vartheta_x = n_x - \varphi_x(2^{n_x-k_x} - 1), \tag{4.9}$$

the notation $\lfloor w \rfloor$ denotes the largest integer smaller than w. Then the parity check matrix $\mathbf{H}_x$ can be defined as

$$\mathbf{H}_x = [\underbrace{\mathbf{I}\,\mathbf{I}\,\ldots\,\mathbf{I}}_{\varphi_x}\ \underbrace{\mathbf{Q}\,\mathbf{Q}\,\ldots\,\mathbf{Q}}_{\varphi_x}\ \mathbf{R}], \tag{4.10}$$

where $\mathbf{I}$ is an $(n_x - k_x) \times (n_x - k_x)$ identity submatrix, $\mathbf{Q}$ is an $(n_x - k_x) \times [2^{n_x-k_x} - (n_x - k_x) - 1]$ submatrix containing all the columns with weight greater than 1, both $\mathbf{I}$ and $\mathbf{Q}$ repeat φ_x times, respectively. $\mathbf{R}$ is an $(n_x - k_x) \times \vartheta_x$ submatrix containing any distinct non-zero column vectors.

When these codes are used for error detection on a binary symmetric channel (BSC), the number of undetectable error patterns of weight 2 is

$$A_2 = n_y - (2^{n_y-k_y} - 1)(\varphi_x - \frac{\varphi_x(\varphi_x - 1)}{2}), \tag{4.11}$$

and the probability of undetected error for $p_b << 1$ will be closely approximated by

$$P_{ue} = A_2\, p_b^2 (1 - p_b)^{n_x - 2}. \tag{4.12}$$

Example: Consider the case where a $(32, 28)$ code is required for error detection. Then from the theorem, we have $\varphi_x = 2$ and $\vartheta_x = 4$, and we choose a code C_x with parity check matrix as

$$\mathbf{H}_x = [\,\mathbf{I}\,\mathbf{I}\,\mathbf{Q}\,\mathbf{Q}\,\mathbf{R}\,]$$

where $\mathbf{I}$ is an 4×4 identity submatrix, $\mathbf{Q}$ is an 4×11 submatrix containing all the columns with weight greater than 1, and $\mathbf{R}$ is an 4×2 submatrix containing any distinct non-zero column vectors. We might choose

$$\mathbf{H}_x = \begin{bmatrix} 1000 & 1000 & 11100011101 & 11100011101 & 10 \\ 0100 & 0100 & 10011011011 & 10011011011 & 01 \\ 0010 & 0010 & 01010110111 & 01010110111 & 00 \\ 0001 & 0001 & 00101101111 & 00101101111 & 00 \end{bmatrix}. \tag{4.13}$$

This code has weight spectrum values $A_2 = 17$, $A_3 = 612$.

The characteristics of (n_x, k_x) small-distance codes are listed in Table 4.1 for various values of $(n_x - k_x) = 1, 2, 3, \ldots$ parity check bits. In the table, A_i, $i = 2$ and 3 represents the number of undetectable error patterns of 2 and 3 errors respectively. These values can be used in equation (4.4) to obtain a good approximation to P_{ue}. In table 4.1, the symbol "$\surd$" indicates the existence of a code with the parameters given. Figure 4.7 shows how the probability, P_{ue}, of undetectable errors varies as function of the number of redundant parity check bits $(n_x - k_x)$ and n_x, for the case where the link bit error rate is $p_b = 10^{-9}$. In Table 4.1 and Figure 4.7, $n_x = 40$ is the standard cell header size, and $n_x = 384$ is the standard cell information field size according to CCITT recommendation [1].

n_x	$n_x - k_x$	d_{min1}	Detects all single error patterns	A_2	Detects all double-error patterns	A_3	Detects all triple-error patterns
40	2	2	√	184	-	3270	-
40	3	2	√	65	-	2481	-
40	4	2	√	38	-	612	-
384	3	2	√	12394	-	1.16×10^7	-
384	4	2	√	5674	-	2.9×10^6	-
384	5	2	√	2439	-	6.7×10^5	-
384	6	2	√	991	-	1.5×10^5	-
384	7	2	√	424	-	4.1×10^4	-
384	8	2	√	297	-	9.7×10^3	-

Table 4.1 Property of the small distance codes for various values of n_x and $n_x - k_x$.

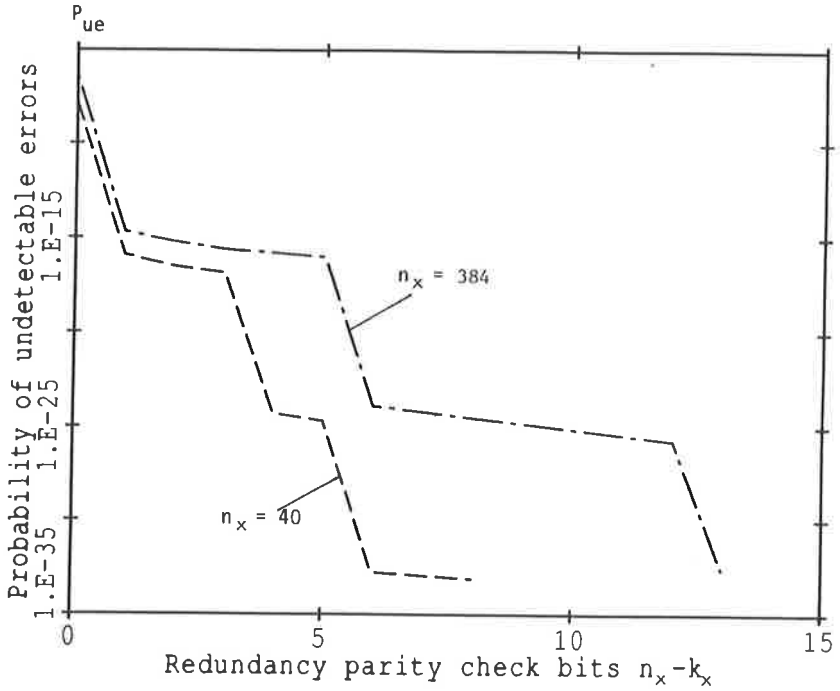


Figure 4.7 The probability of undetectable errors versus the parity check bit. The link bit error rate is $p_b = 10^{-9}$.

For the purpose of error detection in the information field of 384 bits of a ATM cell based on optical fiber transmission links, high rate small-distance codes with only 4 - 6 parity check bits located in the SAR will be appropriate. Some suitable small distance codes such as (384, 380), (384, 379), and (384,378) and their capacity of detecting various error patterns are shown in Table 4.1. If channel error rate values are expected to lie in range $p_b \leq 10^{-9}$, and using small distance codes (384, 380) (384, 379), and (384,378) with very high code rate ranging from 0.98 to 0.99, as shown in Figure 4.7, the probability that undetectable errors contained in the cell information field will be in a very low range from 2.6×10^{-16} to 6.7×10^{-23}. These codes with only 4 - 6 parity check bits incorporated into the SAR sublayer can be sufficient to provide a satisfactory error detection mechanism for a standard cell information field size of 384 bits [1]. We note that, in this coding scheme, at least 4 out of 10 bits reserved in the SAR for error protection are not used. As discussed in section 4.2, these unused bits will be used to set up a separate cell sequence number for cell loss detection, when the coding scheme is used as an edge-to-edge approach for a VP connection. We also note that the error detection of using the small distance codes will cover whole cell payload field including the cell sequence number. We will discuss this in detail in section 4.5.

4.4 Erasure correcting codes for error recovery

In our two-dimensional coding scheme, the y-axis code is designed to perform erasure correction rather than error correction. In this section, we will outline some basic background material on erasure correcting codes and explore their possible utilization for correcting cell errors and recovering lost cells.

After x-axis code decoding, the lost cells detected and the cells containing detected errors are stored in the receiver buffer with an error mark. Other received cells are stored in the buffer without any error mark. In the received y-axis code words, each cell with an error mark appear as a single bit, which is called an erasure, defined as an error whose location is known but whose value is not.

Consider a code with minimum distance d_{min2}. Then there exists at least one pair of

code words in the y-axis code C_y such that there are exactly d_{min2} places in which they differ. Now we suppose that $\overline{r}_{y_1}$ is a transmitted code word, and the resultant received code word $\overline{r'}_{y_1}$ has d_{min2} bits erased. Let $\overline{r}_{y_i}$ be one of the code words in the y-axis code C_y such that there are exactly d_{min2} places in which they are differ from the code word $\overline{r}_{y_1}$. If the d_{min2} erasure locations coincide with the d_{min2} locations in which $\overline{r}_{y_1}$ and $\overline{r}_{y_i}$ differ, then the received code word $\overline{r'}_{y_1}$ will be incorrectly decoded as $\overline{r}_{y_i}$. On the other hand, if $(d_{min2}-1)$ or less bits are erased in the received code word, then there will be at least one place in which it is different from $\overline{r'}_{y_1}$ and $\overline{r}_{y_i}$. Since $\overline{r}_{y_i}$ has been known to be a correctly code word without any erasures, then code word $\overline{r}_{y_1}$ is the only possible code word which can be chosen by the decoder. Hence, a code with a minimum distance d_{min2} can correct up to $(d_{min2}-1)$ erasures. By contrast, if used for error correction, a code with a minimum distance d_{min2} can only correct $(d_{min2}-1)/2$ or fewer random errors [32].

An erasure correcting decoder can be implemented using error-correcting procedures. First, the decoder guesses at the values transmitted in the erased locations. Then it attempts to decode the resultant code sequence using some standard error-correcting algorithm. If a legitimate code sequence results, then the decoding is done. Otherwise, the decoder attempts another guess of erasure values. This process is repeated until the correct code sequence is found.

For example, consider a binary code with a minimum distance d_{min2} used for erasure correcting decoding. Let $X \leq (d_{min2}-1)$ be the number of erasures in the received code word. First, the decoder sets all the erased bits to 0 and attempts decoding. If the decoding fails, the decoder then repeats the procedure with the erased bits set to 1's. Now we suppose that the X erased bits originally contained $X1$ zeros and $X2$ ones. Thus there are $X2$ and $X1$ errors in the first and second trial respectively. The fact is that $X = X1 + X2$ is less than or equal to $(d_{min2}-1)$. This guarantees that either $X1$ or $X2$ or both are less than or equal to $(d_{min2}-1)/2$. Recall that a code with minimum distance d_{min2} can correct $(d_{min2}-1)/2$ or fewer random errors. Hence, decoding will be successful in at least one of the two trials. In general, for an m_2-ary code (each symbol consists of m_2 bits), the decoder takes at most $2(m_2/2)^e$ trials to correct e erasures [50].

This method can also be used to correct combinations of erasures and errors, provided

the number of erasures and errors satisfy the equation

$$d_{min2} \geq 2t_2 + e + 1, \tag{4.14}$$

where e and t_2 are the number of erasures and errors to be corrected respectively. If only erasure correction is desired, the procedures involved are generally simpler than those for error correction, because for erasure decoding the error locations are known. An erasure correcting procedure for Reed-Solomon codes is described in detail in appendix A.

4.4.1 Cyclic codes for erasure correction

In this section, we consider a class of high rate cyclic codes [32] to be implemented as the y-axis code for erasure correction. By restricting the choice of cyclic codes, the encoding and decoding implementation can be considerably simplified and variable length can be encoded/decoded readily. An encoder of a cyclic code (n_y, k_y) appends $(n_y - k_y)$ parity bits to an input binary information string of every k_y bits in such a way that the resulting code words correspond to polynomial multiples of a generator polynomial $g(y)$ of degree $(n_y - k_y)$. The decoding of a cyclic code is also simple because, in the absence of erasures, the received polynomial must be a multiple of $g(y)$.

In an ATM network, for some transfer applications, cells are transmitted in bursts which are separated by long silence period. In this case, an uncompleted block may appear in the receiver buffer at the end of an arriving burst, and the y-axis decoder will not be able to decode the uncompleted block until the next burst arrives at the receiver buffer. This may cause unacceptable buffering delay at the receiver end. Hence, the y-axis code may be required to perform in variable length.

Consider an y-axis encoder based on a generator polynomial $g(y)$ which generates an (n_y, k_y) cyclic code. Let the number of bits to be encoded be $k'_y \leq k_y$. The encoder produces a fixed number $(n_y - k_y)$ of parity redundancy bits. The resultant code is of length $n'_y = k'_y + (n_y - k_y)$ bits. The y-axis code word is said to be one of the code words in a *shortened cyclic* (n'_y, k'_y) code produced by $g(y)$. It has been shown that in [32], the minimum distance of a shortened cyclic code is at least as great as that of the original

code. The error recovery decoding of this shortened code is also compatible with the original code. However, it is necessary to incorporated a flag sequence at the beginning and end of a code word. This can be done by using the MID filed in the SAR header [66] which associates all cells belonging to a given two-dimensional code block. The receiver uses these information to locate the beginning of each code word and also determine the number of bits n'_y to be entered into the syndrome computation circuit before checking the resultant syndrome pattern in each y-axis decoder. Further details can be found in [32].

Since the cell loss due to buffer overflow or network congestion may occur in bursts, we require a burst error correction code. Some very efficient cyclic codes and shortened cyclic codes for single burst error correction have been found either analytically or with the aid of a computer [32]. A number of such codes ranging from high rate to low rate, and which are suitable for our purpose, are shown in Table 4.2.

$n_y - k_y - l$	Code (n_y, k_y)	Code rate	Burst-correcting capability l	generator polynomial $g(y)$
0	(15, 9)	0.6	3	171
	(19, 11)	0.58	4	1151
	(50, 34)	0.59	8	224531
	(56, 38)	0.68	9	1505773
1	(23, 12)	0.52	5	5343
	(48, 37)	0.77	5	4501
	(63, 48)	0.76	7	105437
	(67, 54)	0.81	6	36365
	(96, 79)	0.82	7	114361
	(103, 88)	0.85	8	501001

Generator polynomials are given in an octal representation. Each digit represents three binary digits according to the following code:

$0 \leftrightarrow 000,\ 2 \leftrightarrow 010,\ 4 \leftrightarrow 100,\ 6 \leftrightarrow 110,$

$1 \leftrightarrow 001,\ 3 \leftrightarrow 011,\ 5 \leftrightarrow 101,\ 7 \leftrightarrow 111,$

The binary digits then give the coefficients of the polynomial, with the high-order coefficients at the left. For example, the binary representation of 171 is 001111001, and the corresponding polynomial is $g(y) = y^6 + y^5 + y^4 + y^3 + 1$

Table 4.2 List of primitive polynomials for burst error codes.

$n_y - k_y - l$	Code (n_y, k_y)	Code rate	Burst-correcting capability l	generator polynomial $g(y)$
2	$(17, 9)$	0.53	3	471
	$(35, 23)$	0.66	5	13627
	$(39, 27)$	0.69	5	13617
	$(51, 41)$	0.8	4	3501
	$(57, 39)$	0.69	8	1341035
	$(63, 51)$	0.81	5	16447
	$(85, 73)$	0.86	5	10131
	$(105, 91)$	0.87	6	70521
	$(131, 119)$	0.91	5	55725
	$(169, 155)$	0.92	6	55725
3	$(51, 42)$	0.82	3	1455
	$(93, 82)$	0.88	4	6137
	$(151, 136)$	0.9	6	114371
	$(164, 153)$	0.93	4	6255
	$(195, 182)$	0.93	5	22475
	$(217, 202)$	0.93	6	12247
	$(290, 277)$	0.96	5	24711
4	$(117, 105)$	0.9	4	13413
	$(133, 115)$	0.87	7	1254355
	$(255, 239)$	0.94	6	301565
	$(273, 261)$	0.96	4	10743
	$(595, 581)$	0.98	5	64655
5	$(465, 454)$	0.98	3	7275
	$(1023, 1010)$	0.99	4	22365

Table 4.2 List of primitive polynomials for burst errors codes (continued)

The above codes are suitable for handling single short bursts. We may also need to consider, however, some more powerful codes for burst error correction, particularly when there is a possibility of multiple-error bursts. One such code is the Reed-Solomon (RS) code.

4.4.2 Reed-Solomon (RS) codes for erasure correction

Reed-Solomon (RS) codes were discovered by I.S. Reed and G. Solomon [53]. They are a special subclass of the non-binary m_2-ary Bose Chaudhuri Hocquenghem (BCH) codes [32]. In this section, we consider the class of RS codes to implement a y-axis code for lost cell recovery.

A RS code is composed of symbols which are elements of a galois field. A galois field $GF(2^{m_2})$ is a field with 2^{m_2} elements, including one zero element and $2^{m_2}-1$ nonzero elements. In fact, these $2^{m_2}-1$ nonzero elements in $GF(2^{m_2})$ form a closed group under the multiplication operation [32]. An (n_y, k_y) RS code in $GF(2^{m_2})$ is chosen with $n_y = 2^{m_2}-1$ symbols which represent the $2^{m_2}-1$ nonzero elements in $GF(2^{m_2})$, each symbol consists of m_2 bits. Therefore, a (n_y, k_y) RS code could also be represented as a $(m_2 n_y, m_2 k_y)$ binary code over $GF(2)$.

Now we consider an y-axis code for lost cell recovery made up of code words in a specified Reed-Solomon (RS) code. Let k_{x1} be the number of bits in a cell to be erasure decoded (k_{x1} excludes the cell header, the cell sequence number, and the parity check bit in the AAL). Recall the definition of the two-dimensional code, we know that a code block consists of k_{x1} parallel y-axis codes. If a y-axis code is implemented using a (n_y, k_y) RS code in $GF(2^{m_2})$, then the two-dimensional code will be a block of $m_2 n_y \times k_{x1}$ bits which consists of $m_2 n_y$ cells, because each y-axis code consists of $m_2 n_y$ bits. To understand the structure in a two-dimensional code block in which the y-axis code is made up in a specified (n_y, k_y) RS code in $GF(2^{m_2})$, let $CEL(j, i)$ represent the ith bit of cell j. Then

the cells in a coded block can be arranged in the following format.

$$
\begin{array}{cccccc}
CEL(1,1) & CEL(1,2) & CEL(1,3) & ... & ... & CEL(1,k_{x1}) \\
CEL(2,1) & CEL(2,2) & CEL(2,3) & ... & ... & CEL(2,k_{x1}) \\
. & . & . & . & . & . \\
. & . & . & . & . & . \\
CEL(m_2k_y,1) & . & . & . & . & CEL(m_2k_y,k_{x1}) \\
CEL(m_2k_y+1,1) & . & . & . & . & CEL(m_2k_y+1,k_{x1}) \\
. & . & . & . & . & . \\
CEL(m_2n_y,1) & . & . & . & . & CEL(m_2n_y,k_{x1})
\end{array}
$$

The code block consists of m_2k_y information cells and $m_2n_y - m_2k_y$ redundancy cells.

The next step is to create a "cell symbol" for every m_2 cells. For the first m_2 cells, we convert $CEL(1,i)$, $CEL(2,i)$, ..., $CEL(m_2,i)$ into a y-axis symbol, denoted as $SYM(1,i)$ in $GF(2^{m_2})$. That is

$$
\begin{array}{cccccc}
CEL(1,1) & CEL(1,2) & CEL(1,3) & ... & ... & CEL(1,k_{x1}) \\
CEL(2,1) & CEL(2,2) & CEL(2,3) & ... & ... & CEL(2,k_{x1}) \\
. & . & . & . & . & . \\
. & . & . & . & . & . \\
CEL(m_2,1) & CEL(m_2,2) & CEL(m_2,3) & . & . & CEL(m_2,k_{x1}) \\
\Downarrow & \Downarrow & \Downarrow & . & . & \Downarrow \\
SYM(1,1) & SYM(1,2) & SYM(1,3) & . & . & SYM(1,k_{x1})
\end{array}
$$

We will refer to the symbol array $SYM(1,1)$, $SYM(1,2)$, $SYM(1,3)$, ..., $SYM(1,k)$ as

a *cell symbol.* This conversion results in n_y cell symbols for each coded block, that is

$$\begin{array}{ccccccc}
SYM(1,1) & SYM(1,2) & SYM(1,3) & ... & ... & SYM(1,k_{x1}) \\
SYM(2,1) & SYM(2,2) & SYM(2,3) & ... & ... & SYM(2,k_{x1}) \\
. & . & . & . & . & . \\
. & . & . & . & . & . \\
SYM(k_y,1) & . & . & . & . & SYM(k_y,k_{x1}) \\
SYM(k_y+1,1) & . & . & . & . & SYM(k_y+1,k_{x1}) \\
. & . & . & . & . & . \\
SYM(n_y,1) & SYM(n_y,2) & . & . & . & SYM(n_y,k_{x1}) \\
\Downarrow & \Downarrow & \Downarrow & . & . & \Downarrow \\
\overline{r}_{y_1} & \overline{r}_{y_2} & \overline{r}_{y_3} & . & . & \overline{r}_{y_k}
\end{array}$$

Each column $SYM(1,i)$, $SYM(2,i)$, ..., $SYM(n_y,i)$ forms a RS code word $\overline{r}_{y_i}$, and these k_{x1} RS code words constitute a cell recovery code. This particular code can recover $(n_y - k_y)$ lost cell symbols, and each lost cell symbol can contain up to m_2 lost cells.

An (n_y, k_y) RS code in $GF(2^{m_2})$ has a minimum distance $d_{min2} = n_y - k_y + 1$, and can correct any burst patterns whose erasures are confined to $(n_y - k_y)$ or fewer cell symbol burst groups. However, an erasure cell symbol can contain up to m_2 erasure cells. Thus a single erasure burst group of length less than or equal to

$$L_m = m_2(n_y - k_y) - m_2 + 1 \tag{4.15}$$

can affect at most $(n_y - k_y)$ consecutive m_2-cell symbols and so still be recovered. L_m is the maximum capacity of recovering from single burst loss in a received code block. A number of RS codes with high code rate and suitable for our purpose are shown in Table 4.3. Furthermore, a received block may have more than one separate burst group. The two-dimensional code can recover N_s burst groups with a total length [41] of

$$L_c = (n_y - k_y - 2N_s + 1)m_2 + 2N_s - 1 \tag{4.16}$$

cells, where $N_s \leq (n_y - k_y)$. For example, a (15, 7) RS code over $GF(2^4)$ with $d_{min2} = 9$ and a cell symbol length of 4 cells can correct a total of 8 erasure cell symbols, and the following burst group patterns:

1 burst group of erasures of total length	29 cells
2 burst groups of erasures of total length	23 cells
3 burst groups of erasures of total length	17 cells
4 burst groups of erasures of total length	11 cells

RS code (n_y, k_y)	m_2	code rate	Minimum distance d_{min2}	Maximum burst-correcting capability L_m (cells)
$(127, 119)$	7	0.93	9	50
$(127, 115)$	7	0.9	13	67
$(255, 233)$	8	0.9	23	169

Table 4.3 List of RS codes for burst loss recovery.

Compared to the RS code, the implementation of an erasure code using a binary cyclic code involves simpler and faster encoders and decoders. However, while the binary cyclic code is more suitable for applications in which random errors prevail, the RS code is more powerful for long burst or multiple burst loss recovery. In order to choose a suitable coding scheme, it is necessary to carefully examine the statistics of burst cell loss, the requirements for service quality, and the complexity of the encoder and decoder.

Erasure correction for RS codes mainly involves the solving of a set of simultaneous linear equations constructed by substituting a primitive field element and its power into received code sequence. The actual procedures are explained as in [50][51]. Using the method described in [50] [51], the process of erasure correction decoding for a RS code can be reduced to only V ($V \leq n_y - k_y$) trials. The details are shown in appendix A.

4.5 Cell sequence number

In order for this two-dimensional coding scheme to be possible, a cell sequence number used for cell loss detection is required to be incorporated into the SAR sublayer located in the AAL in the payload field.

End-to-end approach

Firstly, we consider the two-dimensional coding scheme as an end-to-end approach for the virtual channel connections. According to the current CCITT standard [66], 4 bits are reserved in all ALL types 1 to 4 for cell sequence number which is used to detect lost cells and misinserted cells due to buffer overflow, discarding, or misrouting. Therefore, one of the advantages of the end-to-end error recovery approach using two-dimensional coding is simple cell loss detection using cell sequence number checks. However, in end-to-end forward error recovery, the delay caused by receiver buffer may be large for low-speed virtual channels. Thus, search of short codes such as cyclic code, and simplification of hardware equipment are significant issues.

Edge-to-edge approach

Secondly, we consider the two-dimensional coding scheme as an edge-to-edge approach for the virtual path connection. Since the 4-bit cell sequence number reserved in the SAR sublayer are used for end-to-end connections, we will set up a separate cell sequence number, called **virtual path cell sequence number**, in the SAR sublayer for the edge-to-edge connections. According to CCITT standard, 10 parity redundant bits are reserved in the SAR sublayer for error protection in cell information field. However, as discussed in section 4.3, in our two-dimensional coding scheme, the use of a *small distance* code with only 4 - 6 parity check bits can sufficiently provide a satisfactory error detection mechanism for a standard cell information field size of 384 bits. For example, as shown in Figure 4.7, if the transmission bit error rate is 10^{-9}, then using a (384, 379) code with 5 bits redundancy, the probability that a cell contains undetectable errors in the information field is 1.7×10^{-17}. Therefore, in our two-dimensional coding scheme, at least 4 out of 10 bits reserved in the SAR sublayer for error protection are not used. We will use these unused bits to set up the virtual path cell sequence number for cell loss detection at the edge-to-edge connection level (In this case, there are two cell sequence numbers located in the SAR, one for end-to-end connection, and another for edge-to-edge connection). Note that when the two-dimensional coding scheme is used as an edge-to-edge approach, the original 4-bit cell sequence number reserved in the SAR for the end-to-end virtual channel connections are not processed at the encoder and decoder of the two-dimensional code.

At the transmitting edge, the encoding can be done at port controller, before the VPI is added into cell header. Firstly, in the y-axis, k_y information cells which will be delivered into the same virtual path are encoded into n_y cells. Secondly, a virtual path cell sequence number which indicates cell position in the n_y coded cells are incorporated into the SAR sublayer of each cell at the output of the y-axis encoders. Finally, in the x-axis, each cell information field is encoded using a *small distance* code.

The error recovery decoding can be done after cell header decoding is done, and VPI is removed from the cell header, i.e., cells are sorted into individual VPs. The first stage of decoding is the x-axis code decoding. The second stage of decoding is to detect an entire cell loss due to buffer overflow or misrouting using the virtual path cell sequence number. After x-axis decoding and cell sequence number check, the parity check bits and the virtual path cell sequence number are removed from the SAR sublayer. Then y-axis code decoding starts.

The size of the cell sequence number is determined by the maximum length of the lost cell burst in a block which can be recovered. For example, a (169,155) cyclic code as the y-axis code will allow a maximum length of 6 cells in a cell loss burst to be corrected in a block. In this case, 4 bits will be adequate for the cell sequence number. Using a longer cell sequence number would allow longer bursts of lost cells to be corrected. If a non-binary RS code is used as a y-axis code, the cell sequence number is determined by the maximum length of the erasure cell symbols (i.e., a cell symbol unit of m_2 cells containing the lost cells) which can be recovered in a block. The m_2 cells forming a symbol will all have the same cell sequence number. An extended cell sequence number might be incorporated into the reserved bits in the SAR sublayer in the AAL in the payload field [66].

4.6 Uncompleted blocks

As mentioned in section 4.2, in the receiver buffer, y-axis code word of length $m_2 n_y$ cells ($m_2 = 1$ for binary codes) is decoded block-by-block. However, the cells may be transmitted in bursts which are separated by long silence periods. Thus an uncompleted block may appear in the receiver buffer at the end of an arriving burst, and the y-axis

decoder will not be able to decode the uncompleted block until the next burst arrives at the receiver buffer. This may cause unacceptable buffering delay at the receiver end.

To overcome this, we may employ a cyclic code as the y-axis code. When the end of a burst is detected by the y-axis encoder and the cell sequence number shows that it is an uncompleted block, say k'_y ($k'_y \leq k_y$) cells in the uncompleted block. Then a shortened cyclic code (n'_y, k'_y) ($n'_y = k'_y + (n_y - k_y)$) is delivered, where (n'_y, k'_y) is generated based on a generator polynomial $g(y)$ which generates a (n_y, k_y) cyclic code. According to the discussion made in section 4.4.1, the minimum distance of the shortened cyclic code (n'_y, k'_y) is at least as great as that of the original code (n_y, k_y). The error recovery decoding of this shortened code is also compatible with the original code. However, it is necessary to incorporated a flag sequence at the beginning and end of a code word. This can be done by using the MID filed in the SAR header [66] which associates all cells belonging to a given two-dimensional code block. The receiver uses these information to locate the beginning of each code word and also determine the number of bits n'_y to be entered into the syndrome computation circuit before checking the resultant syndrome pattern in each y-axis decoder. Therefore, using a cyclic code as the y-axis code in the two-dimensional coding scheme, the code block can either all have the fixed size or vary from block to block using shortened cyclic code. Hence, the uncompleted blocks is avoided.

4.7 Performance with coding

In this section, we consider the performance with coding in the presence of transmission errors and cell loss due to the network congestion at the cell level. This forward error protection scheme is considered as an edge-to-edge approach for an individual virtual path at the virtual path connection level or as an end-to-end approach for an individual service at the virtual channel connection level.

As discussed in chapters 2 and 3, let us consider that a given virtual path VP_a is modelled as a tandem queueing network consisting of Q single server queues in series operating on a first-in-first-out (FIFO) basis. A queue I (I= 1, 2, ..., Q) along the path VP_a is a bulk arrival queue with a finite buffer of length K. N independent input VPs (including VP_a)

join the queue I. As discussed in section 2.2, the switching functions result in mixing of traffic from different input streams and splitting of traffic to different output streams. The designated traffic stream A is considered to be carried on the virtual path VP_a through the tandem queueing network.

In the following analysis, the performance with coding will be estimated based on two types of cell traffic streams: i) mixtures of constant rate deterministic arrival traffic and ii) mixtures of homogeneous burst-silence sources with short burst length. For the mixtures of fixed rate random arrival traffic, cell loss caused by congestion at the cell level is studied in chapter 2. For the mixtures of homogeneous burst-silence processes, the cell loss due to congestion at the cell level is based on the simulation results obtained in chapter 3, in which the burst length is considered to be less than the queueing buffer size (see section 3.5). The cell arrival rate of the designated A-stream which arrives at the first queue of the tandem queueing network is $p_a^{(1)}$. The following analysis is also based on a RS code in $GF(2^{m_2})$ ($m_2 = 2$ for a binary code). We assume that cell errors caused by transmission errors and cell loss caused by buffer overflow are independent events.

4.7.1 Uncoded case

First, we consider the uncoded case. We assume that the transmission link is a binary symmetric channel with a bit error rate p_b. Then the probability that a cell information field of length k_x bits contains at least one bit error is given by

$$P_e = \sum_{i=1}^{k_x} \binom{k_x}{i} p_b^i (1 - p_b)^{k_x - i} \tag{4.17}$$

Note that here we do not consider the errors in the cell header since the cell header has its own error protection coding scheme, and after decoding the incorrect errors contained in the cell header is negligible by comparing them to the errors in the cell information field.

For random traffic, as discussed in chapter 2, the probability of cell loss for the designated A-stream due to congestion at the cell level can be obtained from equation (2.26) for a given set of parameters of the tandem queueing network namely Q, K, N, $p_a^{(1)}$ and p.

That is

$$P_{loss} = [1 - \prod_{I=1}^{Q}(1 - P_{ls}^{(I)})]/p_a^{(1)}, \tag{4.18}$$

where $P_{ls}^{(I)}$ is given by equation (2.24).

For the mixtures of homogeneous burst-silence sources with short bursts, as discussed in chapter 3, the probability of cell loss, P_{loss}, for the designated A-stream caused by congestion at the cell level can be obtained from equation (3.20) by means of simulation.

Since we assume the cell errors caused by transmission error and cell loss caused by buffer overflow to be independent events, in the uncoded case, the total loss/error probability including cells with transmission errors and cell losses due to buffer overflow is

$$P_{LE} = 1 - (1 - P_e)(1 - P_{loss}) \tag{4.19}$$

4.7.2 Coded case

Next, we consider the coded case. For the performance of the x-axis (n_x, k_x) code ($k_x = k_{x1} + k_{x2}$, where k_{x2} is the number of bits of the cell sequence number) under random bit errors, we will determine the probability P_m that a cell information field including the AAL contains errors which can be detected by the x-axis code. Let P_c be the probability that a decoded cell information field is error free, that is, the corresponding received code word $\overline{r'_x}$ contains no error. Then

$$P_c = (1 - p_b)^{n_x} \tag{4.20}$$

Now let P_u be the probability that the x-axis code (n_x, k_x) fails to detect the errors in a received code word. This is actually the probability of the occurrence of an error pattern of d_{min1} or more errors, whereby decoding results in an incorrect code word. Then

$$P_{ue} = \sum_{i=d_{min1}}^{n_x} A_i p_b^i (1 - p_b)^{n_x - i} \tag{4.21}$$

where the A_i for $i = 1, 2, ..., n_x$ are the weight spectrum values for the code (e.g., the number of code words of weight i). If P_m is the probability that a cell has detectable

errors, then

$$P_c + P_{ue} + P_m = 1 \tag{4.22}$$

$$P_m = 1 - (1 - p_b)^{n_x} - \sum_{i=d_{min1}}^{n_x} A_i p_b^i (1 - p_b)^{n_x - i} \tag{4.23}$$

Note that P_m includes the cell sequence number of k_{x2} bits suffering detectable errors. As mentioned in section 4.2, when errors are detected in a received code word including the cell sequence number, then the parity check bit is removed, and the cell will be marked as a cell with error and the sequence number will be ignored. If a cell sequence number is corrupted with undetectable errors, then that cell will be detected as a lost cell, because the decoded cell sequence number is in a wrong order with the consecutive cells.

In the y-axis direction, for cell error correction and cell loss recovery, $(n_y - k_y)$ redundant cells are required to be transmitted for each block of n_y cell symbols (which contain k_y information cell symbols). Then some extra resultant traffic will be introduced into the network compared with the uncoded case. That is, for the designated A-stream with a cell arrival probability of $p_a^{(1)}$, in the coded case, the VP_a will be required to carry a traffic with the cell arrival probability

$$p_{ac}^{(1)} = B_e p_a^{(1)} \tag{4.24}$$

where

$$B_e = \frac{n_y}{k_y} \tag{4.25}$$

is the traffic expansion factor. Likewise for the other VPs which are encoded with the two-dimensional coding schemes. We note that both the parity redundant bits generated by the x-axis and the cell sequence number used for lost cell detection can be incorporated into the SAR sublayer located in the AAL in the payload field [66]. So they are not included in the traffic expansion factor.

Assume that all of the cells lost due to buffer overflow can be detected by the cell sequence number. Then for a given value of expanded traffic $\eta' = B_e \eta$, the cell loss probability

P'_{loss} can be obtained from equation (2.26) (for random arrival traffic) or equation (3.20) (for bursty arrival traffic). The total detectable error/loss probability in the presence of transmission errors and buffer overflow is given by

$$P'_{LE} = 1 - (1 - P_m)(1 - P'_{loss}) \tag{4.26}$$

Now we analyze the y-axis decoding. If the y-axis code is implemented using a RS code. Each symbol in a RS code is equivalent to a binary string of m_2 bits (this includes the binary code for $m_2 = 1$). Thus a RS code can be considered as a $(m_2 n_y, m_2 k_y)$ binary code. By definition, an (n_y, k_y) RS code with minimum distance $d_{min2} = n_y - k_y + 1$ can correct $(n_y - k_y)$ or fewer erasure symbols. The probability that the RS code fails to correct erasure symbols in a received block will be upper bounded by

$$P_r = \sum_{j=d_{min2}}^{n_y} A_j P_s^j (1 - P_s)^{n_y - j} \tag{4.27}$$

where A_j for $j = d_{min2}, (d_{min2} + 1), ..., n_y$ are the weight spectrum values for the RS code (e.g., the number of code words of weight j). These can be given by

$$A_j = \begin{cases} 1 & \text{for } j = 0 \\ 0 & \text{for } 1 \leq j \leq d_{min2} - 1 \\ \begin{pmatrix} q \\ j \end{pmatrix} \sum_{h=0}^{j-1-2t_2} (-1)^h \begin{pmatrix} j \\ h \end{pmatrix} (q^{j-2t_2-1} - 1) & \text{for } d_{min2} \leq j \leq q \end{cases} \tag{4.28}$$

where $q = 2^{m_2} - 1$, and A_j is the number of code word with weight j. The derivation of this equation is given in the references [50][52]. P_s is the symbol error probability of the RS code given by

$$P_s = \sum_{i=1}^{m_2} \begin{pmatrix} m_2 \\ i \end{pmatrix} {P'_{LE}}^i (1 - P'_{LE})^{m_2 - i} \tag{4.29}$$

where m_2 is the number of bits in a symbol of the RS code. For $P'_{LE} << 1$, the probability of uncorrectable erasures in a received block is approximately

$$P_k \approx \frac{P_r}{m_2} \tag{4.30}$$

Overall, the probability that cell suffers from undetectable errors, uncorrectable errors, and unrecoverable loss is

$$P_{LS} = 1 - (1 - P_k)(1 - P_{ue}) \tag{4.31}$$

Note that P_{LS} includes increased traffic load $\eta' = B_e\eta$ due to the transmission of parity redundancy. In this case, the corresponded average link utilization η will be given by equation (2.7) for random traffic and equation (3.8) for bursty traffic.

4.7.3 Delay

Consider a cell transferred between the two end-points of the network. The resultant overall delay will consist of transmission delay, propagation delay, queueing delay in buffers at exchanges, processing delay for switching and, in our model, processing delay for encoding and decoding. Let the cell delay in passing through all switching nodes (including transmission delay, propagation delay, queueing delay in switch buffers and processing delay for switching) be d_u. This is the delay for the uncoded case. As the cell size is fixed, the transmission delay will be the same for both the uncoded and coded case. Now we consider the code processing delay which is over and above the delays experienced in the uncoded case. This will include encoding and decoding delays, and receiver buffering delay. Let R_a bits/sec be the transmission speed of virtual path VP_a.

The x-axis encoding, y-axis encoding, and x-axis decoding can also be implemented as one-step processes [32]. The delay may be neglected compared to that of the total of y-axis decoding. However the y-axis decoding is performed only when lost cells or cells with bit errors are detected in a received block. If there are no cells with error marks in a received block of n_y cell symbols, the $n_y - k_y$ redundant cell symbols may be removed from the received block, and the k_y information cell symbols can be delivered directly to the user without y-axis decoding. If cells with error marks appear in a received block, the y-axis decoder first guesses at the erasure symbols. Then it attempts to decode the resultant code using standard error-correction techniques. If a legitimate code sequence results, the decoding is done. Otherwise, the decoder makes another guess at erasure values. This trial process is repeated until the correct code sequence is found.

For a code which can correct $2t_2$ erasures, as long as the number of incorrectly guessed symbols in a particular trial is less than or equal to t_2, the decoding will succeed in that trial. To prove this, let us consider the case where a *binary* code that can correct up to t_2 errors is used. Let $X \leq 2t_2$ be the number of erasures in the received sequences. First, the decoder sets all the erased bits to 1 and attempts decoding. If the decoding fails, the decoder then repeats the procedure with erased bits complimented. Now assume that of the X erased bits, X_1 are 1's and X_2 are 0's. There are X_2 and X_1 errors in the first and second trial respectively. The fact that $X = X_1 + X_2$ is less than or equal to $2t_2$ guarantees that either X_1 or X_2 or both are less than or equal to t_2. Thus decoding will be successful in at least one of the two trials. The processing delay of the erasure decoding is therefore upper bounded by

$$d_{c3} \leq 2\tau_{c3} \tag{4.32}$$

where τ_{c3} is the delay for one trial of erasure decoding process of a binary code.

We consider here a non-binary (n_y, k_y) RS code used as an erasure code. Each RS code word contains n_y cell symbols which are elements in the field $GF(2^{m_2})$ and each symbol has 2^{m_2} possible states. However, as mentioned in section 4.6, the total process of erasure correction decoding for a RS code can be finished by V ($V \leq d_{min2} - 1$) trials (see appendix B). Therefore, the processing delay of the erasure correction decoding for a RS code is upper bound by

$$d_{c3} \leq (d_{min2} - 1)\tau_{c4} \tag{4.33}$$

where τ_{c4} is the delay for one trial of erasure decoding process of a RS code.

Erasure code decoding requires a full block to be stored in a buffer before decoding starts. The delay of receiver buffering is approximated as

$$d_{c4} = \frac{m_2 n_x n_y}{R_a}, \tag{4.34}$$

($m_2 = 1$ for a binary code). While decoding of the buffer contents is taking place, incoming cells are stored in a second parallel buffer. In order that decoding not form a bottleneck,

This will be feasible with modern technology as shown for the examples in Tables 4.4 and 4.5.

Then the total delay is approximately given by

$$d_c \leq d_u + d_{c3} + d_{c4} \tag{4.35}$$

The processing delay $d_{c3} + d_{c4}$ is the delay over and above the uncoded case.

4.8 Illustrative schemes

In this section, we present some specific numerical results obtained from our forward error protection schemes. These schemes employ two-dimensional codes for cell error correction and lost cell recovery for each individual VPs at the cell level. In these schemes, a very high rate small distance code [9] is used as the x-axis error detection code. The y-axis codes are implemented using high rate cyclic codes or Reed-Solomon codes. For various combinations of code parameters, the cell loss/error probability, and the capability of recovering from long cell loss bursts are given in Figures 4.8 and 4.9, and Tables 4.4 and 4.5.

The values of cell loss probability in Figures 4.8 and 4.9, and Tables 4.4 and 4.5 were obtained using the following parameters, chosen for illustrative purpose only. The bit error rate of the transmission link is 10^{-9}. The cell length is assumed to be 53 octets (424 bit) [1]. In the coded case, for a cell information field of 384 bits including the AAL, we use an (384,379) *small distance* code [9] with 5 bits parity redundancy ($d_{min1} = 2$) as the x-axis code. This will give a probability that cells contain undetected errors as 1.5×10^{-17}. This is contrasted with the uncoded case with a probability of 4.24×10^{-7} that cells contain transmission errors. The code rate is 0.99. These codes were chosen also because of the ease of decoding at high bit rate.

The evaluations in Figure 4.8 and Table 4.4 are based on the analytical results obtained from chapter 2, in which mixtures of constant rate deterministic arrival traffic are considered (see Figure 2.5). The computed cell loss probability for a particular virtual path is conditioned on a given tandem queueing network, described by its parameters Q, K, N,

p and $p_a^{(1)}$. The VP_a is assumed to be a virtual path contains $Q = 5$ queues. Each queue along the path is modelled as a bulk arrival queue with $N = 50$ input VPs (including VP_a) and a finite buffer of $K = 25$. In the uncoded case, VP_a has a cell arrival probability $p_a^{(1)}$ and other input VPs except VP_a all have the same arrival probability $p = 0.018$. The average link utilization occupied by the traffic load is given by equation (2.7).

Likewise, the calculations in Figure 4.9 and Table 4.5 are based on the simulation results obtained from chapter 3, in which homogeneous mixtures of burst-silence traffic are used. The VP_a is assumed to be a virtual path contains $Q = 5$ queues. Each queue along the path is modelled as a bulk arrival queue with $N = 5$ input VPs (including VP_a) and a finite buffer of $K = 25$. Since we consider the performance with coding at the cell level, traffic with short bursts only (the burst length is less than the queueing buffer size) is considered. In the examples, the burst length is of $T_{on} = 10$ slots, the burst interval length is of $T_{off} = 2$ slots, and the source burstiness is 1.2. In the uncoded case, all burst-silence sources have the same peak arrival rate of λ given by equation (3.3). The average link utilization is given by equation (3.8).

In the examples, all VPs are covered with the same forward error recovery coding schemes. The average link utilization represents the fraction of link traffic load used to carry the information cells excluding the extra traffic load introduced by the transmission of the parity redundant cells. All results of the performance with coding shown in the following sections have been traded off with the average link utilization.

As mentioned in section 4.3, the two-dimensional coding scheme only covers cell information field of 48 octets which includes the AAL of 4 octets in the payload field. 4 bits are used as the cell sequence number for lost cell detection, and 5 bits as the parity redundant bits are used for cell error detection. Both the cell sequence number and the parity redundant bits are incorporated into the SAR sublayer of the AAL [66]. Since the cell header has its own error protection schemes [66], we do not consider the cell header error in this section. We note that since the cell header field changes along a VP, our x-axis error detection can not be used to detect the cell header errors. However, the cell loss caused by cell header errors can also be detected using cell sequence number, and recovered by the y-axis coding scheme. The probability of cell loss caused by cell header errors will

be much smaller than the cell loss probability caused by buffer overflow, because the cell header has its own powerful error correction scheme.

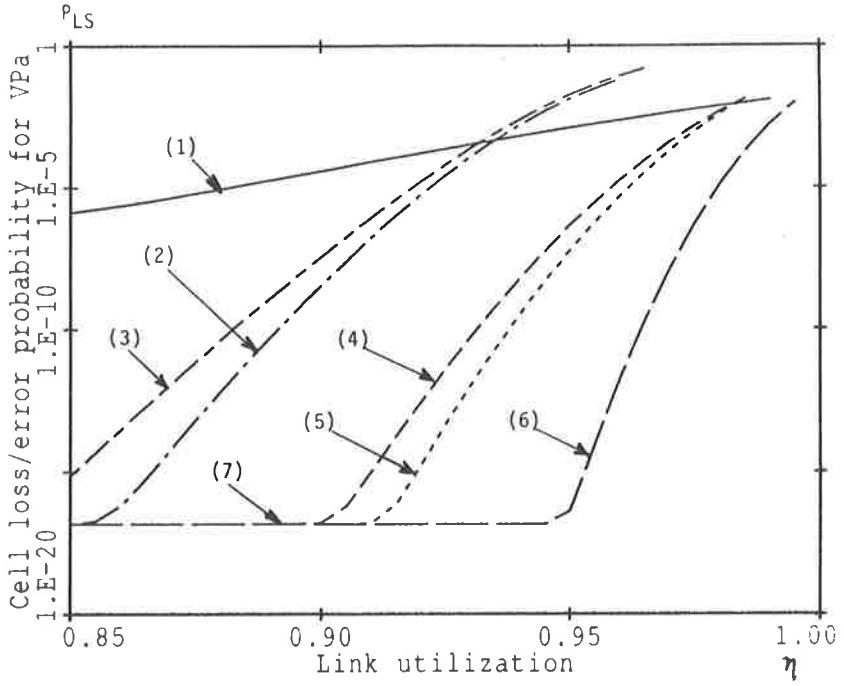


Figure 4.8 Cell loss/error probability versus the average link utilization for different combinations of coding schemes with random arrival traffic.

(1) Uncoded.

(2) (103,88) shortened cyclic code.

(3) (169,155) cyclic code.

(4) (127,119) RS code.

(5) (127,115) RS code.

(6) (255,233) RS code.

(7) Undetectable cell loss/error probability P_{ue}.

where the x-axis code employs an (384,379) *small distance* code which has 5 bits parity redundancy.

x-axis code	d_{min1}	y-axis code	d_{min2}	m_2	Code rate	k_{x2}	Undetec-table rate	Cell error/loss probability		L_m	Coding process delay	
								Mean link utilization				
								0.9	0.95		d_{c3}	d_{c4}
Uncoded	0	Uncoded	0		1.0	0	4.24E-7	3.8E-5	1.24E-3	0		
(384, 379)	2	(103, 88)	9	1	0.85	3	1.5E-17	3.7E-9	1.3E-2	8	20 μs	0.8 ms
(384, 379)	2	(169, 155)	7	1	0.91	3	1.5E-17	3.8E-8	1.8E-2	6	20 μs	1.4 ms
(384, 379)	2	(127, 119)	9	7	0.93	4	1.5E-17	1.7E-17	4.1E-7	50	0.8 ms	7.5 ms
(384, 379)	2	(127, 115)	13	7	0.9	4	1.5E-17	1.5E-17	5.9E-8	67	1.3 ms	7.5 ms
(384, 379)	2	(255, 233)	23	8	0.9	5	1.5E-17	1.5E-17	4.2E-17	169	2.3 ms	17.2 ms

$K = 25$.

m_2 is the number of bits in a symbol of RS code. ($m_2 = 1$ for a binary code)

d_{c3} is the processing delay of erasure decoding.

d_{c4} is the receiver buffering delay.

k_{x2} is the number of bits of cell sequence number.

Table 4.4 Illustrative scheme for various code combinations with random traffic.

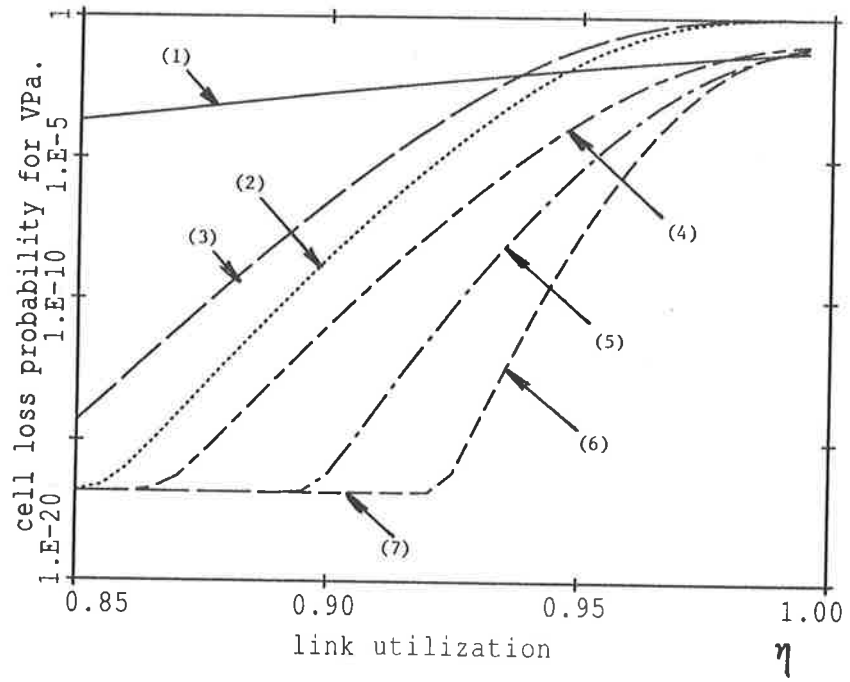


Figure 4.9 Cell loss/error probability versus the average link utilization for different combinations of coding schemes with short burst traffic.

(1) Uncoded.

(2) (103,88) shortened cyclic code.

(3) (169,155) cyclic code.

(4) (127,119) RS code.

(5) (127,115) RS code.

(6) (255,233) RS code.

(7) Undetectable cell loss/error probability P_{ue}.

where the x-axis code employs an (384,379) *small distance* code which has 5 bits parity redundancy.

x-axis code	d_{min1}	y-axis code	d_{min2}	m_2	Code rate	k_{x2}	Undetec-table error rate	Cell error/loss probability		L_m	Coding process delay	
								Mean link utilization				
								0.9	0.95		d_{c3}	d_{c4}
Uncoded	0	Uncoded	0		1.0	0	4.24E-7	3.8E-3	5.24E-2	0		
(384, 379)	2	(103, 88)	9	1	0.85	3	1.5E-17	2.7E-8	3.3E-1	8	20 μs	291 μs
(384, 379)	2	(169, 155)	7	1	0.91	3	1.5E-17	3.8E-7	4.8E-1	6	20 μs	477 μs
(384, 379)	2	(127, 119)	9	7	0.93	4	1.5E-17	8.6E-12	4.1E-5	50	0.8 ms	2.5ms
(384, 379)	2	(127, 115)	13	7	0.9	4	1.5E-17	3.2E-17	8.2E-7	67	1.3 ms	2.5 ms
(384, 379)	2	(255, 233)	23	8	0.9	5	1.5E-17	1.5E-17	5.9E-8	169	2.3 ms	5.7 ms

$K = 25$.

m_2 is the number of bits in a symbol of RS code. ($m_2 = 1$ for a binary code)

d_{c3} is the processing delay of erasure decoding.

d_{c4} is the receiver buffering delay.

k_{x2} is the number of bits of cell sequence number.

Table 4.5 Illustrative scheme for various code combination with short burtst traffic.

4.8.1 Coding gain

Figures 4.8 and 4.9 show how the cell loss/error probability varies with the average link utilization for various combinations of coding schemes. These results show that if the codes are chosen carefully, very low cell loss/error probability ranging from 10^{-10} to 10^{-17} can be achieved for an average link utilization of 0.9 (compared with the uncoded case of 10^{-5} using a queueing buffer of length 25). Even for a high average link utilization, say $\eta = 0.95$, using a (255,233) RS code, cell loss/error probability remains less than 10^{-16} for random traffic in Figure 4.8 (compared with the uncoded case of 10^{-3}), and 10^{-8} for short bursts in Figure 4.9 (compared with uncoded case of 10^{-2}). Comparison of the coding gains for random and bursty traffic, when average link utilization is high, say $\eta > 0.93$ (the corresponded peak rate exceeds the link capacity), the coding gains in the case of bursty traffic is worse than that in the case of random traffic. This is because when the average link utilization increases to the link capacity, and the corresponded peak rate exceeds the link capacity, the cell loss becomes sensitive to the burst length compared to buffer size. In this case, the length of burst loss caused by short bursts is longer than that caused by random traffic, and the capacity of recovering from the long burst losses is limited for a given forward error recovery coding scheme.

The code used to improve the performance under the high link utilization conditions must be very carefully chosen. A more powerful code generally has a lower code rate, because more parity redundancy is required. However under the high link utilization conditions, the extra traffic due to the transmission of the parity check cells may introduce more degradation from cell loss than the benefits obtained from the coding schemes. For example, as shown in Figure 4.8, if using a (127,119) RS code with a code rate of $R = 0.93$, when the average link utilization is greater than 0.97, the cell loss/error rate in coded case will be worse than the uncoded case. However, to increase the coding gain, we may only use the error protection schemes for some particular services which are sensitive to cell error and loss such as video, data, or image. In this case, the services covered with error protection may only occupy a part of the link utilization. The remaining fraction of the link utilization occupied by other services will operate without error protection. The extra traffic load introduced by transmission of parity redundant cells for error protection will

depend on the bandwidth which is encoded with forward error protection. For example, assume that present link utilization is 0.9 and 40 percent of the link utilization is covered with forward error protection. If using a (127,119) RS code with a code rate of 0.93, then an extra traffic equivalent to 0.03 of the link utilization will be introduced into the network. The traffic load actually carried on the transmission link is 0.93 compared to the uncoded case of 0.9. If the cell arrivals from all VPs are constant rate deterministic processes as discuss in chapter 2, and queueing buffer size is 25, then the actual traffic with the average rate of 0.93 will cause a cell loss probability of 1.6×10^{-4} corresponding to uncoded case of 3.8×10^{-5}. This will bring a reasonable effect to the traffic without error protection. However, (127,119) RS code will provide the cell loss rate of 1.7×10^{-17} for the VPs with coding scheme which covers 40 percent of link utilization. Summarizing, the coding gain will be affected by the bandwidth which is covered with forward error protection schemes, as greater bandwidth covered with error protection will introduce a greater amount of additional traffic load into the network.

4.8.2 Coding delay

The values of the delays shown in Table 4.4 were calculated based on the analytical model as discussed in chapter2, that is, a virtual path VP_a is modelled as a tandem queueing network consisting of $Q = 5$ queues. Each queue along the path has a finite buffer of $K = 25$ and $N = 50$ input VPs (including VP_a). The transmission speed of VP_a is $R_a = 50$ Mbits/s which is referred to be a low-speed VP. The maximum queueing delay for a cell transmitted through the VP_a is 1.1 ms.

The values of the delays shown in Table 4.5 were calculated based on the simulation model used in chapter 3, that is, a virtual path VP_a is modelled as a tandem queueing network consisting of $Q = 5$ queues. Each queue along the path has a finite buffer of $K = 25$ and $N = 5$ input VPs (including VP_a). The transmission speed of VP_a is R_a=150 Mbits/s which is referred to be a high-speed VP. The maximum queueing delay for a cell transmitted through the VP_a is 0.35 ms.

In the coding process, as discussed in section 4.7.3, the delays caused by encoding/decoding in x-axis, and encoding in y-axis are negligible compared to the y-axis erasure decoding

process and receiver buffering delay. The processing speed of a single erasure decoder carrying out one trial is assumed to be 10^5 trials per second for a binary cyclic code, and 10^4 trials per second for a non-binary RS code [36]. A large number of these erasure decoders will be operating in parallel (see Figure 4.6). We will neglect overhead due to control of the multiple trials at this stage. Now we summarize the coding delay as follows:

1. For the two-dimensional coding scheme in which the y-axis code is implemented using binary cyclic codes or RS codes, the coding delay are mainly caused by y-axis erasure decoding and receiver buffering. As shown in the tables, the delay caused by erasure decoding d_{c3} is much less than the receiver buffering delay d_{c4}, so that there are never bottleneck in the receiver buffer.

2. When coding scheme is used for low speed virtual paths or virtual channels, the delay caused by y-axis code decoding and receiver buffer may be large as shown in Table 4.4. In this case, the use of high rate binary codes such as (169, 155) cyclic code is more suitable. On the other hand, the simplification of hardware equipment is also a significant issue.

4.8.3 Equivalence of coding and buffering

Our proposal prefers to use the two-dimensional coding scheme combined with small size queues rather than to use large size queues. It is well known that cell loss due to congestion at the cell level may be controlled using large size queues located in the multiplexers and switching nodes along the virtual path.

Figure 2.5 has illustrated the cell loss probability versus the average link utilization for various queueing buffers, which is obtained using a tandem queueing network with random traffic.

A comparison of the use of coding with short queues and large queues without coding is summarized as follows:

1. The use of forward error recovery coding scheme is equivalent to increasing the buffer sizes. For example, comparing the results shown in Figures 4.8 and 2.5, for an average

link utilization $\eta = 0.95$, the cell loss rate can be reduced to 2.37×10^{-8} by using a large queue of size $K = 120$, but by using a (127,115) RS code combined with a short queue of size $K = 25$, the cell loss rate can be also reduced to 5.9×10^{-8}.

2. When the average link utilization is high, say $\eta = 0.95$, the use of two-dimensional coding scheme may cause more delay than the use of large size queue. Because we need more powerful codes with longer code length and more redundancy to overcome the long burst losses. In above example, using large queues of size $K = 120$ without coding, the maximum queueing delay for a cell transmitted through the virtual path VP_a is 5.1 ms for a low speed VP of R_a=50 Mbits/s, and 1.7 ms for a high speed VP of $R_a = 150$ Mbits/s. By contrast, using a (127,115) RS code combined with a short queue of size $K = 25$, the delay including the coding delay and the maximum queueing delay for a cell transmitted through the virtual path VP_a is 8.6 ms for the low speed VP, and 4.1 ms for the high speed VP. However, when the average link utilization is lower, say $\eta = 0.9$, some high rate binary codes such as (169,155) cyclic code may sufficiently provide a satisfactory loss recovery mechanism, as well as allowing greater control over cell delay than the use of large size queues. For example, for $\eta = 0.9$, using of a large size queue $K = 120$ can reduce the cell loss probability to 3.4×10^{-12} and the maximum queueing delay is 8.6 ms for the low speed VP, and 1.7 ms for the high speed VP. By contrast, using of a (169,155) cyclic code with a short buffer of $K = 25$ can reduce the cell loss probability to 9.8×10^{-10}, and the related delay is 2.5 ms for the low speed VP, and 0.83 ms for the high speed VP (including the queueing delay due to a cell through the tandem queueing network).

3. For the use of large size queue, the large queues must be located in all the access multiplexers and switching nodes along the path through the ATM network. They increase both cell delay and the variation in cell delay. The buffer delays also increase proportionally to the length of the virtual path cross the network. This may cause difficulties for real time services which require synchronization between the transmitter and receiver. Therefore, for real time service, an elastic receiver buffer is also needed to remove the cell delay jitter. By contrast, using the two-dimensional coding with small size queues located in the access multiplexers and switching nodes,

only one receiver buffer is required to be located at the receiving end. As discussed in section 4.7, for our two-dimensional coding scheme, the extra delays are mainly caused by receiver buffer and the y-axis decoding, but the delay is fixed. After the y-axis decoding, the cell delay jitter is automatically removed. Furthermore, the two-dimensional coding scheme can be also used for recovery of cell loss caused by misrouting, cell header error, priority discarding, and switching errors.

4. Recovery of long burst loss is possible using two-dimensional coding scheme with a short queue, and this avoids the need for very large queues along the virtual path through the ATM networks, particularly for long distance transmission.

4.9 The design of coding schemes

In this section, the numerical results used in the discussions are obtained from chapter 3 based on the simulation of a tandem queueing network with bursty arrivals of short burst length as the same as discussed in Figure 4.9. The design of coding schemes for random traffic will be in a similar manner.

The above results show that if the codes are chosen carefully, very low cell loss/error probabilities ranging from 10^{-10} to 10^{-17} can be achieved. Even for a very high rate code such as an (255, 233) RS code, as shown in Figure 4.9, the cell loss/error probability remains less than 10^{-8} (compared with uncoded case of 10^{-2}) for a bit error rate on the transmission link of 10^{-9} and the average link utilization $\eta = 0.95$.

The cell loss due to network congestion may occur in bursts. In our schemes, the y-axis code using an erasure decoding can recover $(d_{min2} - 1)$ erasures (or erasure symbols) in a received block. However for an error correction code, the capability of recovery from long cell loss bursts is limited. We must note that given a code (n_y, k_y) which can recover burst loss of length L_m cells (L_m is given by equation (4.15)) in a received block, if most losses are due to bursts with the length $l > L_m$, then we have to search for some other suitable codes.

To design a suitable coding scheme for burst loss recovery, we must carefully examine the

cell loss-free run distribution $P(0^r|1)$ given by equation (3.21). $P(0^r|1)$ is a conditional probability that a loss-free run between two cell losses is of length r or greater. While a particular code may be suitable to recover a long burst, it will also require enough loss free guard spaces in the received block [32]. $P(0^r|1)$ can indicate whether this requirement will be met. The results obtained from chapter 3 have confirmed that $P(0^r|1)$ is sensitive to the traffic load, as shown in Figure 3.9 (a), but the result of Figure 3.9 (b) shows that extending the buffer size can significantly increase the burst interval length. This will give more opportunities for using error correction coding techniques, because error correction coding requires a certain loss free spaces between the losses. However, increasing the burst interval length may result in increasing the burst density. This requires us to further carefully examine the cell loss distribution $P(u, v)$ which is given by equation (3.22). $P(u, v)$ is the probability that a block of v packet contains exactly u lost cells due to network congestion. $P(u, v)$ can help the designer to search for a suitable coding scheme to recover the lost cells due to buffer overflow. Therefore, the error correction coding technique can be combined with limited extension of the buffer size to improve the performance suffering from the short-term congestion due to heavy traffic load. The procedures are

1. Carefully examine the loss-free run distribution $P(0^r|1)$ versus the traffic load for different values of queueing buffer sizes. Using the given desirable overall quality requirements (loss and delay) and the reserved traffic load for the desired VP, we can determine the best range of the y-axis code length, say v, and queueing buffer size.

2. Examine the cell loss distribution $P(u, v)$ for different queueing buffer sizes corresponding to the given traffic load conditions. Then we can determine the minimum distance d_{min2} which is required to recover u erasures in a block of length v and with code rate of the two-dimensional code.

3. Since coding requires to transmit parity redundancy which will introduce extra traffic into the network. This extra traffic may cause more cell losses than the benefits obtained from the coding schemes. So we have to evaluated the cell loss/error probability and the performance delay for the coded case. If the network has sufficient

capacity to carry the extra traffic, and legitimate results are obtained, then the design is done. Otherwise, we have to repeat (1) using the adjusted parameters such as queueing buffer size, code length, or code rate until the cell loss/error rate and the performance delay in coded case satisfy the service quality requirements.

For example, consider a VP_a as mentioned in the previous section with a cell arrival probability $p_a^{(1)} = 0.18$ which corresponds to an average link utilization $\eta = 0.9$. Suppose the services carried on VP_a require the cell loss/error probability $P_{LS} \leq 10^{-10}$. From Figure 3.9 (a), we can find the best range of y-axis code length is from 100 to 300. According to the cell loss distribution $P(u, v)$ shown in Figure 3.12, we find that the (127,119) RS code with $d_{min2} = 9$ may be suitable. Figure 4.10 shows the calculated the cell loss/error probability for coded cases. In Figure 4.10, when link utilization is 0.9, and using a (127,119) RS code combined with a buffer length of 15, the cell loss/error probability after decoding is 3.8×10^{-8} (compared with the uncoded case of 2.0×10^{-3}). The performance is not satisfied. However, if we extend the buffer size from 15 to 20, under the same traffic load condition and still using the same code, the cell loss/error probability after decoding is 3.3×10^{-13} (compared with the uncoded case of 5.4×10^{-4}). The improvement will be significant. From table 4.4, we know that using an (127,119) RS code only introduces about 2.8 ms extra delay compared with the uncoded case.

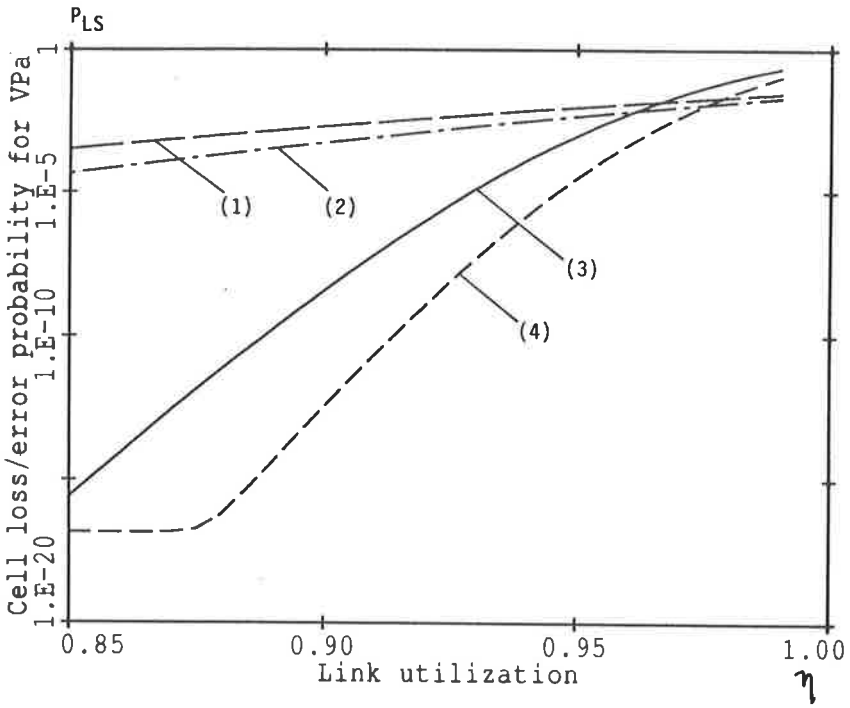


Figure 4.10 Cell loss/error probability for different combinations of coding schemes.

(1) Uncoded case with $K = 15$.

(2) Uncoded case with $K = 20$.

(3) (127,119) RS code with $K = 15$.

(4) (127,119) RS code with $K = 20$.

4.10 ARQ scheme for cell loss control at the cell level

As we pointed out in chapter 1, ARQ scheme has been often used in data network, and can provide very reliable transmission, but it also introduces additional delay and variation in delays. Miller [7] discussed the possibility of using ARQ scheme against the transmission errors for speech and data traffic in a packet-switched network under the condition of without packet loss due to buffer overflow. In this section, we first consider the performance of using a *M-try* ARQ scheme for cell loss control against the congestion at the cell level. Secondly, we compare our forward error recovery coding scheme with the *M-try* ARQ scheme.

4.10.1 *M-try* ARQ scheme

We first consider a *M-try* ARQ scheme [7] as an edge-to-edge error control approach for the virtual path connections or as an end-to-end approach for the virtual channel connections. The *M-try* ARQ is a selective-repeat ARQ scheme [32] except that each cell is only permitted up to M transmission attempts at achieving error-free transmission. In this scheme, as discussed in section 4.2, each cell is encoded with an (n_x, k_x) error detection code using *small distance* code, where the k_x information bits includes a cell sequence number of k_{x2} bits used for cell loss detection. Both the cell sequence number and parity redundancy bits are incorporated into the SAR sublayer in the ALL. At the transmitting edge of a virtual path (transmitting end for an end-to-end virtual channel connection), the coded cells are transmitted in order, and subsequently stored in a transmitter buffer because each cell is pending the return of a positive acknowledgement (ACK) or a negative acknowledgement (NAK) indication from the receiving edge at the outgoing of the virtual path (receiver end for an end-to-end virtual channel connection). On the receipt of a cell, the receiving edge uses the cell sequence number to determine whether a cell is lost, and uses the code parity bits to determine whether one or more errors are detectable in the received cell. It returns an ACK via a return path if no error or lost cell is detected. Otherwise, it returns a NAK. When the transmitting edge receives a NAK for a certain cell, C_{el} say, it retransmitts that cell in a selective-repeat mode [32]. At the receiving

edge, subsequent cells received error-free must be stored in a receiving buffer until they can be delivered in correct order. This retransmission process is repeated until either cell C_{el} has been received without detectable errors or until it has been transmitted a total of M times. If after M transmission attempts all copies of cell C_{el} are found to contain errors or found to be lost, then cell C_{el} is discarded or ignored.

Now we consider the performance for a *M-try* ARQ scheme. In following analysis, we assume that the return of ACK/NAK signal via a return path is idealized transmission without loss and transmission errors. We also assume that all lost cells can detected by cell sequence number (here we only consider the cell loss caused by congestion at the cell level). Let us define the cell loss/error probability P_{cd} as the probability that a cell is discarded or ignored because M transmission attempts fail. Then P_{cd} can be given by

$$P_{cd} = P_{LE}^{M}, \tag{4.36}$$

where P_{LE} is the loss/error probability including cells with transmission errors and cell loss due to buffer overflow given by equation (4.26), that is

$$P_{LE} = 1 - (1 - P_m)(1 - P_{loss}),$$

where

$$P_m = 1 - (1 - p_b)^{n_x} - \sum_{i=d_{min1}}^{n_x} A_i p_b^i (1 - p_b)^{n_x - i}$$

is the probability that a cell information field including the AAL contains errors which can be detected by the small distance code ($d_{min1} = 2$). P_m is give by equation (4.23). P_{loss} is the cell loss probability caused by congestion at the cell level for the observed VP, which given by equation (2.26) for random traffic obtained from chapter 2, or given by equation (3.20) for short burst traffic estimated by means of simulation in chapter 3.

Let $2\tau_a$ (second) be the round trip delay from a transmission of cell to the return of its ACK/NAK signal. Let D_1, D_2, ..., D_l, ..., D_M be the delay when the number of transmission attempts per cell is $l = 1, 2, ..., M$, respectively. Then D_l is given to a good

approximation by

$$D_l = \tau_a(2l-1), \tag{4.37}$$

and l transmission attempts have the probability density

$$P_D(l) = \begin{cases} (1-P_{LE})P_{LE}^{l-1} & \text{for } l = 1, 2, ..., M-1 \\ P_{LE}^{M} & \text{for } l = M \\ 0 & \text{otherwise} \end{cases} \tag{4.38}$$

Then the average delay is given by

$$\overline{D} = \sum_{l=1}^{M} D_l P_D(l). \tag{4.39}$$

The delay can never exceed $\tau_a(2M-1)$. Obviously, in this ARQ scheme, a pair of buffers of

$$\frac{R_a \tau_a (2M-1)}{n_x} \tag{4.40}$$

cells are required at both the transmitting edge and the receiving edge, respectively, where R_a is the transmission speed (bits/sec) of the virtual path or virtual channel, and n_x (bits) is the cell size.

Using a CRC-10 [38] code to replace the small distance code in the SAR of each cell, the *M-try* ARQ scheme becomes a *Hybrid M-try* ARQ scheme which is a Type I hybrid ARQ scheme [7]. A CRC-10 code with a minimum distance $d_{min1} = 3$ can correct one-bit random error and detect most other error patterns contained in the cell information field. On the receipt of a cell, the receiving edge uses the cell sequence number to determine whether a cell is lost, and uses the CRC-10 code to determine whether more than one-bit error is detectable in the received cell. It returns an ACK via a return path if lost cell or uncorrectable errors are not detected. Otherwise, it returns a NAK to request a retransmission.

The analysis of performance with a *Hybrid M-try* ARQ is similar to that of a *M-try* ARQ scheme, but just using $d_{min1} = 3$ to replace $d_{min1} = 2$ in the calculation of equation (4.36).

Since we are considering the performance of a VP which is modelled as a tandem queueing network with small size queues and high link utilization, in this case, in equation (4.36), P_m for both the *M-try* ARQ and *Hybrid M-try* ARQ is much smaller than P_{loss}. For example, for an optical fiber link with a bit error rate of 10^{-9}, and a cell information field of 384 bits, we have $P_m = 4.24 \times 10^7$ for the *M-try* ARQ scheme, and $P_m = 7.3 \times 10^{14}$ for the *Hybrid M-try* ARQ scheme. Likewise, for a given average link utilization $\eta = 0.85$, and queueing buffer size is $K = 20$, from the results shown in Figure 3.7 we have $P_{loss} \geq 2.4 \times 10^{-3}$. Obviously, P_m for both the *M-try* ARQ and *Hybrid M-try* ARQ are negligible by comparing them to P_{loss}. Therefore, we may conclude that under high link utilization conditions, the performance of using *M-try* ARQ and *Hybrid M-try* ARQ will be almost the same. In the following analysis, we only consider the performance with the *M-try* ARQ scheme.

4.10.2 Comparison of two-dimensional coding scheme and ARQ scheme

In this section, we present some specific numerical results obtained from the two-dimensional coding scheme and the *M-try* ARQ scheme. Both schemes are employed as an edge-to-edge approach for each individual virtual paths at the cell level. The bit error rate of the transmission link is 10^{-9}. In both schemes, for a cell information field of 384 bits including the payload field, we use an (384,379) *small distance* code [9] with 5 bits parity redundancy ($d_{min1} = 2$) for cell error detection. 4 bits as the cell sequence number and 5 bits as the parity redundancy bits are incorporated into the SAR sublayer [66]. We assume all lost cells can be detected by the cell sequence number. Since the cell header has its own error protection schemes [1], we are not consider the cell header error in this section. We note that the cell loss caused by cell header errors can also be detected by the cell sequence number, and retransmitted by ARQ scheme or recovered by the two-dimensional coding scheme. The probability of cell loss caused by cell header errors will be much smaller than the cell loss probability caused by buffer overflow, because the cell header has its own powerful error correction scheme.

The evaluations are based on the simulation results obtained from chapter 3, in which homogeneous mixtures of burst-silence traffic with short bursts are used (burst length is less than the queueing buffer size). The VP_a is assumed to be a virtual path contains

$Q = 5$ queues. Each queue along the path is modelled as a bulk arrival queue with $N = 5$ input VPs (including VP_a) and a finite buffer of $K = 25$. All VPs are coded with the two-dimensional coding schemes with the same codes. In the examples, the burst length is of $T_{on} = 10$ slots, the silence interval length is of $T_{off} = 2$ slots, and the source burstiness is 1.2. In the uncoded case, all VPs have the peak arrival rate of λ given by equation (3.3). The average link utilization is given by equation (3.8). The transmission speed of VP_a is R_a=150 Mbits/s. The round trip delay is assumed to be 5 ms (1,000 km edge-to-edge).

Firstly, we consider a situation with an average link utilization of $\eta = 0.85$. The cell loss/error probability of uncontrolled case is 1.4×10^{-4}. If the cell loss/error probability at the output of the receiving edge is required to be less than 10^{-8}, then for the two-dimensional coding scheme, as shown in Figure 4.9 and Table 4.5, we use a (169, 155) binary cyclic code as the y-axis code which reduces the cell loss/error probability to 5.6×10^{-16}. The coding delay (including the delay caused by receiver buffering and the queueing delay for cells transmitted through the network) is 0.83 ms. The scheme requires a buffer of 169 cells to assist the y-axis decoder at the receiving edge. By contrast, we use a *M-try* ARQ with M=2, which can reduce the cell loss/error probability to 1.9×10^{-8} (from equation (4.39)). However, this ARQ scheme requires a pair of buffers of size 2250 cells at both the transmitting edge and the receiving edge, respectively. The maximum delay (including queueing delay for cells transmitted through the network) is 12.9 ms and the average delay is 2.9 ms. We note that if the ARQ scheme is used as an end-to-end approach for a low-speed virtual channel connection, the buffer size may be smaller. The buffer size will depends on the transmission speed of the virtual channel.

Secondly, when the average link utilization increases to $\eta = 0.95$, the cell loss/error probability of uncontrolled case is 1.56×10^{-2}. If the cell loss/error probability at the output of the receiving edge is also required to be less than 10^{-8}. For the two-dimensional coding scheme, we use a (255, 233) RS code with a code rate of 0.9 as the y-axis code which reduces the cell loss/error probability to 5.9×10^{-8}. The coding delay is 6.1 ms. The scheme requires a buffer of 2040 cells to associate the y-axis decoder at the receiving edge. By contrast, using a *M-try* ARQ with $M = 4$ can reduce the cell loss/error probability to 6.25×10^{-8}. However, this ARQ scheme require a pair of buffers of size 5250 cells at both

the transmitting edge and the receiving edge, respectively. The maximum delay is 17.9 ms and the average delay is 3.1 ms.

With the comparison of the two-dimensional coding scheme and *M-try* ARQ scheme, we may conclude that:

1. The ARQ can provide very reliable transmission.
2. The ARQ scheme needs a return path for transmission of ACK/NAK signal.
3. Large buffers are involved at both the transmitting edge and receiving edge for the ARQ scheme. We note that when the ARQ is used as an end-to-end approach for low-speed virtual channel connection, the buffer could be smaller. On the other hand, the delay is also relative to the round trip delay. The performance of short distance connection with ARQ scheme is much better than the performance of long distance connection.
4. For the ARQ scheme, the margin between the maximum delay and the average delay is large. This may be not suitable for real-time services transmission in the ATM networks. However, ARQ scheme can be used for non-real time services due to their tolerance of delay.

Chapter 5

CELL LOSS CONTROL AT THE BURST LEVEL

Summary

In this chapter, we consider the procedures for cell loss control at the burst level. Firstly, a powerful method of cell loss control along a virtual path or virtual channel based on a joint usage of priority discarding of cells, two-dimensional forward error recovery coding, and interleaving is described. The application of this method allows one to decrease significantly the number of lost cells during the period of congestion at burst level which may occur due to traffic fluctuations during a call.

Secondly, we analyze the long-term performance with coding under burst overload conditions. The performance of a virtual path or virtual channel will be determined by the traffic sources which are present at all the nodes along the path through the ATM network. These bursty sources are modelled as homogeneous burst-silence processes with long burst length. Since the exact queueing evaluation for that model may not be tractable due to queue transient effects, in this chapter, we shall solve the problem with approximation.

The results will show that coding techniques can be a useful supplement to the conventional use of congestion control techniques. Such approach is of significant practical and theoretical value. It extends the range of loads and condition over which the ATM networks can be safely exploited.

5.1 Introduction

Congestion in an ATM network usually occur at 3 different levels: call, burst, and cell level [56]. A overload situation at the call level leads to call blocking whereas congestion in the lower levels may lead to a loss of cells. In connection with cell loss recovery, only the lower levels must be taken into consideration. At the cell level, congestion is caused by simultaneous cell arrivals occurring in a time span equivalent to a source inter-cell time [72]. At the burst level, congestion occurs when the total arrival rate, averaged over a period greater than an inter-cell time, exceeds the capacity of the output link [72].

As discussed in section 1.3, to flexibly support an entire range of existing and foreseeable services, and efficiently share the network resources, the ATM network will allow sources to access the network even though the total peak rate of all sources may exceed the link capacity [57]. The length of bursts permitted to enter the network may need to be controlled by a cell policing mechanism, but it is not clear whether to allow long bursts or short bursts [83]. The results obtained from chapter 3 have shown that short bursts can take advantage of queue dynamics to absorb losses, that is, using buffers of sufficient size in multiplexers and switches to absorb the short burst arrivals, and store cells for transmission when the links are free. In this case, the loss belongs to the cell level. The disadvantage is that requiring short bursts only will restrict the performance to the users. By contrast, the use of long bursts can give better user access, but it causes the network to suffer from burst collision giving long burst cell loss. In this case, the loss belongs to the burst level. The results shown in Figures 3.8, 3.10, and 3.13 have indicated that when long burst overload is involved, the cell loss-free interval between the two cell losses become narrower, and the cell loss due to buffer overflow will occur more frequently and in long bursts. In chapter 4, we have studied coding schemes for cell loss recovery at the cell level. The results have shown that the performance with the two-dimensional coding at the cell level is very good. However, at the burst level, the use of two-dimensional coding with long bursts does not make much impact. For the purpose of cell loss control using forward error recovery coding schemes at the burst level, we must consider other countermeasures in addition to the two-dimensional coding schemes.

The cell priority discard mechanism is a useful countermeasure for overload control, to transfer the cell loss onto lower priority services [79]. Two discard priorities have been recommended by CCITT [22]. A priority mechanism can guarantee high quality for the services with high priority, but must pay a high price of higher loss for the low priority services.

In this chapter, we study the application of forward error recovery coding scheme for cell loss control at the burst level based on a joint usage of priority discard of cells, two-dimensional forward error recovery coding, and interleaving techniques. Our study is again based on a tandem queueing network consisting of Q single server queues in series. Each queue receives traffic from N homogeneous burst-silence sources as discussed in section 3.2. The burst length is greater than the queueing buffer size, and the peak rate of overall arrival traffic exceeds the output link capacity. In this case, the loss belongs to the burst level.

5.2 Strategy of cell loss control for burst level congestion

The proposed strategy for cell loss control at the burst level is based on a joint usage of three approaches: priority discarding of cells, two-dimensional coding, and interleaving.

1. The admissible traffic is divided into 2 classes corresponding to different cell loss rate, where the high-priority traffic has the high-quality requirement while the low-priority traffic has the the medium-quality requirement. When the peak rate of all arrivals exceeds the link capacity, cells with low priority will be discarded using a partial sharing buffer mechanism [45]. This is to transfer the loss from high to low priority traffic.

2. Each virtual path or virtual channel with low priority is covered by a two-dimensional coding scheme combined with a interleaving/deinterleaving pair. The discarded cells will be recovered at the receiver end.

3. Since the discarded cells may appear in long bursts particularly under burst overload conditions, the interleaver/deinterleaver pair located at the transmitter and receiver

end, respectively, is used to break up and distribute the burst cell loss patterns so that to the decoder, they appear in short bursts or randomly distributed [32].

Now we describe the strategy in details as follows.

5.2.1 Priority discarding of cells

Estimation of instantaneous traffic load

Recall the definition of $P(j)$ in section 2.5, that is the probability that j cells are waiting in a switch output queue, where $0 \geq j \geq K$, K being the buffer size. $P(j)$ is a function of the instantaneous cell arrival process. Figure 5.1 and Table 5.1 show the simulation results of $P(j)$ versus j for a given set of peak rate η_p of overall cell arrivals, where η_p is given by equation (3.9). An actual bulk arrival queue with $N = 50$ homogeneous burst-silence sources with average burst length of 500 slots and average silence interval length of 100 slots as described in chapter 3 is considered in the simulation.

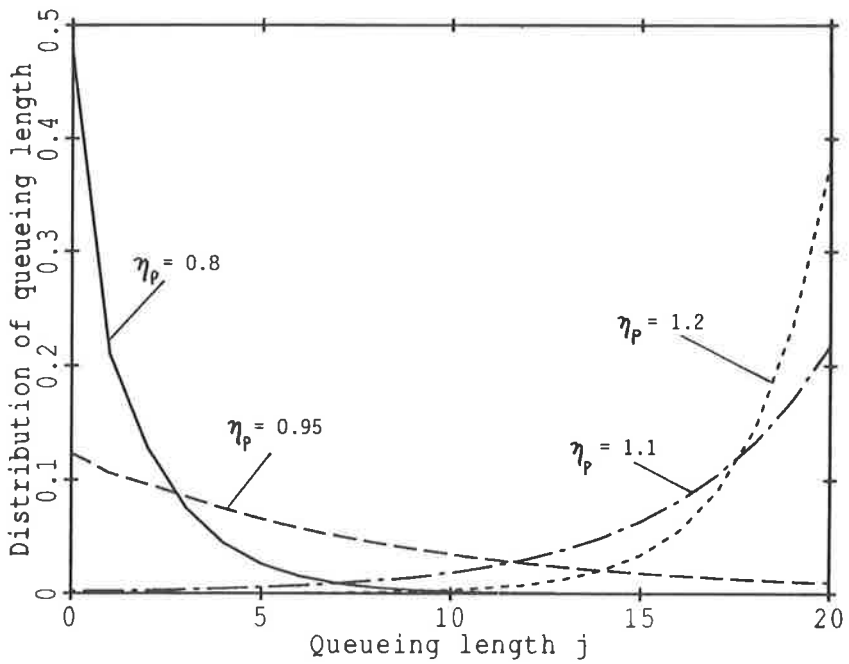


Figure 5.1 $P(j)$ versus j for different values of the traffic loads.

η_p	$P(0)$	$P(2)$	$P(4)$	$P(8)$	$P(12)$	$P(16)$	$P(18)$	$P(20)$
0.75	0.56	0.11	2.9E-2	2.0E-3	1.3E-4	8.9E-6	2.3E-6	6.0E-7
0.85	0.38	0.14	6.3E-2	2.9E-2	2.7E-3	5.4E-4	2.6E-4	1.2E-4
0.95	0.15	9.6E-2	7.5E-2	4.5E-2	2.7E-2	1.6E-2	1.3E-2	9.9E-3
1.05	1.3E-2	1.2E-2	1.7E-2	2.9E-2	4.7E-2	7.6E-2	0.10	0.13
1.10	2.0E-3	2.7E-3	4.4E-3	1.2E-2	3.1E-2	8.2E-2	0.14	0.22
1.15	2.6E-4	6.4E-4	9.2E-4	3.9E-3	1.7E-2	7.1E-2	0.15	0.31
1.20	3.1E-5	6.9E-5	1.8E-4	1.2E-3	8.3E-3	5.6E-2	0.14	0.38
1.25	3.7E-6	1.1E-5	3.4E-5	3.6E-4	3.9E-3	4.2E-2	0.13	0.45
1.30	4.3E-7	1.5E-6	6.3E-6	1.1E-4	1.8E-3	3.0E-2	0.12	0.51

Table 5.1 $P(j)$ versus j for different values of the peak traffic loads.

Figures shown in Table 5.1 indicate that the queueing length j is very sensitive to the instantaneous traffic loads. For example, let us consider a case of $j = 18$. From the Table 5.1, we see that $P(18)|_{\eta_p \leq 0.85}$ will be much smaller than $P(18)|_{\eta_p \geq 1.05}$. This means when the queueing length reaches $j = 18$, the probability that the instantaneous traffic with overload will be much greater than it with underload. By contrast, for $j = 4$, $P(4)|_{\eta_p \geq 1.05} << P(4)|_{\eta_p \leq 0.85}$. This suggests a possibility of using a threshold point along the queueing buffer to monitor the instantaneous traffic overloads.

Priority discarding of cells

When these two classes of traffic, high-priority traffic and low-priority traffic, come from different VPs, and join an ATM multiplexer, the buffer management in the switch becomes more complex and a priority discarding mechanism must be implemented. The priority discarding of cells at an ATM multiplexer has been considered in a number of proposals for traffic control. The first is the push-out mechanism proposed by Doshi and Heffes [86]. An arriving high-priority cell may enter a saturated queue where at least one low-priority cell is already waiting in the queue. Then one of the low-priority cells is discarded, and the arriving high-priority cell joins the queue. If the queue contains only high-priority cells, the arriving priority cell is discarded. On the contrary, low-priority cells cannot enter a saturated queue and are discarded. The push-out mechanism is the best method so far as there is no interference of the high priority cells by the low priority cells. They

appear to have the entire bandwidth to themselves, and so dimensioning can be done conservatively for those services giving top performance. By contrast, the low priority services suffer from much poorer performance. Since cell sequence must be preserved, the push-out mechanism requires a complicated buffer management logic [59].

The second solution is the partial buffer sharing mechanism proposed by Li [45]. When queue occupancy reaches a given threshold, only the high-priority cells may enter the queue. Obviously, the partial buffer sharing mechanism is less efficient than the "ideal" push-out mechanism since high-priority cells can be discarded while low-priority cells are still in the queue, but it is much simpler to implement.

Doshi and Heffes [86] have analyzed push-out mechanism with replacement strategy using M/M/1/N queue for Poisson arrivals traffic. Hou and Wang [85] have considered the use of a partial buffer sharing scheme with geometric arrivals for low loss priority and burst-silence sources for high loss priority. Kröner et al. [59] have compared various space priority mechanisms and their short-term queueing behavior and long-term characteristics.

At an ATM access multiplexer, the instantaneous queueing length is sensitive to the instantaneous traffic load. When the capacity actually used is less than or equal to the allocated capacity of the network link, the queueing length will be predominantly in a lower range. When the capacity actually used exceeds the link capacity, the queueing length will quickly build up and cell loss will increase.

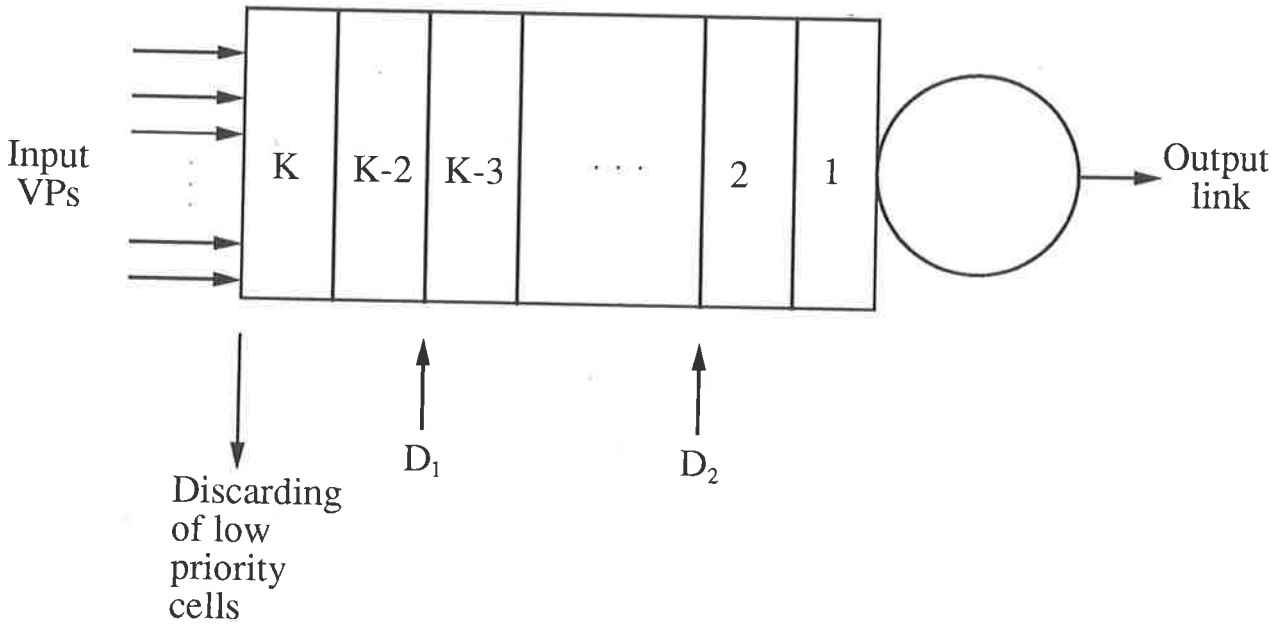


Figure 5.2 A diagram of controlled discarding of cells.

Now we consider an approach, called *priority discarding of cells,* based on a partial buffer sharing mechanism [45]. As shown in Figure 5.2, to guarantee a much better performance for high priority cells, we set up two threshold control points along the buffer at the D_1 and D_2 positions ($D_1 > D_2$), where D_1 is a position of queueing length j corresponding to the instantaneous traffic load with very large probability of full load or overload. Likewise, D_2 is a position of queueing length j corresponding to the instantaneous traffic load with very large probability of under allocated capacity. When the queueing length j exceeds the threshold point D_1, the switching node only allows the cells with high priority to enter the queue, and starts to discard the arriving cells with low priority, which are marked as a candidates for loss in case of congestion. According to current CCITT standard, 1-bit priority indicator has been reserved in the ATM cell header [66]. When the queueing length j returns back to the threshold point D_2 or below, the instantaneous traffic load will be equal to or less than the allocated capacity of the transmission link. The switching node will stop discarding the low priority cells. The positions of D_1 and D_2 are adjustable. These could, for example, depend on the statistics of the traffic carried on the VPs. If D_1 is lower, the action of discarding will occur more frequently. On the other hand, if D_2

is lower, each discarding action will be taken longer and more low priority cells will be discarded.

In following sections, we will assume all high priority can pass the tandem queueing network without loss based on above priority discarding mechanism.

In our priority discarding scheme, the non-real-time services such as data and image are preferred to be low priority, while real time services such as voice and video are preferred to be high priority. This is because:

1. The data and image services are tolerant of delay and their transmission rates are controllable during the service.

2. Burst overload is more likely to arise from the bursty data or image transfer applications, because their expected bandwidth may not be predictable [13]. Discarding of data or image cells can significantly reduce the instantaneous traffic load.

3. Data and image cells are tolerant of delay, so that interleaving/deinterleaving techniques could be applied. We note that priority discarding of cells combined with an end-to-end ARQ scheme is also appropriate for non-real-time services, because of their tolerance of delay. This is to tradeoff of cell loss for delays.

We note that this approach may also be used as a reactive control procedure. When the queueing length j exceeds D_1, the switching node starts to discard the low priority cells and sends an overload signal back to all input VPs to notify the virtual path controller to stop accepting new connections, and to request a reduction of the transmission rates. When the queueing length j returns to D_2 or below, the switch node will stop discarding cells and sends a underload signal back to the virtual path controller. The network can accept new connections according to the allocated capacity. However, as discussed in chapter 1, the reactive congestion control will have some significant disadvantages, and is considered to be inappropriate [78]. In the following sections, we assume that when the queueing length exceeds the threshold point, the switching node discards the low priority cells but does not send any choke cells to request a reactive control.

5.2.2 Two-dimensional codes combined with interleaving for loss recovery

As shown in Figure 5.3, a two-dimensional coding scheme combined with interleaving is proposed for recovery of lost cells. This coding scheme can operate end-to-end at the virtual channel connection level for individual services or edge-to-edge at the virtual path connection level for a group of services. Note that a two-dimensional coding scheme without interleaving can also be employed for services which are sensitive to both delay and loss such as video transfer applications.

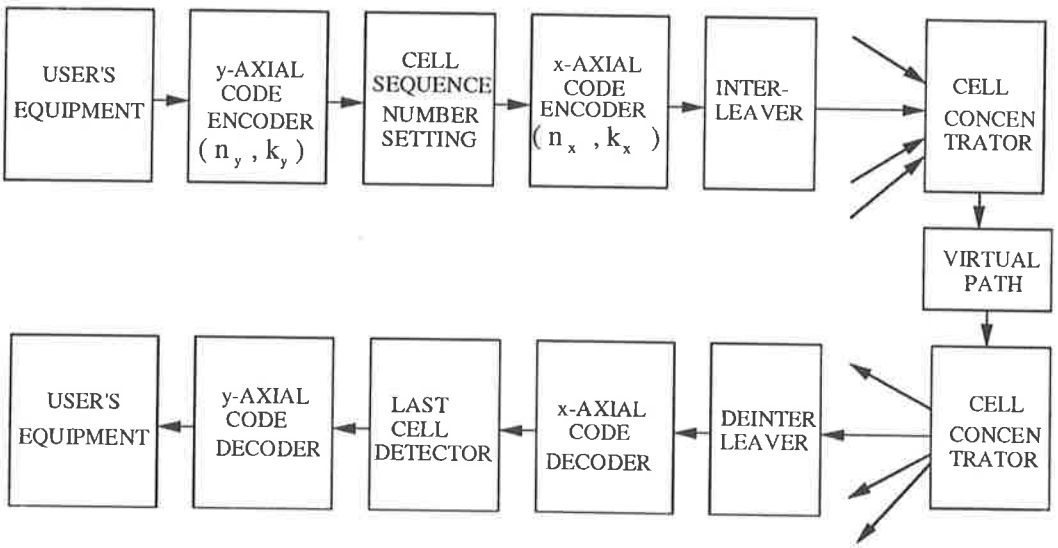


Figure 5.3 Two-dimensional code combined with interleaving.

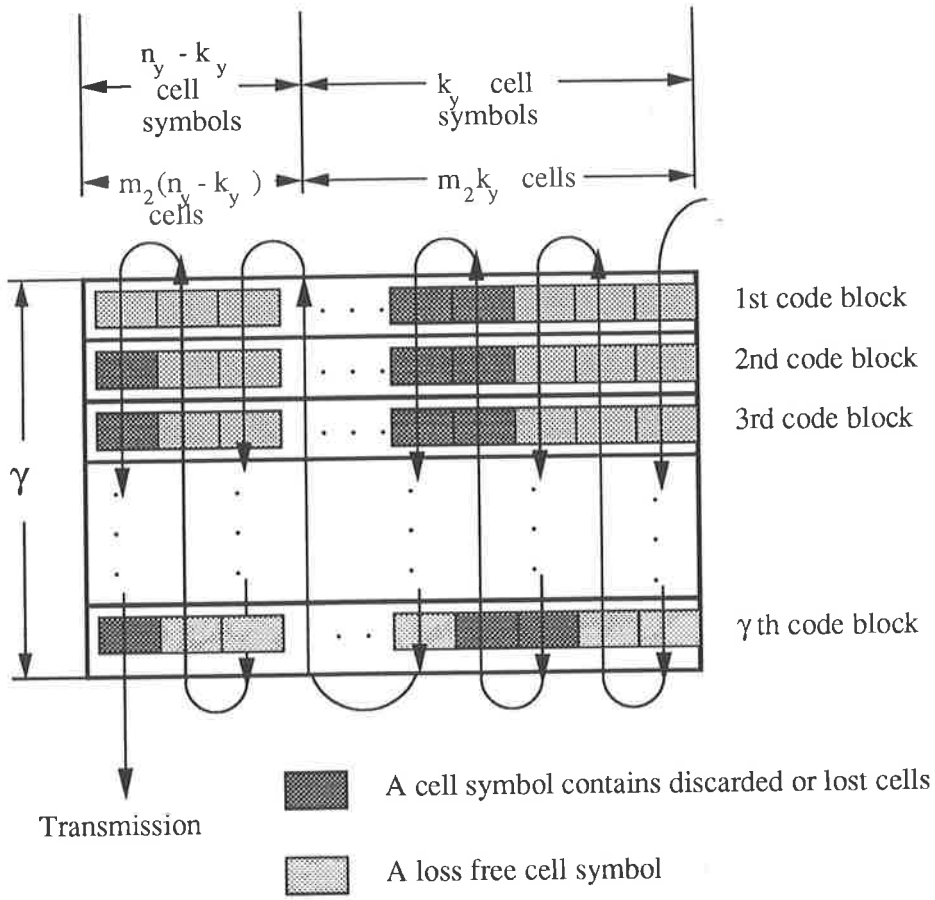


Figure 5.4 A diagram of symbol-by-symbol block interleaver.

Most forward error correction codes are designed to be effective when the transmission disturbances give rise to randomly distributed cell losses and errors. For performance at the burst level, we must consider the effects of buffer overflow due to overload. In this case, cell loss patterns will be in bursts. One way of approaching loss recovery in such a case is to combine coding with an interleaver at the transmit end and a deinterleaver at the receiver end.

For interleaving which associates a RS code (n_y, k_y) in $GF(2^{m_2})$, we consider using a symbol-by-symbol block interleaver/deinterleaver. This is done by arranging $\gamma m_2 n_y$ cells into γ rows of a rectangular array and then transmitting them column by column in cell symbols (each cell symbol consists of m_2 cells, and $m_2 = 1$ for binary code).

The purpose of the interleaver/deinterleaver pair is to break up and distribute the burst cell loss patterns so that to the decoder, they appear randomly distributed [32]. A given burst overload may affect a sequence of successive transmission cells. However, if the transmitted cells are interleaved, and after transmission the received cells are deinter-

leaved, this results in approximately independent disturbances of each cell when viewed from the deinterleaver output. Interleaving is a form of time diversity which requires no detailed knowledge of the system memory other than its approximate length. It may consequently be quite robust to change in cell loss statistics.

Interleaving involves delay, although this can be reduced by the use of a convolutional form of interleaving [32]. The delay will be directly related to the maximum length of the cell loss bursts to be recovered. Therefore the joint usage of priority discarding of cells, two-dimensional coding scheme combined with interleaving attempts to solve the problem of cell loss control at the burst level with limited burst size for all services. We should note that if high priority services are allocated bandwidth conservatively, the two-dimensional coding scheme without interleaving may be satisfactory to improve the loss performance.

5.3 Estimation of the performance with coding under burst overload conditions

In following sections, we will not consider the effects of cell error caused by transmission error. From equation (4.17) we have known that for an optical fiber with a bit error rate of 10^{-9}, the probability that a cell information field of 384 bits contains at least one bit error, will be less than 4.24×10^{-7}, and is negligible by comparing it to the cell loss probability caused by congestion at the burst level (see section 3.5.2). We will also not consider the cell loss caused by cell header errors, because the cell header has its own error protection scheme, the errors in the cell header can be ignored.

5.3.1 Uncoded case

Now we estimate the loss due to burst level congestion using a tandem queueing network with mixtures of homogeneous burst-silence traffic. As discussed in chapter 3, for an ATM multiplexer with bursty cell arrival process, the queue transient effect will be taken into account, and this leads to very complex queueing models [73]. An exact queueing analysis for this model seems not tractable and therefore the performance evaluation

must be done by simulation or approximation. In chapter 3, we have studied this model to obtain the statistics of cell loss due to cell level congestion using simulation. Simulation using Monte Carlo technique is only feasible for relatively high cell loss probability, due to limited computer resources (see chapter 3). In this chapter, we shall give an approximate analysis for the cell loss caused by congestion at the burst level, i.e., we neglect the effects of the buffers in the ATM network. A similar approach has been used in [21][59]. In this analysis, the overload period is assumed to be much greater than the queueing buffer size, so that the queue transient time will be much smaller than the whole overload period. In the long run, the cumulated transient time will be much smaller than the cumulated time of overload period. This calculated cell loss probability will appear worse than a more exact model which takes exact the queue transient effect into account, as cells would not be lost during the time from the instant of a burst overload arriving at the queue until the buffer fills. The approximate analysis is derived as following.

Consider a virtual path VP_a through a tandem queueing network consisting of Q single server queues in series operating on a first-in-first-out (FIFO) basis. The switching functions result in mixing of traffic from different input streams and splitting of traffic to different output streams.

A queue I ($I = 1, 2, ..., Q$) along the path VP_a is a bulk arrival queue which has a finite buffer of length K, and receives input from N independent homogeneous burst-silence sources as described in section 3.3. A designated A-stream is carried on the virtual path VP_a through the tandem queueing network. The designated A-stream departing the current node goes to the next node while the traffic carried on other VPs is assumed to leave the tandem queueing network immediately [61].

In this section, we will consider two cases: i) the loss probability for VP_a without priority discarding, i.e., all admissible traffic have the same priority. ii) the loss probability for VP_a with priority discard mechanism as described in section 5.2.2.

The cell loss probability for VP_a without priority discarding

Consider the queue I: the performance will be determined by the number of sources which are presented at the queues in an overload period. Recall the definition of burst-silence

source in section 3.3, from equation (3.6) we know a burst-silence source in burst state with probability

$$q = T_{on}/(T_{on} + T_{off}),$$

and silent with probability $1 - q$, where T_{on} is the average burst length and T_{off} is the average length of silence interval given by equation (3.4). Since the queue I is fed by N independent homogeneous burst-silence sources, the probability that ψ sources are in a burst state (are active) can be computed from a binomial distribution:

$$p(\psi) = \begin{pmatrix} N \\ \psi \end{pmatrix} q^{\psi}(1-q)^{N-\psi}. \tag{5.1}$$

The cell arrival rate, given that ψ sources in an active state, is given by $\psi\lambda$. The mean aggregate arrival rate can be expressed as

$$\eta = \lambda q N, \tag{5.2}$$

where λ is the cell arrival rate for a single burst-silence source in the burst state, and given by equation (3.3). The peak rate of the mixed traffic is

$$\eta_p = N\lambda. \tag{5.3}$$

Cells will be lost if the total cell arrival rate exceeds the link capacity. For a given state ψ, the aggregate loss is given by maximum $\psi\lambda - 1$. With this definition, the aggregate cell loss probability is given by

$$B = \frac{1}{Nq\lambda} \sum_{\substack{0 \le \psi \le N \\ \psi\lambda > 1}} p(\psi)(\psi\lambda - 1). \tag{5.4}$$

If random service among the cells arriving at the queue I in the same slot, the average probability for a cell from the designated A-stream and discarded at the queue I is given by

$$B_a^{(I)} = \frac{1}{Nq\lambda} \sum_{\substack{0 \le \psi \le N \\ \psi\lambda > 1}} p(\psi)(\psi\lambda - 1) \tag{5.5}$$

We assume that the interdeparture process of the designated A-stream is also a burst-silence process as it departs from the queue I and goes to join the next queue $I+1$. As discussed in chapter 2, this involves an independence assumption between the adjacent queues in the tandem queueing network. Over the extent of the tandem queueing network, the average cell loss probability for the designated A-stream is approximated by

$$P_{loss1} = [1 - \prod_{I=1}^{Q}(1 - B_a^{(I)})]/\lambda q \tag{5.6}$$

The cell loss probability for VP_a with priority discarding

Consider the queue I: here we assume that the N independent input burst-silence sources consist of N_1 sources with low-priority and N_2 sources with high-priority. Both the high-priority and low-priority sources have the same parameters as defined in section 3.3. The probability that ψ_i sources of priority i ($i = 1, 2$) are in a burst state (are active) can be computed from a binomial distribution:

$$p(\psi_i) = \begin{pmatrix} N_i \\ \psi_i \end{pmatrix} q^{\psi_i}(1-q)^{N_i - \psi_i}. \tag{5.7}$$

The cell arrival rate, given that ψ_1 sources of low priority and ψ_2 sources of high priority in an active state, is given by

$$(\psi_1 + \psi_2)\lambda.$$

The mean aggregate arrival rate can be expressed as

$$\eta = q\lambda(N_1 + N_2), \tag{5.8}$$

where λ is the cell arrival rate for a single burst-silence source in the burst state, and given by equation (3.3). The peak rate of the mixed traffic is

$$\eta_p = (N_1 + N_2)\lambda. \tag{5.9}$$

Cells will be lost if the total cell arrival rate exceeds the link capacity. For a given state (ψ_1, ψ_2), the aggregate loss is given by maximum $(\psi_1 + \psi_2)\lambda - 1$. With this definition,

the aggregate cell loss probability is given by

$$B = \frac{1}{(N_1 + N_2)q\lambda} \sum_{\substack{0 \le \psi_1 \le N_1 \\ 0 \le \psi_2 \le N_2 \\ (\psi_1 + \psi_2)\lambda > 1}} p(\psi_1)p(\psi_2)[(\psi_1 + \psi_2)\lambda - 1]. \tag{5.10}$$

Since we have assumed that the high priority cells are not lost under selective discarding mechanism, then the loss probability for a low priority cell, denoted as B_1, can be evaluated from the following conservation law for the aggregate loss probability B, that is

$$N_1 q\lambda B_1 = (N_1 + N_2)q\lambda B \tag{5.11}$$

If random discarding among the low priority cells arriving at a saturate queue in the same slot, the average probability for a cell from the designated A-stream and discarded at the queue I is given by

$$B_a^{(I)} = \frac{1}{N_1 q\lambda} \sum_{\substack{0 \le \psi_1 \le N_1 \\ 0 \le \psi_2 \le N_2 \\ (\psi_1 + \psi_2)\lambda > 1}} p(\psi_1)p(\psi_2)[(\psi_1 + \psi_2)\lambda - 1], \tag{5.12}$$

Over the extent of the tandem queueing network, the average cell loss probability for the designated A-stream with a low priority can be approximated by

$$P_{loss2} = [1 - \prod_{I=1}^{Q}(1 - B_a^{(I)})]/q\lambda. \tag{5.13}$$

5.3.2 Coding combined with interleaving

Now we consider the performance of coding combined with interleaving. In this section, we assume that all cells with high priority are not lost in the tandem queueing network. All sources with low priority, which join the queues along the path VP_a, are coded by the same two-dimensional code. Since we are not considering the effects of cell errors, in following analysis we only need to consider the performance with the y-axis code.

For the y-axis code using a RS code (n_y, k_y) in $GF(2^{m_2})$, $m_2(n_y - k_y)$ redundant cells are transmitted in a block of $m_2 n_y$ cells which contains $m_2 k_2$ information cells ($m_2 = 1$ for

a binary code). Then some extra traffic will be introduced into the network compared with the uncoded case. The multiplying factor for a low priority source coded with two-dimensional is be required to carry traffic with a rate

$$B_e = \frac{n_y}{k_y}.$$

The total average rate of the traffic load will be

$$\eta' = (B_e N_1 + N_2)\lambda q. \tag{5.14}$$

If all of the discarded low priority cells are detectable using a cell sequence number, then the total detectable cell loss probability P'_{loss2} in the presence of priority discarding can be obtained from equation (5.13) for a given set of parameters n_y, k_y, d_{min2}, Q, K, N_1, N_2, λ, and q. Note that P'_{loss2} represents the probability of cell loss P_{loss2} given by equation (5.13) with the expanded traffic load $\eta' = (B_e N_1 + N_2)\lambda q$.

We now consider the y-axis decoding. Given an (n_y, k_y) RS code in $GF(2^{m_2})$ as the y-axis code ($m_2 = 1$ for binary code), each two-dimensional code block consists of $m_2 n_y$ cells. Now we use a symbol-by-symbol block interleaver/deinterleaver. As shown in Figure 5.4. each interleaved block contains γ two-dimensional code blocks of $\gamma m_2 n_y$ cells. The interleaving is done by arranging $\gamma m_2 n_y$ cells into γ rows of a rectangular array and then transmitting them column by column in cell symbols, where γ is the depth of interleaving, and a cell symbol consists of m_2 cells which is a element in $GF(2^{m_2})$ (see section 4.4.2). In the following analysis, we assume that after deinterleaving, in each two-dimensional code block, all erasure symbols occur at random or in bursts with a length less than $(d_{min2} - 1)$ cell symbols, where d_{min2} is the minimum distance of the y-axis code. To satisfy this assumption, the maximum length of recoverable burst loss in a interleaved block must be less than $\gamma(d_{min2} - 1)$ cell symbols. All burst losses of length greater $\gamma(d_{min2} - 1)$ cell symbols are assumed to be unrecoverable.

Obviously, a pattern of cell losses can be recovered for the whole array if the pattern of losses in each row is a recoverable pattern for the original code. If an (n_y, k_y) y-axis code has minimum distance $d_{min2} = n_y - k_y + 1$, and can recover a burst of length $n_y - k_y$ cell symbols or less, which consist of lost cells. Then the interleaved code will recover a burst

of length $\gamma(n_y - k_y)$ cell symbols or less. In fact, a single erasure burst group of length less than or equal to $\gamma m_2(n_y - k_y) - m_2 + 1$ can affect at most $\gamma(n_y - k_y)$ consecutive cell symbols and so still be recovered. This is the maximum capacity, denoted as B_{vm}, of recovering from a single burst loss in an interleaved block. Then the B_{vm} can be given by

$$B_{vm} = \begin{cases} \gamma m_2(n_y - k_y) - m_2 + 1] & \text{for RS code in } GF(2^{m_2}) \\ \gamma(d_{min2} - 1) & \text{for binary codes} \end{cases} \tag{5.15}$$

($\gamma = 1$ for coding without interleaving).

The probability that the interleaved RS codes fail to correct erasure symbols in an interleaved block will be upper bounded by

$$P_r = \sum_{j=B_{vm}}^{n_y} A_j P_s^j (1 - P_s)^{n_y - j} \tag{5.16}$$

where A_j for $j = B_{vm}, (B_{vm} + 1), ..., n_y$ are the weight spectrum values for the RS code (e.g., the number of codewords of weight j). These can be given by

$$A_j = \begin{cases} 1 & \text{for } j = 0 \\ 0 & \text{for } 1 \le j \le \gamma(d_{min2} - 1) \\ \begin{pmatrix} q \\ j \end{pmatrix} \sum_{h=0}^{j-1-2t_2} (-1)^h \begin{pmatrix} j \\ h \end{pmatrix} (q^{j-2t_2-1} - 1) & \text{for } \gamma d_{min2} \le j \le q \end{cases} \tag{5.17}$$

where $q = 2^{m_2} - 1$, and A_j is the number of codeword with weight j. The derivation of this equation is given in the reference [50][52]. P_s is the symbol error probability of the RS code given by

$$P_s = \sum_{i=1}^{m_2} \begin{pmatrix} m_2 \\ i \end{pmatrix} {P'_{loss2}}^i (1 - P'_{loss2})^{m_2 - i} \tag{5.18}$$

where m_2 is the number of bits in a symbol of the RS code. Then the probability of uncorrectable erasures in a received block is approximately

$$P_{LS2} \approx \frac{P_r}{m_2} \tag{5.19}$$

Note that P_{LS2} includes the expanded traffic load due to transmission of redundant cells. In this case, the trade-off average link utilization is given by equation (5.2).

Interleaving/deinterleaving involves delay. If an interleaver/deinterleaver pair with a depth of γ is used, then one of buffer equal to $\gamma m_2 n_y$ cells will be required at the transmitter and receiver end. At the receiver end, the deinterleaving buffer can be combined with the receiver buffer which is used for y-axis decoding. The coded virtual path is assumed to have a transmission speed R_a, hence the total delay caused by interleaving/deinterleaving pair can be given by

$$\tau_l = \frac{\gamma 2 m_2 n_y}{R_a}. \tag{5.20}$$

5.3.3 Virtual path outage probability

In digital networks, the traditional bit error rate has been found inadequate for describing network performance, because it does not provide an accurate characterization of the distribution of errors with time. CCITT [88] has recommended to use the outage probability to represent the transmission quality and availability of long term performance in the digital networks. In the digital networks, the outage probability is defined as the probability that a specified bit error threshold is exceeded [87] [88]. We consider this concept for our network application. To estimate the long term performance for a virtual path through an ATM network suffering from burst overload, we introduce a concept of **virtual path outage probability** defined as the probability that the cell loss probability is greater than a given maximum acceptable value ε. The virtual path outage probability can be used to measure the service quality and availability for a particular VP under burst overload conditions. Normally, the network must be designed to operate under non-overload conditions with large cell loss/error margins to ensure that a specified maximum probability of outage is achieved.

Uncoded case

For the uncoded case, we will consider the virtual path outage probability for a virtual path VP_a without priority discarding of cells. The performance will determined by the number of sources which are present at the queues in an overload period. Consider a queue I (I = 1, 2, ..., Q) along the virtual path VP_a is fad by N independent homogeneous burst-silence sources. According to the definition of the burst-silence process described

in section 3.2.2. we have known that the cell arrival rate, given that ψ sources are in the burst state, is given by $\lambda\psi$, where λ is given by equation (3.3).

Obviously, when $\lambda\psi \geq 1$, the traffic will be overloading the link. For a given set of λ and N, in the long term performance, the average probability that overload occurs is given by

$$P_o = \sum_{\psi=\psi_s}^{N} p(\psi), \tag{5.21}$$

where ψ_s is the solution of equation $\lambda\psi = 1$, and $p(\psi)$ is the probability of ψ sources in the burst state among N sources and given by equation (5.1).

Now we consider the virtual path outage probability for VP_a. The cell loss probability for VP_a in the uncoded case without priority discarding of cells is given by equation (5.6). Using equations (5.6) and (5.5), for a given set of parameters λ, Q, K, and N, we can obtain a solution for the number ψ of burst-silence sources which are in a burst state, when outage occurs (i.e., $P_{loss1} \geq \varepsilon$). Let ψ_c be the solution, which is a critical value. The critical value must be a real number $0 \leq \psi_c \leq N$. If $\psi_c = N$, there is never any outage. If $\psi_c = 0$, outage occurs for all values of ψ. Obviously, outage occurs when ψ exceeds ψ_c. Hence, for all $\psi \geq \psi_c$, we have $P_{loss1} \geq \varepsilon$. Then the outage probability for the uncoded case can be given by

$$P_{out} = \sum_{\psi=\psi_c}^{N} p(\psi) \tag{5.22}$$

Coding combined with interleaving

Now we consider the virtual path outage probability for coding combined with interleaving. In this section, we assume that all cells with high priority are not lost in the tandem queueing network due to the use of priority discarding mechanism. All sources with low priority, which join the queues along the path VP_a, are coded with the same two-dimensional coding schemes combined with interleaving.

For a RS code (n_y, k_y) with interleaving of depth γ ($\gamma = 1$ for without interleaving), a received interleaved block of length $\gamma m_2 n_y$ ($m_2 = 1$ for binary code) is deemed to only contain one detected burst loss group. Firstly, recall the equations (5.19), (5.18), (5.17),

(5.16), and (5.15) for a given set of parameters λ, n_y, k_y, d_{min2}, Q, K, N_1, N_2, q, and $gamma$, we can obtain a solution for P'_{loss2} when outage occurs, that is

$$P_{LS2} \geq \varepsilon. \tag{5.23}$$

Let ε' be the solution, i.e., when $P'_{loss2} \geq \varepsilon'$, we have $P_{LS2} \geq \varepsilon$. Secondly, from the equations (5.14) and (5.13), we can obtain a number ψ_v of burst sources with low priority which are in an active state for $P'_{loss2} \geq \varepsilon'$. That is,

$$P'_{loss2} = [1 - \prod_{I=1}^{Q}(1 - B_a^{(I)})]/\lambda q \geq \varepsilon', \tag{5.24}$$

and

$$B_a^{(I)} = \frac{1}{N_1 q \lambda} \sum_{\substack{0 \leq \psi_1 \leq \psi_v \\ 0 \leq \psi_2 \leq N_2 \\ (\psi_1 + \psi_2)\lambda > 1}} p(\psi_1)p(\psi_2)[(\psi_1 + \psi_2)\lambda - 1]. \tag{5.25}$$

The solution ψ_v is also a critical value. If $\psi_v = N_1$, there is never any outage. If $\psi_v = 0$, outage occurs for all values of ψ_1.

Now we assume that for a given value of ψ_1, the duration length M_{ψ_1} of the overload period, in which ψ_1 sources with the low priority are on active, obeys a negative exponential distribution. Its probability density function is denoted as $\Upsilon(M_{\psi_1})$. For an RS code, if $M_{\psi_1} \geq \gamma m_2 n_y$, ($m_2 = 1$ is for binary code), obviously, outage occurs when $\psi_1 \geq \psi_v$. For $0 \leq M_{\psi_1} < \gamma m_2 n_y$, in an interleaved block, if it at most only contains one overload period of length M_{ψ_1}, and

$$P'_{loss2} \times \lambda M_{\psi_1} \leq B_{vm} \qquad (M_{\psi_1} < \gamma m_2 n_y), \tag{5.26}$$

then the burst of cell loss/error still can be recovered even if $\psi_1 \geq \psi_v$. In the latter case,

$$M_{\psi_1}|_{max} = \frac{B_{vm}}{\lambda P'_{loss2}} \tag{5.27}$$

Overall, outage occurs for all $M_{\psi_1} \geq M_{\psi_1}|_{max}$. Thus the outage probability can be given

by

$$P_{out} = \sum_{\psi_1=\psi_v}^{N_1} \int_{M_{\psi_1}|_{max}}^{\infty} \Upsilon(M_{\psi_1}) p(\psi_1) dM_{\psi_1}. \tag{5.28}$$

5.4 Illustrative schemes

Now we present some specific numerical results obtained from the joint usage of priority discarding of cells, two-dimensional coding, and interleaving. We assume that no cells with high priority are lost in the tandem queueing network due to the use of the priority discarding mechanism. All traffic with low priority is coded with the same two-dimensional codes and associated with interleaving. The parameters of the two-dimensional code are the same as considered in section 4.8. All the results of performance with coding shown in this section are tradeoff of the average link utilization. The transmission speed of each virtual path is assumed to be 150 Mbit/s. A virtual path VP_a with low priority is considered based on a given tandem queueing network, described by its parameters $Q = 5$, $K = 25$. Each queue in the tandem queueing network is fed by $N = 50$ independent homogeneous burst-silence sources including $N_1 = 40$ sources with low priority and $N_2 = 10$ sources with high priority. The fraction of high-priority traffic is taken as $N_2/N = 0.2$. Figure 5.5 illustrates the cell loss/error probability versus the average link utilization for coding combined with different interleaving depths. All traffic sources which join the queues along the path are homogeneous burst-silence sources with an average burst length of $T_{on} = 500$ slots, and average silence interval of $T_{off} = 100$ slots. The source burstiness is 1.2.

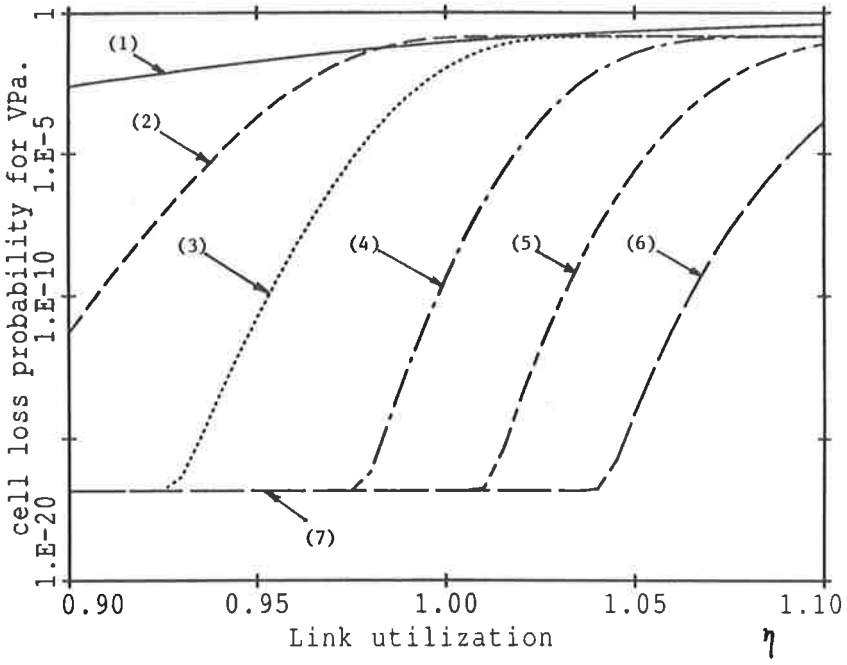


Figure 5.5 Average cell loss/error probability versus the average link utilization.

(1) Uncoded.

(2) (127,119) RS code with $\gamma = 1$.

(3) (127,119) RS code with $\gamma = 2$.

(4) (127,119) RS code with $\gamma = 3$.

(5) (127,119) RS code with $\gamma = 4$.

(6) (127,119) RS code with $\gamma = 5$.

(7) Undetected cell loss/error probability P_{ue}.

Code	T_{on}(slots)	T_{off}(slots)	ϵ	P_o	P_{out1}	B_{lm}	γ	τ_I (ms)	B_s	P_{out2}	B_{vm}
Uncoded	30	6	10^{-6}	1.5×10^{-5}	0.32	0	0	0	0	0.32	0
(169,155) Block code	30	6	10^{-6}	2.6×10^{-4}	2.2×10^{-3}	6	5	4.8	845	7.6×10^{-6}	30
(103,88) Block code	30	6	10^{-6}	1.4×10^{-3}	0.01	8	5	2.9	515	7.3×10^{-4}	40
(127,119) RS code	30	6	10^{-6}	1.1×10^{-4}	4.7×10^{-4}	50	5	25.1	4445	1.8×10^{-12}	274
(127,115) RS code	30	6	10^{-6}	2.6×10^{-4}	9.7×10^{-5}	67	5	25.1	4445	8.0×10^{-13}	414
(255,233) RS code	30	6	10^{-6}	2.6×10^{-4}	1.5×10^{-6}	169	5	57.6	10200	1.1×10^{-22}	873

(a)

Code	T_{off}(slots)	T_{off}(slots)	ϵ	P_o	P_{out1}	B_{lm}	γ	τ_I (ms)	B_s	P_{out2}	B_{vm}
Uncoded	500	100	10^{-6}	0.13	0.68	0	0	0	0	0.68	0
(169,155) Block code	500	100	10^{-6}	0.19	0.31	6	5	4.8	845	1.2×10^{-2}	30
(103,88) Block code	500	100	10^{-6}	0.23	0.42	8	5	2.9	515	5.8×10^{-2}	40
(127,119) RS code	500	100	10^{-6}	0.17	6.7×10^{-2}	50	5	25.1	4445	3.2×10^{-9}	274
(127,115) RS code	500	100	10^{-6}	0.19	5.7×10^{-2}	67	5	25.1	4445	2.4×10^{-9}	414
(255,233) RS code	500	100	10^{-6}	0.19	3.0×10^{-3}	169	5	57.6	10200	7.6×10^{-20}	873

(b)

Table 5.2 Effects of coding combined with interleaving.

P_o is the overload rate.

P_{out1} is the outage probability without interleaving.

B_{lm} is the maximum length of lost cells in a received code block which can be recovered.

B_s is interleaving block size.

γ is the interleaving depth.

τ_I is the interleaving delay.

P_{out2} is the outage probability with interleaving.

B_{vm} is the maximum length of lost cells in an interleaved block which can be recovered.

Table 5.2 shows the virtual path outage probability, the maximum capability of recovering from the long burst losses, and interleaving delay for various combinations of coding schemes with different burst length. The virtual path outage probability is calculated based on an acceptable cell loss probability of 10^{-6} for the low priority traffic, i.e., cell loss probability for low-priority traffic must be less than 10^{-6} [59]. The calculated virtual path outage probability for both coding with and without interleaving are illustrated. For a given value of ψ_1, the duration length M_{ψ_1} of an overload period, in which ψ_1 sources with the low priority are on active, obeys a negative exponential distribution with a probability density

$$\Upsilon(M_{\psi_1}) = \frac{1}{\overline{M}_{\psi_1}} e^{-\frac{M_{\psi_1}}{\overline{M}_{\psi_1}}},$$

where $\overline{M}_{\psi_1}$ is the mean of M_{ψ_1}.

In Table 5.2(a), we consider the performance with medium average burst length of $T_{on} =$ 30 slots and average silence interval of $T_{off} = 6$ slots. $\overline{M}_{\psi_1}$ is assumed to be 20 cells. In Table 5.2(b) we consider the performance with large average burst length of $T_{on} = 500$ slots and average silence interval of $T_{off} = 100$ slots. $\overline{M}_{\psi_1}$ is assumed to be 300 cells. The burst length obeys a geometrical distribution, and the silence interval length is negative exponentially distributed given by equations (3.1) and (3.2), respectively.

5.4.1 Discussion

1. The joint usage of priority discarding of cell, two-dimensional coding, and interleaving can be used for cell loss control under burst overload condition. For example, as shown in Figure 5.5, for an average link utilization of $\eta = 1.05$, using a (127,119) RS code combined with interleaving of depth $\gamma = 4$, the cell loss probability for low priority traffic can be reduced to 5.6×10^{-7} (compared to uncoded case of 0.34). However, to achieve this coding gain, a price of 21 ms delay must be paid. We note that the high priority traffic is considered to be loss free due to priority discarding.

2. When burst length is greater than the buffer size but not too large, and overload is not very heavy. In this case, coding without interleaving may make some

contributions to the network performance. If the codes are chosen properly, the coding efficiency is quite good. For example, as shown in Table 5.2 (a), each burst-silence source has an average burst length of $T_{on} = 30$ slots and average silence interval length of $T_{off} = 6$ slots. In the uncoded case, if transmission suffers from burst overload traffic with a rate of $P_o = 1.5 \times 10^{-5}$, the outage probability is 0.32 for a specific limit of cell loss rate $\varepsilon = 10^{-6}$. If an (127,119) RS code without interleaving is used, the outage probability will drop to 4.7×10^{-4}.

3. When the burst length is much greater than the buffer size, and heavy overload is involved. The interleaving techniques must be used to associate the coding schemes. The results shown in Table 5.2 (b) show that when the overload rate is 0.13, in the uncoded case, the outage probability is 0.68 for $\varepsilon = 10^{-6}$. Using coding combined with interleaving of depth $\gamma = 5$, the change is significant. For example, an (127,119) RS code combined with interleaving of a depth of 5, can recover any pattern or fewer of 274 lost or errored cells in a interleaved block of 4445 cells. The outage probability with interleaving is 3.2×10^{-9}. However, to achieve this benefit a price must be paid for a low priority VP with an interleaving delay of 25 ms.

4. The disadvantage of the use of interleaving is the need of a pair of large buffers of size $\gamma m_2 n_y$ cells at both the transmitter and receiver end, and relative delay, although the deinterleaving buffer can be combined with the receiver buffer for the y-axis decoding. To overcome this disadvantage, as discussed in section 5.2.1, the non-real-time services such as data and image are preferred to be low priority because of their tolerance of delay.

Chapter 6

CONCLUSIONS

In this thesis we have studied the use of forward error recovery coding techniques for recovery of cell loss against the network congestion.

6.1 Cell loss caused by network congestion

In an ATM network, congestion may be considered at 3 different levels: i) call connection level; ii) burst level; iii) cell level. A overload situation at the call connection level leads to call blocking whereas congestion at the lower levels leads to loss of cells. In connection with recovery of cell loss, only the lower level congestion need be taken into consideration. It is appropriate to distinguish the effects which occur at the burst and cell levels, in particular:

- at the burst level, congestion occurs when the total arrival rate exceeds the capacity of the output link averaged over a period greater than an inter-cell time of a referenced source.
- at the cell level, congestion is caused by simultaneous cell arrivals occurring over a time span equivalent to a source inter-cell time.

By considering a virtual path as a tandem queueing network consisting of a number of queues in series, it is possible to characterize the cell loss due to buffer overflow in terms

of the network and traffic parameters. Each queue along the path is modelled as a bulk arrival queue with a finite buffer. Our studies have concentrated on a designated traffic stream as it goes through the tandem network. Two different cell arrival processes are used for traffic models: i) constant rate deterministic arrival process as the random traffic, and ii) homogeneous burst-silence process as the bursty traffic.

For a VP with constant rate deterministic processes, the analysis is approximate, but it was shown to be quite accurate by comparing it to the simulation results. The exact queueing analysis for a VP with mixtures of the more realistic homogeneous burst-silence sources leads to a very complex queueing model due to queue transient time effects, and seems not tractable. Therefore, its performance evaluation has been done by simulation. In the simulation, an actual tandem queueing network has been used to obtain the performance measures for the designated traffic stream as it goes through the network. That is,

- We have introduced no "independence assumption" between the adjacent queues.
- Interdeparture process has been used to observe the designated traffic stream as it goes through the network.
- The queue transient time effects have been taken into account.

Cell loss due to buffer overflow may occur in bursts. To properly understand the statistical dependence that occurs in the cell loss patterns is a very important step toward obtaining more meaningful evaluations of error control codes and their suitability for this purpose. High order statistical distributions of burst cell loss patterns such as cell loss-free run distribution and cell loss distribution are estimated by analytical approximation or simulation for both random and bursty traffic. These results are subsequently used as a basis for analyzing the performance with forward error recovery coding schemes. The conclusions we derived from a study of the statistics of cell loss can be summarized as follows.

1. Cell loss caused by congestion is considered at two different levels: cell level and burst level. The performance of a VP with random traffic, short bursts (i.e., the

average burst length is less than the queueing buffer size), or long bursts without overload, belong to the cell level. The performance of a VP with long burst overload belongs to burst level.

2. At the cell level, extending the queueing buffer size can reduce the cell loss probability for an individual VP, but the improvement becomes smaller as the average link utilization increases toward the link capacity. This situation may be overcome by using larger buffers. However, larger buffers increase both cell delay and the variations in cell delay. This can cause difficulties for real time service which require synchronization between the transmitter and receiver. Therefore, it is desirable to consider other countermeasures such as the use of error correction to improve the loss performance under high link utilization conditions.

3. At the cell level, extending the queueing buffer size can significantly increase the cell loss-free intervals between the burst losses, even for high link utilization. This leads to a possibility of using the error correction techniques combined with suitable queueing buffer size to improve the network performance.

4. At the burst level, using a larger buffer to reduce the cell loss rate will be insignificant because long bursts with overload are involved.

5. At the burst level, extending the queueing buffer size may also increase the cell loss-free intervals between the burst losses, but this improvement is limited. On the other hand, the cell loss pattern distribution has shown that the cell loss will occur in bursts. However, for the purpose of cell loss control at the burst level, a proposal based on a joint usage of priority discarding of cells, forward error recovery coding, and interleaving techniques have described and analyzed.

6. Under overload conditions, congestion is sensitive to the burst length. If it is desired to avoid burst level congestion, the cell policing mechanism must control the average burst length to be less than the queueing buffer size. However, it has not yet been decided whether to allow long bursts or short bursts. For different burst lengths, different congestion control protocols have to be considered.

6.2 Forward loss recovery at the cell level

In this thesis we have also examined whether forward error correction coding techniques can be employed effectively to supplement general congestion control countermeasures at the cell level. A forward error protection scheme is proposed for error recovery in ATM network. This is done by a two-dimensional coding scheme, in which a high rate code is used for individual cell error detection, and an erasure correction code is used for lost or damaged cell recovery. In order for this scheme to work, a cell sequence number is required for lost cell detection. According to the current CCITT standards, the cell sequence number and the parity bits used for cell error detection, can be incorporated into the 14 reserved bits (4 bits for cell sequence number and 10 bits for cell error detection) in the ATM adaptation layer in the payload field.

This forward error recovery coding scheme can be used end-to-end for virtual channel connections. It can also be used for individual VPs edge-to-edge at the virtual channel connection level. The encoding and decoding can be done in parallel, resulting in encoders and decoders which would be fast enough to match the cell transmission speed over the transmission link.

The ATM network is conceived to provide an integrated access that will support a wide variety of applications for its customers in a flexible and cost-effective manner. This requires the ATM network to have high utilization and efficiency and limits the amount of code redundancy available for cell error detection, cell error correction, and lost cell recovery. We have therefore concentrated on efficient applications of error correction codes. Since the optical fiber links involve a low bit error rate of the order of one in 10^{-9}, we use a class of high rate, less powerful codes referred as *small distance* codes [9]. These codes can be used to detect all single-error patterns and large fraction of double-error and other patterns. For example, if the bit error rate of the optical link is 10^{-9}, and a cell information field, including the payload field of 48 octets (384 bits), a (384,379) small distance code with only 5 parity bits of redundancy will give the probability that a cell contains undetectable errors, as 1.5×10^{-17}. This is contrasted to the uncoded case with the probability of 4.24×10^{-7} that a cell contains transmission errors. The code rate in

this case is 0.99. These codes were chosen also because of the ease of decoding at high bit rate.

For lost cell recovery, we considered three different types codes: binary cyclic code, and Reed-Solomon (RS) code. The RS code is non-binary code suitable for recovery of bursty cell loss which we showed to be predominant in the ATM networks.

The performance with coding was estimated for both random and short burst traffic using the statistics of cell loss obtained by analytical method and simulation in chapters 2 and 3. Our studies have analyzed the tradeoffs involved in using coding techniques. A summary of relative cell error/loss rate, coding delay, bandwidth efficiency, and the capability of recovering from long cell loss bursts has been given in Figures 4.8 and 4.9, and in Tables 4.4 and 4.5 for a number of suitable codes.

The results indicate that if the codes are chosen properly, cell loss probability ranging from 10^{-10} to 10^{-17} can be achieved for an average link utilization of 0.9 with a short queue of $K = 25$ for both random traffic and short burst traffic. Even for a high average link utilization, say $\eta = 0.95$, using a (255,233) high rate RS code, cell loss rate remains less than 10^{-16} for random traffic, and 10^{-8} for short burst traffic. However, it should be noted that codes used to improve the performance under high link utilization conditions must be very carefully examined. A more powerful code generally has a lower code rate, because of the large amount of parity redundancy. Under heavy traffic load conditions, the extra traffic due to transmission of the parity check cells may introduce more cell losses than the benefits obtained from the coding schemes.

To properly design the coding schemes, it is necessary to carefully examine the statistics of cell loss due to network congestion. The principles of designing the suitable coding schemes were discussed in section 4.9. This involves choosing the code word length, code rate, and suitable queueing buffer sizes utilizing the statistics of cell loss obtained from chapters 2 and 3.

6.3 Cell loss control at the burst level

Finally, the strategies of forward error control coding techniques for cell loss control at the burst level were investigated. A powerful method of forward error recovery along a virtual path or virtual channel based on a joint of usage of priority discarding of cells, two-dimensional coding, and interleaving is described and evaluated. The average cell loss/error rate, relative delays, virtual path outage probability, and the maximum length of the overload period which can be handled are given in Table 5.2. The results have indicated that

1. When the burst length is just greater than the buffer size, coding without interleaving may make some improvement to the network performance. If the codes are chosen properly, the coding gain is acceptable. At the burst level the overload probability is related to the peak utilization of the transmission link. Control of the peak utilization by the cell policing mechanism is also a useful approach to avoid burst overload.
2. When the burst length is much greater than the buffer size, interleaving techniques would need to be used in association with the coding schemes. These techniques however significantly increase delay and memory requirements.

This method allows one to significantly decrease the number of lost cells and recover from errors during periods of short-term overload which may occur due to traffic fluctuations during a call. It is shown that coding techniques can be a useful supplement to the conventional use of generic congestion control techniques.

6.4 Future work

Finally, the author would like to make some suggestions for the future study in this area.

1. Cell loss due to network congestion is sensitive to the traffic statistics, i.e., traffic load and burst distribution. The studies in the literature has rarely addressed these statistics, and the average loss rate that has been mainly studied so far, does not give enough information. Studies are needed understand the characteristics of the

multiplexed traffic which are carried on VPs, and to establish better models for traffic with short bursts or other traffic types in order to determine equivalent measures for cell loss statistics by simulation or approximation. These are the most important steps toward the evaluation of the design of coding schemes.

2. The implementation of the proposed schemes in hardware, and the synchronization of the coding, needs more attention in future work.

3. An ATM network will interconnect to other digital networks, such as narrow band ISDN, satellite networks, mobile networks, and digital radio network. In these networks the available bandwidth, and bit error rate of the transmission link will be different from that of optical fiber. For an end-to-end transmission or edge-to-edge transmission, the lost cell recovery and cell error correction will become significant topics for future study.

Appendix A

Erasure decoding procedures for RS codes

Erasure correction for RS codes mainly involves the solving of a set of simultaneous linear equations constructed by substituting a primitive field element and its power into received code sequence. The actual procedures are explained as follows [50][51].

Now consider a (n_y, k_y) RS code defined in $GF(2^{m_2})$ with a minimum distance d_{min2} is used as the y-axis code for erasure correction. Then we have that

$$n_y = 2^{m_2} - 1,$$

$$k_y = 2^{m_2} - d_{min2} - 1,$$

$$n_y - k_y = d_{min2} - 1.$$

Let $\overline{r}_y(y)$ be the transmitted code sequence and $\overline{e}(y)$ be the added error pattern. Furthermore, let us assume that $\overline{e}(y)$ has values d_1, d_2, ... , d_V at locations y_1, y_2, ... , y_V $(V \leq (n_y - k_y))$, respectively.

The first step in decoding is to calculate $(n_y - k_y)$ parity check sum $\mathcal{S}_i^{(1)}$ from the received code sequence $\overline{r'}_y(y)$. This is done by

$$\mathcal{S}_i^{(1)} = \overline{r'}_y(a^i) \tag{A.1}$$

$$= \overline{r}_y(a^i) + \overline{e}(a^i) \tag{A.2}$$

where a is a primitive element for $i = 1, 2, ..., (n_y - k_y)$. A legitimate code sequence has $a^1, a^2, ..., a^{n_y - k_y}$ as its roots. Then we have

$$\overline{r}_y(a^i) = 0 \qquad \text{for } i = 1, 2, ..., (n_y - k_y) \tag{A.3}$$

Therefore,

$$\mathcal{S}_i^{(1)} = \overline{e}(a^i) \tag{A.4}$$

$$= \sum_{j=1}^{V} d_j y_j^i \qquad \text{for } i = 1, 2, ..., (n_y - k_y) \tag{A.5}$$

Since the error location number y_j's which are detected by the x-axis code or cell sequence number are known to the y-axis code decoder, then the equations (B.4) is a set of linear independent equations which can be used for evaluating the erasure values d_j's. As long as V is less than or equal to $(n_y - k_y)$, there is a unique set of solutions.

Another alternative is to evaluate the unknowns interactively which requires the introduction of a set of elementary symmetric functions Z_h [32] for the erasure location numbers which are given as

$$\prod_{i=1}^{V}(y + y_i) = \sum_{h=0}^{V} Z_h y^{V-h} \tag{A.6}$$

Now suppose we wish to solve for d_j, an erasure at location y_j. We will start by defining a new term Z_{jh} as a set of symmetric functions excluding y_j. That is

$$\prod_{i \neq j}^{V}(y + y_i) = \sum_{h=0}^{V-1} Z_{jh} y^{V-h-1} \tag{A.7}$$

Observing that equations (B.6) and (B.7) differ only by the term $(y + y_j)$, then we can obtain that

$$(y + y_j) \sum_{h=0}^{V-1} Z_{jh} y^{V-h-1} = \sum_{h=0}^{V} Z_h y^{V-h}. \tag{A.8}$$

Equating coefficients we arrive at the relation

$$Z_h = Z_{jh} - y_j Z_{j(h+1)} \tag{A.9}$$

Now combining equation (B.4) and (B.9), we have

$$\sum_{h=0}^{V-1} Z_{jh} \mathcal{S}_{V-h}^{(1)} = \sum_{h=0}^{V-1} Z_{jh} \sum_{i=1}^{V} d_i y_i^{V-h}$$

$$= \sum_{i=1}^{V} d_i y_i \sum_{h=0}^{V-1} Z_{jh} y_i^{V-h-1} \tag{A.10}$$

However, recall the definition of the parity check sum, the sum on the right hand side of equation (B.10) is nonzero only when $i = j$, hence

$$\sum_{h=0}^{V-1} Z_{jh} \mathcal{S}_{V-h}^{(1)} = d_j \sum_{h=0}^{V-1} Z_{jh} y_j^{V-h} \tag{A.11}$$

Finally, we can obtain

$$d_j = \frac{\sum_{h=0}^{V-1} Z_{jh} \mathcal{S}_{V-h}^{(1)}}{\sum_{h=0}^{V-1} Z_{jh} y_j^{V-h}} \tag{A.12}$$

The above equation can be used to evaluate all the erasure values for $V \leq n_y - k_y$. However, this approach involves the evaluation of a new set of elementary symmetric functions for every erasure symbol to be determined. Another possibility is to solve for one unknown and then use the new found erasure value to modify the parity check sums, so that they reflect only the effect of $V - 1$ erasures. Assuming d_J is an erasure which has been determined. Then the new set of parity check sums can be expressed as

$$\mathcal{S}_i^{(2)} = \sum_{j \neq J}^{V} d_j y_j^i \qquad \text{for } i = 1, 2, ..., (n_y - k_y) \tag{A.13}$$

Notice that parity check sums $\mathcal{S}_i^{(1)}$ and $\mathcal{S}_i^{(2)}$ differ by the term $d_J y_J^i$, then we have

$$\mathcal{S}_i^{(1)} = \mathcal{S}_i^{(2)} + d_J y_J^i \tag{A.14}$$

which implies that

$$\mathcal{S}_i^{(2)} = \mathcal{S}_i^{(1)} + d_J y_J^i \qquad \text{for } i = 1, 2,..., (n_y - k_y)$$

Likewise, we can have

$$\mathcal{S}_i^{(v+1)} = \mathcal{S}_i^{(v)} + d_J y_J^i \qquad \text{for } i = 1, 2, \ldots, (n_y - k_y) \tag{A.15}$$

where $1 \leq v \leq V$. Hence, the $\mathcal{S}_i^{(v)}$ can be used to evaluate the next erasure, if there are V erasures in the received code sequence. This procedure can be repeated until all the erasure values are determined.

Summarizing, it follows that the erasure decoding for a RS code can be processed using the following procedures:

Step 1: Calculate the $(n_y - k_y)$ parity check sums.

$$\mathcal{S}_i^{(1)} = \overline{r'}_y(a_i) \quad \text{for } i = 1, 2, \ldots, (n_y - k_y).$$

Step 2: Calculate the elementary symmetric functions Z_{jh} $(h = 0, \ldots, V-1)$ excluding y_j using

$$\prod_{i \neq j}^{V} (y + y_i) = \sum_{h=0}^{V-1} Z_{jh} y^{V-h-1}$$

Step 3: Calculate the erasure value d_J by

$$d_J = \frac{\sum_{h=0}^{V-1} Z_{jh} \mathcal{S}_{V-h}^{(1)}}{\sum_{h=0}^{V-1} Z_{jh} y_j^{V-h}}.$$

Step 4: Calculate the new set of parity check sums using the obtained erasure value d_J.

$$\mathcal{S}_i^{(2)} = \mathcal{S}_i^{(1)} + d_J y_J^i$$

Step 5: Repeat step 2 until the V erasures are determined $(V < (n_y - k_y))$.

The final correct code values are $r_2(y_j) = \overline{r'}_y(y_j) + d_j$. Using the above method, the process of erasure correction decoding for a RS code can be reduced to only V $(V \leq n_y - k_y)$ trials.

Example: Now let us consider an example to clarify the erasure decoding procedure. A (15, 7) RS code is a nonbinary code over $GF(2^4)$, which has a minimum distance $d_{min2} = 9$

and which can correct a total of 8 or fewer erasure symbols. Each symbol in a (15,7) RS code is equivalent to a binary string of length 4 bits as shown in Table 5.2. Suppose that after x-axis decoding there are 4 symbols erased in the received code sequence. Let us assume the erasures occur in the locations a^{14}, a^{11}, a^{13}, and a^{12} with values a^4, a, 1, and a^7, respectively. The received erasures may be written

$$
\begin{aligned}
&V = 4 \\
&y_3 = a^{14} \qquad \overline{r'}_y(y_3) = a^7 \\
&y_2 = a^{11} \qquad \overline{r'}_y(y_2) = a^2 \\
&y_1 = a^{13} \qquad \overline{r'}_y(y_1) = a \\
&y_0 = a^{12} \qquad \overline{r'}_y(y_0) = a^8
\end{aligned}
$$

The following decoding procedure is then required. First, using equation (B.1), the calculated parity check sums $\mathcal{S}_i^{(1)}$ for $i = 1, 2, \ldots, 8$ are given by

$$
\begin{aligned}
&\mathcal{S}_1^{(1)} = a^{14} \quad \mathcal{S}_2^{(1)} = a^{13} \quad \mathcal{S}_3^{(1)} = a^5 \quad \mathcal{S}_4^{(1)} = a^6 \\
&\mathcal{S}_5^{(1)} = a^9 \quad \mathcal{S}_6^{(1)} = a^{13} \quad \mathcal{S}_7^{(1)} = a^{10} \quad \mathcal{S}_8^{(1)} = a^4
\end{aligned}
$$

Second, to solve for d_3, we evaluate the elementary symmetric functions for the three erasure location numbers excluding y_3. That is

$$\sum_{h=1}^{3} Z_{3h} y^{3-h} = \prod_{i \neq 3} (y + y_i)$$

which is expanded to

$$Z_{31} y^2 + Z_{32} y^1 + Z_{33} = (y_0 + y_1 + y_2) y^2 + (y_0 y_1 + y_0 y_2 + y_1 y_2) y + y_{c0} y_1 y_2.$$

Then we have

$$Z_{33} = y_0 y_1 y_2 = a^6$$

$$Z_{32} = y_0 y_1 + y_0 y_2 + y_1 y_2 = a^3$$

$$Z_{31} = y_0 + y_1 + y_2 = a^6$$

Thus from the equation (B.12), we can obtain

$$d_3 = \frac{\mathcal{S}_4^{(1)} + Z_{31}\mathcal{S}_3^{(1)} + Z_{32}\mathcal{S}_2^{(1)} + Z_{33}\mathcal{S}_1^{(1)}}{y_3^4 + Z_{31}y_3^3 + Z_{32}y_3^2 + Z_{33}y_3}$$

$$= \frac{a^6 + a^6 a^5 + a^3 a^{13} + a^6 a^{14}}{a^{11} + a^6 a^{12} + a^3 a^{13} + a^6 a^{14}}$$

$$= \frac{a^5}{a} = a^4$$

Having computed d_3, we utilize equation (B.15) to evaluate a new set of parity check sums. That is

$$\mathcal{S}_3^{(2)} = \mathcal{S}_3 + d_3 y_3^3 = a^5 + a^4 a^{12} = a^2,$$

$$\mathcal{S}_2^{(2)} = \mathcal{S}_2 + d_3 y_3^2 = a^{13} + a^4 a^{13} = a^{14},$$

$$\mathcal{S}_1^{(2)} = \mathcal{S}_1 + d_3 y_3 = a^{14} + a^4 a^{14} = 1.$$

With the new parity check sums, we repeat the same procedure for d_2. The new elementary symmetric function for the two erasure numbers excluding y_3 and y_2 are

$$Z_{22} = y_0 y_1 = a^{10},$$

$$Z_{32} = y_0 + y_1 = a.$$

Then using equation (B.12),

$$d_2 = \frac{\mathcal{S}_3^{(2)} + Z_{21}\mathcal{S}_2^{(2)} + Z_{22}\mathcal{S}_1^{(2)}}{y_2^3 + Z_{21}y_2^2 + Z_{22}y_2}$$

$$= \frac{a^2 + a a^{14} + a^{10}}{a^3 + a a^7 + a^{10} a^{11}}$$

$$= a.$$

Likewise, using equation (B.15) for evaluating the new set of parity check sums, we have

$$\mathcal{S}_2^{(3)} = a^6$$

$$\mathcal{S}_1^{(3)} = a^{11}$$

The new elementary symmetric function excluding y_1, y_2, and y_3 is simply

$$Z_{11} = y_0 = a^{12}.$$

Therefore, we can obtain

$$d_1 = \frac{\mathcal{S}_2^{(3)} + Z_{11}\mathcal{S}_1^{(3)}}{y_1^2 + Z_{11}y_1}$$

$$= \frac{a^6 + a^{12}a^{11}}{a^{11} + a^{12}a^{13}}$$

$$= \frac{a^{14}}{a^{14}} = 1$$

and finally we have

$$\mathcal{S}_1^{(4)} = \mathcal{S}_1^{(3)} + d_1 y_1 = a^{11} + a^{13} = a^4,$$

which gives that

$$d_0 = \frac{\mathcal{S}_1^{(4)}}{y_0} = \frac{a^4}{a^{12}} = a^7.$$

Hence the 4 erasures are solved, and the decoding process is finished. Therefore, the

decoding results can be given by

$$
\begin{aligned}
&V = 4 \\
&y_3 = a^{14} \qquad \overline{r}_y(y_3) = \overline{r'}_y(y_3) + d_3 = a^7 + a^4 = a^3 \\
&y_2 = a^{11} \qquad \overline{r}_y(y_2) = \overline{r'}_y(y_2) + d_2 = a^2 + a = a^5 \\
&y_1 = a^{13} \qquad \overline{r}_y(y_1) = \overline{r'}_y(y_1) + d_1 = a^7 + 1 = a^9 \\
&y_0 = a^{12} \qquad \overline{r}_y(y_0) = \overline{r'}_y(y_0) + d_0 = a^7 + a^7 = 0
\end{aligned}
$$

Appendix B

Publications

1. L. Zhang, M.J. Miller, K.W. Sarkies, " A Forward Error Protection Scheme for B-ISDN Using a Virtual Path Technique ", Australian Telecommunication Research (ATR), Vol. 23, No. 2, pp. 1-9, November 1989.
2. L. Zhang and K.W. Sarkies, "Modelling of a Virtual Path and Its Application for Forward Error Recovery Coding Schemes in ATM Networks", Proc. of SINCON'91, pp. 247-253, Singapore, September 1991.
3. L. Zhang and K.W. Sarkies, "Simulation Study of Cell Loss in ATM Networks", Proc. of ABSSS'92, Melbourne, July 1992.

Zhang, L., Miller, M.J., and Sarkies, K., (1989) A forward error protection scheme for B-ISDN using a virtual path technique.
Australian Telecommunication Research (ATR), v. 23 (2), pp. 1-9.

Zhang, L., and Sarkies, K., (1991) Modelling of a virtual path and its application for forward error recovery coding schemes in ATM networks.
Proceedings of the 1991 Singapore International Conference on Networks (SINCON'91), Singapore, September 1991, pp. 247-253.

Zhang, L., and Sarkies, K.W., (1992) Simulation study of cell loss in ATM networks. *Proceedings of the Australian Broadband Switching and Services Symposium '92 (ABSSS'92), Melbourne, July 1992.*

Bibliography

[1] CCITT Recommendation I.121, "On the Broadband Aspects of ISDN", CCITT Blue Book, Geneva, Switzerland, 1989

[2] "*Draft Revised Recommendation X.25, Interface Between Data Terminal Operating in the Packet Mode in Public Data Network*", Study Group VII, CCITT, Geneva, Feb. 1980; Reprint in *Computer Comm. Rev.*, Vol. 10, Nos. 1 and 2, pp. 56-129, Jan./April 1980.

[3] S. Sutherland, J. Burgin, R. Addie, " Information Transfer Protocols for the B-ISDN", Proc. 3rd FPS Research Workshop, Melbourne, May 1988.

[4] C. O'Neill, "Congestion Control in the Broadband ISDN," Proc. 4th Australian FPS Research Workshop, July 1989, Sydney.

[5] P. Kirton, J. Ellershaw, and J. Park, "Switching in the Future ISDN," Branch Paper 79, Switch and Signalling Branch, Telecom Australia Research Laboratory, December 1985.

[6] W. Verbiest, M. De Somer, and B. Voeton, "VBR Video Coding and ATM Switching: A Bell-RC Laboratory Experiment," International Workshop on Packet Video, Torino, 1988.

[7] M.J. Miller, "Error Control Techniques for Integrated Service Packet Networks," IEEE Journal on Selected Areas in Comm. Vol. 7, No. 5, pp. 690-697, June 1989.

[8] L. Zhang, M.J. Miller, K. Sarkies, " A Forward Error Protection Scheme for B-ISDN Using a Virtual Path Technique ", Australian Telecommunication Research (ATR), Vol. 23, No. 2, pp. 1-9, November 1989.

[9] M.J. Miller, L. Zhang, and D. Tait, " Error Detection Codes for Packet Switching Networks," 4th Australian FPS Research Workshop, Sydney, July 1989.

[10] R. Palmer, "A 32-Channel, 50 Mbit/s Experimental Fast Packet Switch," Proc. 4th Australian FPS Research Workshop, Sydney, July 1989.

[11] S.Q. Li, "A New Performance Measurement for Voice Transmission in Burst and Packet Switching," IEEE Trans. Comm., Vol. COM-35, pp. 1083-1094, Oct. 1987.

[12] B. Veiro, " Traffic Measurement on Variable Bit Rate (VBR) Sources with Application to Charging Principles," Proc. of ITC Special Seminar, Adelaide, September 1989.

[13] C. O'Neill, "A Method for Congestion Control in ATM Network," Proc. 5th Australian FPS Research Workshop, Melbourne, July 1990.

[14] M. Schwartz, "*Telecommunication Networks: Protocols, Modelling and Analysis,*", Addison Wesley Publishing Co., 1987.

[15] B. G. Kim," Characterization of Arrival Statistics of Multiplexed Voice Packets", IEEE Journal on Selected Area in Comm., Vol. SAC-1, No. 6, pp.1133-1139, Dec. 1983.

[16] J.N. Daigle, and J.D. Langford, "Models for Analysis of Packet Voice Communication Systems," IEEE J. on Selective Areas in Comm., Vol. SAC-4, No. 6, pp. 847-855, 1986.

[17] A. Kuczura, "The Interrupted Poisson Process as An Overflow Process", Bell Syst. Tech. J., Vol. 52, pp. 437-448, 1973.

[18] M.H. Rossiter, " A Switched Poisson Model for Data Traffic," Australian Telecommunication Research (ATR), Vol. 21, No.1, pp. 53-57, May 1987.

[19] H. Heffes, D. M. Lucantoni, "A Markov Modulated Characterization of Packetized Voice and Data Traffic and Related Statistical Multiplexer Performance", IEEE Journal on Selected Area in Comm., Vol. SAC-4, No. 6, pp. 856 - 867, Sept. 1986.

[20] B. Maglaris, D. Anastassiou, P. Sen, G. Kalsson, and J. Robbins, " Performance Models of Statistical Multiplexer in Packet Video Communications," IEEE Trans. Comm., Vol. 36, pp. 834-844, July 1988.

[21] P. Sen, B. Maglaris, N. Rikli, D. Anastassiou, "Models for Packet Switching of Variable-Bit-Rate Video Sources", IEEE Journal on Selected Areas in Comm., Vol. 7, No. 5, pp. 865-869, June 1989.

[22] S.E. Minzer, "Broadband ISDN and Asynchronous Transfer Mode (ATM)", IEEE Comm., Magazine, Vol.27, No.9, pp.17-24, Sep. 1989.

[23] L. Kleinrock, "*Queueing systems*", Vol. 1, J. Wiley&Son, New York, 1979.

[24] J. Gechter and P. O'Reilly, "Conceptual Issues for ATM", IEEE Network, pp. 14-16, Jan. 1989.

[25] T. Meisling, " Discrete-time Queueing Theory", Operational Research, Vol. 6, pp. 96-105, Jan./Feb. 1958.

[26] M.J. Karol and M.G. Hluchij, "Using a Packet Switch for Circuit-Switched Traffic: A Queueing System with Periodic Input Traffic," IEEE Trans. on Comm.., Vol. 37, No. 6, pp. 623 - 625, June 1989.

[27] B.D. Fritchman, "A Binary Channel Characterization Using Partitioned Markov Chain", IEEE Trans. on Information Theory, Vol. IT-13, No. 2, April 1967.

[28] S. Tsai, "Evaluation of Burst Error Correcting Codes Using a Simple Partitioned Markov Chain Model", IEEE Trans. on Comm., pp. 1031-1034, September, 1973.

[29] A.H. Haddad, S. Tsai, B. Goldberg, and G.C. Ranieri, "Markov Gap Models for Real Communication Channels," IEEE Trans. Comm., Vol. COM-23, No. 11, pp. 1189-1197, Nov. 1975.

[30] W. Feller, *An Introduction to Probability Theory and Its Applications,* Vol I, Wiley (New York), 1966.

[31] A. Seeto, " A Preview of the 1988 CCITT Standard on ATM", Proc. 3^{rd} Fast Packet Switching Research Workshop, pp. 1 - 5, May 1988, Melbourne.

[32] S. Lin and D.J. Costello Jr.,"*Error Control Coding: Fundamentals and Applications*", Prentice Hall, 1983.

[33] M.Y. Hiao, "A Class of Optimal Minimum Odd-Weight-Column SEC-DED Codes", IBM J. Res. Develop., Vol.14, pp. 402 - 408, July 1970.

[34] F. Polkinghorn, "Decoding of Double and Triple Error Correcting Bose-Chaudhuri-Hocquenghen Code", IEEE Trans. Inform. Theory. Vol. IT-12, pp.480-481, Oct. 1966.

[35] R. T. Chien, "Cyclic Decoding Procedures for Bose-Chaudhuri-Hocquenghen Codes", IEEE Trans. Inform. Theory, Vol. IT-10, pp. 357-363, Oct. 1964.

[36] H. Okano and H. Imai, "A Construction Method of High Speed Decoders Using ROM'S for Bose-Chaudhuri-Hocquenghem and Reed-Solomon Codes", IEEE Trans. on Computers, Vol. C-36, N0.10, pp. 1165 - 1171, Oct. 1988.

[37] W.W. Peterson and D.T. rown, "Cyclic Codes for Error Detection," Proc. IRE, Vol. 49, pp. 228-235, Jan. 1961.

[38] A.S. Tanenbaum, "*Computer Network*," Prentice Hall, 1981.

[39] J.K. Wolf, A.M. Michelson, and A.H. Levesque, "On The Probability of Undetected Error for Linear Block Codes", IEEE Trans. Comm., Vol. COM-30, pp. 317-324, Feb. 1982.

[40] P. Merkey and E.C. Posner, "Optimum Cyclic Redundancy Codes for Noisy Channel," IEEE Trans. Infom. Theory, Vol. IT-30, pp. 865-867, Nov. 1984.

[41] W.W. Peterson and E.J. Weldon, Jr., "Error-Correction Codes," Cambridge, MA:M.I.T. Press, 1972, 2nd ed.

[42] G. Castagnoli, J. Ganz, and P. Graber, "Optimum Cyclic Redundancy-Check Codes with 16-bit Redundancy," IEEE Trans. Comm., Vol. 38, pp.111-114, Jan. 1990.

[43] *Advanced Data Communication Control Procedures (ADCCP)*, American National Standards Institute, Washington, D.C., 1977.

[44] M.J. Miller and S. Lin, "The Analysis of some Selective-Repeat ARQ Schemes with Finite Receiver Buffer," IEEE Trans. Comm., COM-29, pp. 1307-1315, September 1981.

[45] S.Q. Li, "Study of Information Loss in Packet Voice Systems," IEEE Trans. Comm., Vol. 37, pp. 1192-1202, Nov. 1989.

[46] V. Verbiest, L. Pinnoo, and B. Voeten, "The Impact of the ATM Concept on Video Coding," IEEE J. on Selective Areas in Comm., Vol. 6, pp. 1623-1632, Dec. 1988.

[47] G. Karlsson and M. Vrtterli, "Packet Video and Its Integration into the Network Architecture," IEEE J. on Selective Areas in Comm., Vol. 7, pp. 739-751, June 1989.

[48] CMTT, "Digital Transmission of Component-coded Television Signals at 30-40 Mbits/s and 45 Mbits/s ", Doc. IWP CMTT/2-51, May 1987.

[49] A.J. Madry, P.R. Crosthwaite, and A.W. Seeto, "Error Recovery Procedures for Packet Video," Proc. of IREECON'90, pp. 1026-1029, September 1989.

[50] G.D. Forney, "Concatenated Codes," The M.I.T. Press, Cambridge, Massachusetts, 1967.

[51] G.D. Forney, "On Decoding BCH Codes," IEEE Trans. Inform. Theory, Vol. IT-11, pp. 549-557, Oct. 1965.

[52] T. Kasami, S. Lin, and W.W. Peterson, "Some Results on Weight Distributions of BCH Codes," IEEE Trans. Inform. Theory, Vol. IT-12, pp. 274, April 1966.

[53] I.S. Reed and G. Solomon, "Polynomial Codes Over Certain Finite Fields", J. SIAM, Vol. 8, pp. 300-304, June 1960.

[54] A.E. Eckberg, D. Luan, and D. Lucantoni, "Bandwidth Management: A Congestion Control Strategy for Broadband Packet Networks - Characterizing the Throughput - Burstness Filter", Proc. of ITC Spec. Sem. Adelaide 89, Adelaide, September 1989.

[55] C.J. Weinstein, "Fractional Speech Loss and Talker Activity Model for TASI and for Packet-Switched Speech", IEEE Trans. on Comm., Vol. 26, No. 8, August 1978.

[56] V.A. Bolotin, J.G. Kappel, and P.J. Kuehn, "Teletraffic Analysis of ATM Systems", IEEE J. Select Areas Comm., Vol.9, No.3, April 1991.

[57] J.Y. Hui, "Resource Allocation for Broadband Networks", IEEE J. Select. Areas Comm., Vol.6, No.9, pp 1598-1608, Dec. 1988.

[58] M. Butto, E. Cavallero, and A. Tonietti, "Effectiveness of the Leaky Bucket Policing Mechanism in ATM Networks", IEEE JSAC, Vol. 9, No. 3, pp.335, 1991.

[59] H. Kröner, G. Hebuterne, P. Boyer, and A. Gravey, "Priority management in ATM switching Nodes", IEEE JSAC, Vol. 9, No. 3, pp. 418, 1991.

[60] V. Ramaswami et. al., "Modelling Packet Arrivals from Asynchronous Input Lines", Proc. ITC 12, Torino, Italy, 1988.

[61] Y. Ohba, M. Murata, and H. Miyahara, "Analysis of Interdepature Process for Bursty Traffic in ATM Networks", IEEE JSAC, Vol.9, No.3, pp.468, 1991.

[62] M. Devault et. al., "The Prelude Experiment: Assessments and Future Prospects", IEEE J. Select. Areas Comm., pp.6-9, Dec. 1988.

[63] J. L. Boudec, "An Efficient Solution Method for Markov Models of ATM Links with Loss Priority", IEEE JSAC, Vol.9, No.3, pp. 408, 1991.

[64] J.W. Roberts and J.T. Virtamo, "The Superposition of Periodic Cell Arrival Streams in An ATM Multiplexer", IEEE Trans. on Comm., Vol.39, No.2, pp. 298, 1991.

[65] L. Zhang, K. Sarkies, M.J. Miller," Modelling of Virtual Path for B-ISDN ", Proc. of 5th Australian FPS Research Workshop, Melbourne, July 1990.

[66] CCITT Draft Recommendations I.361-361, "ATM Layer Specification for B-ISDN" and "ATM Adaptation Layer Functional Specification", Study Group XVIII, Geneva, Switzerland, Feb. 1990.

[67] G.M. Woodruff, R.G.H. Rogers, and P.S. Richards, "A Congestion Control Framework for High-Speed Integrated Packetized Transport", Proc. IEEE GLOBECOM'88, pp. 7.1.1-7.1.1, Florida, Nov. 1988.

[68] G. Gallassi, G.Rigolio, and L. Fratta, "ATM: Bandwidth Assessment and Bandwidth Enforcement Policies", Proc. IEEE GLOBECOM'89, pp.49.6.1-49.6.6, Dallas, TX, 1989.

[69] H. Brunnel, "Queueing Behavior of Statistics Multiplexers with Correlated Inputs", IEEE Trans. on Comm., Vol.36, pp.1339-1341, Dec. 1988.

[70] J.J. Gordon, "Modelling Bursty Traffic with Two-State Sources", Proc. 3rd Australian Teletraffic Research Seminar, Paper 4.2, Nov. 1988.

[71] D.A. Hughes, G. Anido, and H.S. Bradlow, "Queueing Analysis of Multiplexed Bursty Traffic with Application to ATM Switch Performance", Proc. 5th Australian Teletraffic Research Seminar, Melbourne, Dec. 1990.

[72] I. Norros, J.W. Roberts, A. Simonian, and J.T. Virtamo, "The Superposition of Variable Bit Rate Sources in an ATM Multiplexer", IEEE J. Selec. Areas on Comm., Vol.9 No.3, pp.378-386, April 1991.

[73] K. Srinam and W. Whitt, "Characterizing Superposition Arrival Processes in Packet Multiplexers for Voice and Data", IEEE J. Selec. Areas on Comm., Vol.4, No.6, pp.833-846, Sep. 1986.

[74] J.N. Daigle and J.D. Langford, "Models for Analysis of Packet Voice Communication Systems", IEEE J. Selec. Areas on Comm., Vol.4, No.6, pp.847-855, Sep. 1986.

[75] A.E. Eckberg, "The Single Server Queue with Periodic Arrival Process and Deterministic Service Time", IEEE Trans. on Comm., Vol.27, pp.556-562, March 1979.

[76] J.R. Louvion *et al*, "A Discrete-Time Single Server Queue with Bernoulli Arrivals and Constant Service Time", Proc. ITC 12, Torino, Italy, 1988.

[77] J.S. Turner, "New Directions in Communications (or Which Way to the Information Age?), IEEE Comm. Mag., Vol.25, No.10, pp.8-15, October 1986.

[78] D. Hong and T. Suda, "Congestion Control and Prevention in ATM Networks", IEEE Network Mag., Vol. 5, No. 4, pp.10-17, July 1991.

[79] A.E. Eckberg, B.T. Doshi, and R. Zoccolillo, "Controlling Congestion in B-ISDN/ATM: Issues and Strategies", IEEE Comm. Mag., Vol. 29, No. 9, pp.64-74, September 1991.

[80] C.A. Cooper and K.I. Park, "Toward a Broadband Congestion Control Strategy", IEEE Network Mag., Vo4., No. 3, pp.18-23, May 1990.

[81] O. Hashida, Y. Takahashi, and S. Shimogawa, "Switched Batch Bernoulli Process (SBBP) and The Discrete-Time SBBP/G/1 Queue with Application to Statistical Multiplexer Performance", IEEE Selec. Areas on Comm., Vol. 9, No.3, pp 394-401, April 1991.

[82] H. Kröner, T. Theimer, and U. Briem, "Queueing Models for ATM Systems - A Comparion", Proc. of 7th ITC Specialists Seminar, Paper 9.1, New Jersey, Oct. 1990

[83] K. Sarkies, "Features of Traffic in ATM Networks", TRC Report No. 11, Teletraffic Research Centre, Adelaide University, Dec. 1991.

[84] H. Ohta and T. Kitami, "A Cell Loss Recovery Method Using FEC in ATM Networks", IEEE Selec. Areas on Comm., Vol. 9, No.9, pp 1471-1483, December 1991.

[85] T. Hou and A. Wong, "Queueing Analysis for ATM Switching of Mixed Continuous-bit-rate and Bursty Traffic", Proc. INFOCOM'90, pp. 660-667, San Francisco, June 1990.

[86] B.T. Doshi and H. Heffes, "Overload Performance of Several Processor Queueing Disciplines for the M/M/1 queue", IEEE Trans. on Comm., Vol. COM-34, No. 6, pp.538-546, June 1986.

[87] M.J. Miller and S.V. Ahamed, "*Digital Transmission Systems and Networks*", Vol. I, Computer Science Press, Inc. U.S.A., 1987.

[88] CCITT, "VIII Plenary Assembly Red Book,", International Telecom, Geneva, 1984.