Title	Priority Traffic and ARQ Error Controls in Communication Systems
Author(s)	Chen, Chun-Xiang
Citation	大阪大学, 1994, 博士論文
Version Type	VoR
URL	https://doi.org/10.11501/3094169
rights	
Note	

Priority Traffic and ARQ Error Controls in Communication Systems

Department of Applied Physics
Osaka University
大阪大学

Jan. 1994

Chun-Xiang CHEN
陳　春 祥

1

Priority Traffic and ARQ Error Controls in Communication Systems

(通信システムにおける優先権トラヒック
制御及び ARQ 誤り制御に関する研究)

A Dissertation for the Degree of Doctor of Engeneering

Department of Applied Physics
Faculty of Engeneering
Osaka University
大阪大学

Jan. 1994

Chun-Xiang CHEN
陳　春　祥

Acknowledgments

This work was accomplished under the supervision of Prof. Kozo Kinoshita, with the Department of Applied Physics, Faculty of Engineering, Osaka University. I am greatly indebted to him, for giving me the support and guidance which greatly enhance my study while at Osaka University. I would like to express my deepest gratitude and appreciation to him for sincere interest in my life in Osaka.

I would like to express my special acknowledgments to associate Prof. Masaharu Komatsu, who taught me a lot about the fundamental theories in the fields of communication systems and networks, and also guided me through the whole course of this work. I am very thankful to him for much appreciated assistance, constructive discussion, and patient guidance, comments and encouragements.

I would like to express my gratitude to Prof. Yoshiki Ichioka, Prof. Hiroaki Ishii and Prof. Junichi Toyoda for their valuable comments and critical reading of the dissertation. Their concern for my dissertation and guidance is deeply acknowledged. Also, suggestions and encouragements from Prof. Hiroshi Masuhara, Prof. Ryuichi Shimizu, Prof. Satoshi Kawata, Prof. Shinichi Nakashima, Prof. Ayao Okiji, Prof. Seiichi Goto, and Prof. Hiroshi Iwasaki are greatly acknowledged.

I also wish to thank Dr. Noriyoshi Itazaki, who taught me a lot about the knowledge of workstation. And, I am very thankful to Dr. Seiji Kajihara for his valuable discussions and encouragements.

Many thanks go to all members of Kinoshita Lab. whose unusual hospitality and cooperation made me enjoy in Japan. They help me to understand Japanese culture and customs.

I would like to express my sincere thanks to Japanese Ministry of Education, Science and Culture, for the support in this study through Monbusho scholarship.

Finally, I would like to express my special thanks to my parents for their hearty support and encouragements which have also been of major importance in finishing this work.

Chun-Xiang CHEN

Osaka, Japan

Jan. 1994

TABLE OF CONTENTS

PREFACE

Two major problems, traffic (including flow, congestion and priority controls, etc.) and error controls, must overcome for implementing information transmission. Various traffic and error controls have been studied for different communication environments because different communication environment requires different control scheme.

Recently, the infrastructures and the environments of the communication systems have been changing rapidly. The transmission speed becomes very high and the capacity of the transmission line becomes very large. These changes make the communication systems enable to support multimedia traffic requiring bandwidth ranging from a few kilobits per second (e.g., a slow terminal) to several megabits per second (e.g., moving image source). In high-speed communication network (such as ATM: Asynchronous Transfer Mode), the bandwidth and quality-of-service (QOS) required by a source are highly varied, e. g., the data transmission (file transmission, etc.) require the error-free transmissions but the transmission delay is not so critical; the real-time voice sources require rapid transfer through a communication network, but the loss of small amounts of voice information is tolerable. These characteristics make it necessary to reexamine the traffic control schemes under the high-speed communication environments.

In error control schemes, there are three well-known schemes named FEC (Forward Error Correction), ARQ (Automatic Repeat reQuest) and Hybrid-ARQ. FEC employs error correction, while ARQ employs pure error detection only and retransmits the erroneous data. Hybrid-ARQ was proposed to have the advantages of both FEC and ARQ. It employs one code for error correction and another code for error-detection (i.e., combining FEC and ARQ). ARQ is commonly used due to its high system reliability and its simplicity. Many ARQ's and their modified schemes have been studied. However the existing ARQ schemes are almost designed for a single-channel system in which the transmitter only sends out one packet at a time. For the parallel-channel-system in which the transmitter sends out several packets over parallel channels at a time, the half-duplex line system and the high-speed transmission system, the error controls are also necessary.

In this dissertation, the multiclass traffic control in a high-speed communication system by using the priority queueing strategy will be studied, where the priority sources include voice and video sources which have the burstiness of the cell arrivals, the non-priority source includes data sources. Due to the properties of the priority and non-priority sources (e.g., the priority source is sensitive to the transmission delay, the non-priority source is sensitive to the transmission errors), a priority control for multiclass traffic is considered and the system performances such as transmission delay for the priority sources and the cell discard probability due to the buffer overflow for the non-priority sources are analyzed; and then, main attention is concentrated to the error control schemes. Among the many error control schemes, here only the ARQ and its modified schemes are considered in this thesis to adapt various communication environments such as half-duplex transmission line, multiple-channel and high-speed communications. Needless to say, it is insufficient only to provide error-free communication without considering the system performances. So the purpose of this work is to analyze the system performances such as the throughput, the packet delay or the buffer occupancy under various communication environments.

1 General Overview

1.1 On Communication Systems

Primitive forms of communication systems have a long history, including the smoke and optical signals [1, 2, 5, 6], and certainly including nineteenth-century telegraphy. A major development, in early 1960s, was the use of communication links to connect central computers to remote terminals and other peripheral devices, such as printers and remote job entry points. Since 1970 there has been an explosive growth in the number of wide area (WAN such as the seminal ARPANET and TYMNET) and local area networks (LAN such as Ethernets and token ring networks). With the multiplicity of different data networks in existence in the 1980s, more and more networks have been connected via gateways and bridges so as to allow users of one network to communicate with users of other networks [10, 12, 13, 14, 20].

With the gradual maturing of optical fiber technology, transmission costs, particularly for high data rate links, are dropping at an accelerating rate which is expected to continue well into the future. The capacity of a single optical fiber using today's technology is 10^9 to 10^{11} bits/sec, and in the future this could rise to 10^{14} or more [2, 4, 5]. Optical fiber is becoming widespread in use and is expected to be the dominant mode of transmission in the future. There are major economies of scale associated with higher link data rate; e.g., the cost of a 1.5 Mbps link is about six times that of a 64 Kbps link, but the data rate is 24 times higher. Furthermore, the high data rate link makes it possible to indiscriminately deal with various information such as video, voice and data files, by an integrated network [1, 5, 8, 11]. Meanwhile, this will greatly change the nature of network applications. On the other hand, the communication requirements for accessing files and data bases have been increasing rapidly in recent years. Part of the reason for this is just the natural growth of an expanding field and the appearances of new media and service; for example, the facsimile service which has become very popular in recent years, the electronic mail service which is more economical than facsimile. Also it is not difficult to imagine that not so faraway, the moving video transmission and the image telephone service will be provided.

1.2 On Traffic Controls

Traffic control includes many actual aspects, such as congestion control, flow control. For congestion and flow controls, there has a well-known reactive control, which instructs the source nodes to throttle the traffic flow by giving feedback to them if congestion was detected [89]. Congestion or flow control for ATM can be performed in two ways: admission control and bandwidth enforcement. Admission control determines whether to accept or reject a new connection at the time of call setup. This decision is based on traffic characteristics of the new connection and the current network load. The bandwidth enforcement monitors individual connections to ensure that the actual traffic flow conforms with that reported at call establishment [90]. To implement these control schemes, practically, priority control discipline is very useful, e.g., according to the quality-of-service, the different priority can be assigned to different traffic source, when the congestion will occur, or the traffic flow is not smooth, traffic control can determine whether accept or reject a call, and which traffic source is serviced preferentially. Fig. 1.1 shows the source utilization (fraction of time for which the source transmits) versus the bit rate during such periods for typical broadband applications. In this thesis, we will consider a priority discipline for multiclass traffic.

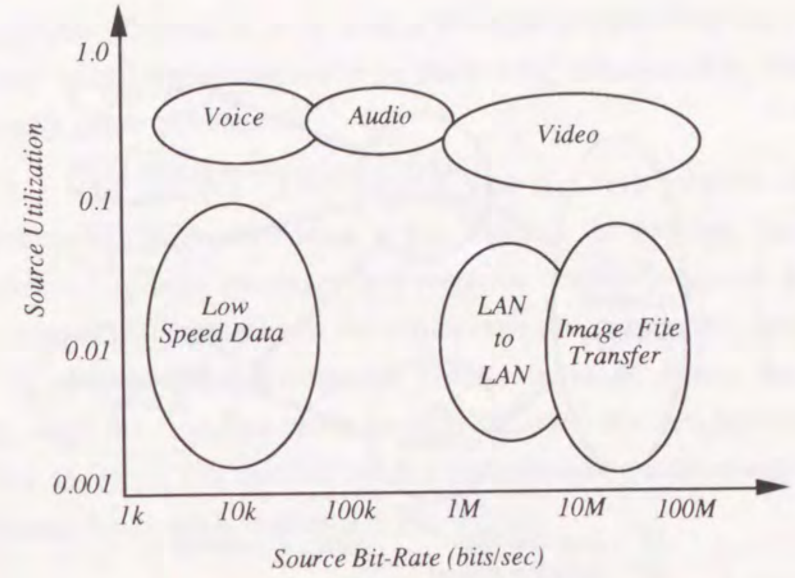


Fig. 1.1 Traffic Sources for Broadband Networks

The different traffic source has the different transmission requirements. Real-time voice, for instance, requires rapid transfer through a network, but a loss of small amounts of voice information is tolerable, and in many data sources, real-time service is not of primary importance, but high throughput and error-free transmission are required. Some sources, such as real-time stationary video, require error-free transmission as well as rapid transfer. Therefore, a powerful integrated network should be flexible enough to adapt various service requirements.

1.3 On Error Controls

In order to maintain reliable and efficient communications over noisy channel, there are three basic error control schemes: ARQ (Automatic Repeat reQuest), hybrid-ARQ and FEC (Forward Error Correction). ARQ employs pure error detection only and retransmits the erroneous data; FEC employs error correction, and hybrid-ARQ employs one code for error correction (FEC) and another code for error-detection (ARQ) (Fig. 1.2).

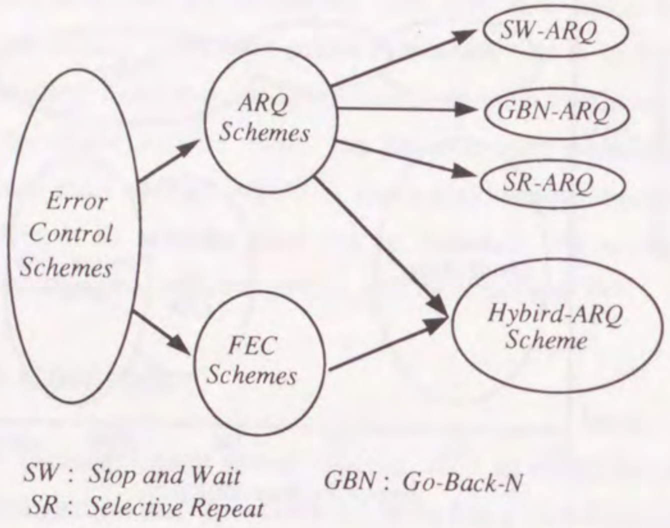


Fig. 1.2 Classification of Error Control Schemes

FEC scheme can provide a constant throughput efficiency, set by the code rate, regardless of the channel conditions. As the decoded word, however, must be delivered to the user regardless of whether it is correct or not, the system reliability falls as the channel degrades. On the contrary, an ARQ scheme, incorporating powerful error-detecting codes, can provide high system reliability, reasonably independent of the channel quality, but the throughput depends on the number of requested retransmissions, i.e., on channel quality, and therefore decreases as the channel error rate increases. Between FEC and ARQ, hybrid-ARQ is proposed to have the advantages of both FEC and ARQ. It employs one code for error-correction (FEC) and another code for error-detection (ARQ). The receiver first attempts to correct any error, and then the decoded data packet is tested for error detection. The retransmissions are requested if necessary. The hybrid-ARQ is able to achieve high throughput and reliability but at the expense of complex error correction and detection algorithms. The ARQ scheme is attractive and will be considered later due to its high reliability and simple control algorithm.

Finally, a classification of error control schemes is depicted in Fig. 1.2. As a preliminary work, the operations of the basic ARQ schemes (SW, GBN and SR) are briefly given as follows.

SW-ARQ: Stop-and-Wait ARQ scheme. This is a very primitive form of ARQ schemes. The transmitter sends a data block to the receiver, then waits for the acknowledgment message. If a negative acknowledgment message (NAK for simplicity) comes back, the transmitter retransmits this data block. While if a positive acknowledgment (ACK) message comes back, the transmitter sends the next data block. In receiving side, if a data block arrived, it is checked for error. The receiver sends a corresponding acknowledgment to the transmitter. An example is shown in Fig. 1.3 (a).

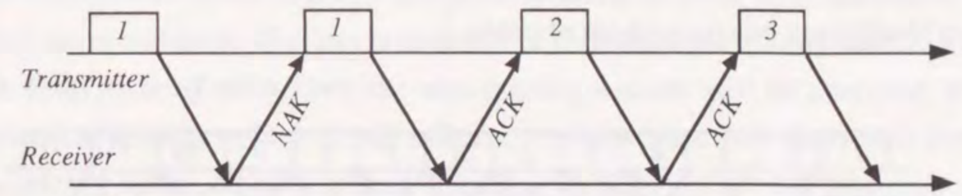


Fig. 1.3 (a): Operation of SW-ARQ Scheme

GBN-ARQ: Go-Back-N ARQ scheme. The transmitter consecutively transmits data blocks without waiting for the ACK or NAK. While a NAK comes back, the transmitter immediately retransmits the NAK'ed block and all the data blocks which have been transmitted after the NAK'ed block (Fig. 1.3 (b)).

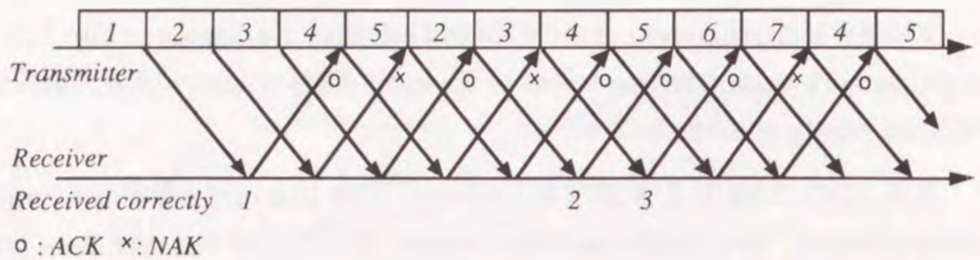


Fig. 1.3 (b) Operation of GBN-ARQ scheme

SR-ARQ: Selective Repeat ARQ scheme. The transmitter consecutively transmits data blocks without waiting for the ACK or NAK. While a NAK comes back, only the NAK'ed block is retransmitted immediately (Fig. 1.3 (c)). In addition, the resequencing buffer in the receiver is necessary because the data blocks may be received out of order.

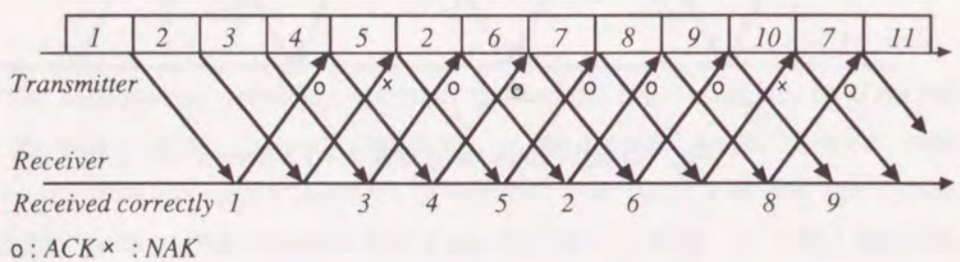


Fig. 1.3 (c) Operation of SR-ARQ Scheme

1.4 Object and Organization of the Thesis

Various schemes about traffic and error controls have been studied because different communication environment requires different control scheme [1-5, 10-40]. However, how to balance the cell delay and cell discard probability for multiclass traffic is also a problem which have not been settled (in ATM networks, fixed-length packet is called cell), and efficient error control schemes for high-speed communication, multiple-parallel-channel and half-duplex transmission line systems have not been studied. In this thesis, a

priority control discipline for multiclass traffic in an integrated transmission system is considered, and some novel ARQ error control schemes are proposed to provide reliable transmission for (1) high-speed, (2) half-duplex transmission line, and (3) multiple-parallel-channel communication systems. Generally, we use the system throughput, the packet delay time and/or the queueing length in the transmission buffer as main measurements to evaluate the system performances. By analytical and numerical results, the performances of the proposed schemes are clarified. The rest of this thesis is organized as follows.

In chapter 2, a traffic control scheme in high-speed transmission system is presented, under the considerations that the priority sources (such as voice and video) are sensitive to the transmission delay and the non-priority sources to the transmission errors and loss. The priority source exhibits some burstiness in the form of talkspurt. The loss probability of cells from the priority sources and the delay time of cells from the non-priority sources will be analyzed. By numerical results, we clarify the influences of the burstiness of arrivals from the priority source on both the cell discard probability and the cell delay.

Chapter 3 describes a dialogue communication system which consists of two stations over a half-duplex transmission line and studies its performances. When a station seizes the right to send its message (a message is divided into some fixed-length packets), it consecutively transmits k packets at most. We analyze the transmission time of a message and the throughput performances of SW, GBN and SR protocols for the half-duplex transmission line system. Based on the analytical and numerical results, we clarify the influences of the switching and thinking times, which inherently exist in the system, on the throughput performance, and give the optimal k which makes the throughput to become maximum. Also it is observed that the throughput performances are greatly influenced not only by the switching and thinking times but also by the average message length.

Chapter 4 proposes a block SR-ARQ scheme for the high-speed communication system in which the protocol processing overhead time becomes remarkable in determining the system performances. The block SR-ARQ uses a single acknowledgment to acknowledge a block of packets in order to reduce the overhead time. The maximal number of the packets acknowledged by an acknowledgment packet is defined as block size. The throughput and the packet delay are analyzed and validated by simulation. The numerical results show that there exists an optimal block size which obtains both the maximum throughput and the minimum average packet delay.

In chapter 5, a communication system in which a transmitter is connected to a receiver through multiple-parallel channels , is considered, and the GBN-ARQ scheme is used to handle transmission error. The different channels are assumed to have different packet error probabilities, and the errors occur independently. A packet error on one channel results in retransmission of packets assigned to other channels under GBN-ARQ error control scheme. Therefore, the channel-grouping (a grouped-channel is used to transmit a packet and its copies at a time), would affect the system throughput performance. The throughput performance is analyzed, and the channel-grouping methods are discussed. And finally, a tree-algorithm used to efficiently search the optimal channel-grouping which would make the throughput to become maximal, is proposed.

Chapter 6 considers a two-channel system which is a special case of parallel-channel-system described in chapter 5 under GBN-ARQ scheme. A dynamic packet assignment rule is developed to efficiently use the idle channel. It is shown that the dynamic assignment rule is much powerful than the static one by analyzing the average queueing length.

Chapter 7 gives our concluding remarks and some further problems.

2 Multiclass Traffic Control with Priority Discipline in High-Speed Transmission System

2.1 Introduction

As mentioned earlier, B-ISDN implemented by ATM networks is proposed to support various traffic, such as video, voice and data, due to the different transmission requirements (e.g., the real-time transmission for voice, the error-free transmission for data, and not only the real-time but the error-free transmission for stationary video). Here such various traffic is considered as *multiclass* traffic. In ATM networks, traffic control for multiclass is one important subject.

In order to use a certain network to transmit multiclass traffic which are from different type users, priority discipline is a powerful method for traffic control for satisfying various quality-of-service (QOS) (such as tolerable delay and discard probability) and avoiding congestion [3-5, 9, 11, 14, 15, 84, 85]. In [16], dynamic call admission control has been presented using the distribution of the number of cells arriving during the fixed interval. In [83], two dynamic priority schemes, Minimum Laxity Threshold (MLT) and Queue Length Threshold (QLT), try to reduce the performance degradation for the low priority traffic. In these dynamic priority schemes, priority level changes with time.

To evaluate the performances of ATM networks, modeling for traffic source and analysis for the system are necessary. In this chapter, some traffic sources are given first, then a basic queueing model with two kinds of sources, in which one is assigned high priority and another is assigned low priority (non-priority), is considered. We analyze the discard probability of the priority source and the delay of the non-priority source.

2.2 Traffic Model

In ATM networks, the cell length is fixed, time axis is divided into cell-sized slot, i.e., the slot time is equal to the cell service time, so that just one cell can be transmitted in one slot.

2.2.1 Modeling for time-sensitive source

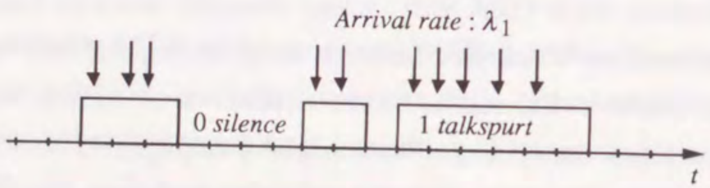


Fig. 2.1 IIP model

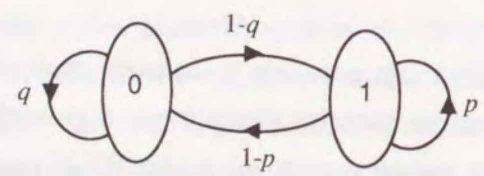


Fig. 2.2 Transition diagram of IIP model

An arrival process of cells from a time-sensitive source (voice or/and video) is very complex due to the strong correlation among arrivals. The correlated generation of cells from a single source can be modeled by an Interrupted Poisson Process (IPP) [12, 13, 20]. In an IPP model (Fig. 2.1), each source is characterized by *talkspurt* (state 1) and *silence* (state 0) periods (Fig. 2.2), which appear by turn. The state transition from talkspurt (silence) to silence (talkspurt) generates a Markov process. The state transition diagram is depicted in Fig. 2.2. Let P_{10} (P_{01}) be the state transition probability from state 1 (state 0) to state 0 (state 1) at the end of slot. They are given as

$$\begin{cases} P_{10} = 1 - p \\ P_{01} = 1 - q. \end{cases} \tag{2.1}$$

The length distribution of talkspurt or silence is assumed to be geometrical which has the average length $A=1/(1\text{-}p)$ or $B=1/(1\text{-}q)$. The state transition from talkspurt (silence) to silence (talkspurt) only occurs at the end of the slot. Here we introduce a ratio of talkspurt (denoted by ξ) to be the ratio of average length of talkspurt to average length of silence. It is given by

$$\xi = \frac{A}{A+B}. \tag{2.2}$$

Now we consider the arrival process of cells from a single source. The cell arrivals just occur in state 1, and in state 0, there are no arrivals. For convenience, the talkspurt duration is also called bursty duration because the arrivals aggregate in talkspurt duration, and the source is also called bursty source. Let $P_r(1, n, \lambda_1)$ be the probability that n cells from 1 source arrive in a slot during state 1, where λ_1 is the arrival rate. It is given by

$$P_r(1,n,\lambda_1) = \frac{e^{-\lambda_1}(\lambda_1)^n}{n!}. \tag{2.3}$$

Here we introduced another parameter ρ_1 which is defined as the traffic density, given by

$$\rho_1 = \lambda_1 \xi. \tag{2.4}$$

So, if the traffic density is constant, λ_1 becomes bigger while ratio ξ becomes smaller, i.e., the bursty source has strong burstiness.

2.2.2 Modeling for loss-sensitive source

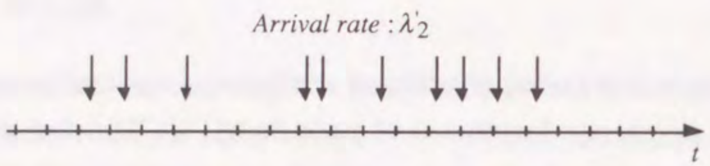


Fig. 2.3 Arrival process of loss-sensitive source

For a loss-sensitive source such as data source, it is well-known that generation of data from a single source is well characterized by a Poisson process or by a geometric interarrival process. Here under consideration, we simply assume that the number of arrivals from a single source (called non-priority source later) in a slot is distributed to a Poisson process with mean λ'_2. Let $P'_r(1, n, \lambda_2)$ denote the probability that n cells from a loss-sensitive source arrive in a slot (Fig. 2.3). It is given by

$$P'_r(1,n,\lambda'_2) = \frac{e^{-\lambda'_2}(\lambda'_2)^n}{n!}. \tag{2.5}$$

Since the loss-sensitive source requires the high reliable transmission, if a cell is lost due to the buffer overflow (i.e., if a cell is discarded), the retransmission is requested. A simple model is described in Fig. 2.4, where λ'_2 is the original arrival rate of the source, η is the retransmission rate due to the buffer overflow. Therefore, the practical arrival rate is identical to

$$\lambda_2 = \frac{\lambda'_2}{1-\eta}. \tag{2.6}$$

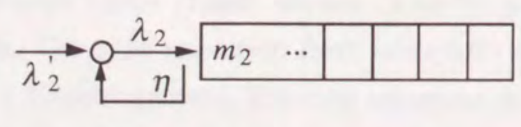


Fig. 2.4 Operation of loss-sensitive source

2.3 Analytical System Model

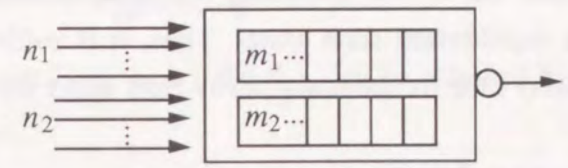


Fig. 2.5 Analytical system model

Consider a system shown by Fig. 2.5 in which n_1 sources act as active priority sources which are considered as time-sensitive sources and n_2 sources as non-priority sources which are considered as loss-sensitive sources. The priority source exhibits some burstiness in the form of talkspurt modeled by IPP described in 2.2.1. Non-priority source is modeled in 2.2.2. In addition, assume that arrivals among different sources are mutually independent, and if priority cell exists in the buffer they are served superiorly to non-priority cells, but the cells among homogeneous sources are served according to FCFS (First-Come First-Service).

For time-sensitive priority cells, the delay time must be kept under a threshold value which is determined by the characteristics of source and the transmission speed of the system. So that, the cell of which delay time exceeds the threshold value, must be discarded. If we prepare enough buffer to store the priority cells, the discard probability can be controlled less, but the queueing delay must become larger. On the other hand, if the buffer is too small, the discard probability of priority cells should become greater. So, it is a problem how to balance the delay time and the discard probability. Here we assume that the priority buffer has finite capacity m_1. The buffer used to store the priority cells, is called priority buffer.

For the loss-sensitive non-priority cells (e.g., data), the discard probability is severe, however a small amounts of delay is tolerable. It is reasonable to assume the capacity of the buffer used to store the non-priority cells is infinite, but it is impossible in real system. Therefore, let the buffer capacity of the non-priority cells be a certain value m_2. Furthermore, if an arriving non-priority cell is discarded due to the buffer overflow, it must be immediately retransmitted (Fig. 2.4). Assume the arrival including the retransmitted cells is Poisson process, and the equilibrium state exists. Thus, it is sufficient that we only investigate the delay time for the non-priority cells under the equilibrium state.

2.4 Performance Analysis

2.4.1 Analysis for priority cells

In the analytical model (Fig. 2.5), one cell can be transmitted in a slot if and only if the cells exist in the buffer. The service for non-priority cells does not influence the transmission of priority cells due to their priority property. Thus, only considering the priority cell for the analysis about the priority cells is sufficient.

Define Φ_{ij} to be the probability that j sources will be in state 1 in the next slot when i sources are in state 1 in the current slot. Thus

$$\Phi_{ij} = \sum_{u=(i-j)^+}^{\min(i,n_1-j)} \binom{i}{u} P_{10}^u (1-P_{10})^{i-u} \binom{n_1-i}{j-i+u} \cdot P_{01}^{j-i+u}(1-P_{01})^{n_1-j-u}, \tag{2.7}$$

where $(k)^+=\max\{k, 0\}$.

Let (q_n, t_n) denote the state of the system at the beginning of the n-th slot, where t_n is the number of the priority sources in state 1, and q_n is the queueing length of the priority cells. The queueing length is defined as the number of the priority cells in priority buffer at the end of a slot. Thus the system state transition probability $P_{ij,kl}$ can be defined as

$$P_{ij,kl} = P\{q_{n+1} = l, t_{n+1} = k | q_n = j, t_n = i\},$$
$$0 \le i, k \le n_1, \qquad 0 \le j, l \le m_1.$$

Define $P_r(m, n, \lambda_1)$ to be the probability that n cells arrive in a slot at the condition of m priority sources are in talkspurt. Under the assumption that the arrival process is Poisson process in talkspurt, so we have

$$P_r(m,n,\lambda_1) = \frac{e^{-m\lambda_1}(m\lambda_1)^n}{n!}. \tag{2.8}$$

Since only one cell can be sent in a slot, thus

$$P_{ij,kl} = 0, \quad if \quad l < j-1.$$

Using the Φ_{ij} and $P_r(m, n, \lambda_1)$, the $P_{ij,kl}$ can be obtained by considering the following four cases.

A: $j = 0$ and $0 \le l \le m_1$-1:

In this case, there are no priority cells in the buffer. Also, a non-priority cell is served if non-priority cells exist. Thus

$$P_{i0,kl} = \Phi_{ik} P_r(i,l,\lambda_1).$$

B: $1 \le j \le m_1$ and j-1 $\le l < m_1$:

On this occasion, one cell can be served and it must have l-j+1 arrivals of priority cells. So we have

$$P_{ij,kl} = \Phi_{ik} P_r(i,l-j+1,\lambda_1).$$

C: $j = 0$ and $l = m_1$:

As in case A, there are no service for the priority cells. A great deal of priority cells may arrive. So it is very likely that some of them will be discarded when no idle space is found in the buffer. Thus

$$P_{i0,kl} = \sum_{v=m_1}^{\infty} \Phi_{ik} P_r(i,v,\lambda_1) = \Phi_{ik}\left(1 - \sum_{v=0}^{m_1-1} P_r(i,v,\lambda_1)\right).$$

D: $j > 0$ and $l = m_1$:

In this case, one priority cell can be served and overflow may cause if the number of arrivals is greater than m_1-j+1. Thus

$$P_{ij,km_1} = \sum_{v=m_1-j+1}^{\infty} \Phi_{ik} P_r(i,v,\lambda_1) = \Phi_{ik}\left(1 - \sum_{v=0}^{m_1-j} P_r(i,v,\lambda_1)\right).$$

In the equilibrium state, the state probability of the system for the priority cells is defined as π_{ij}, where i is the number of the priority sources in talkspurt and j is the queueing length of the priority cells in the buffer. Let Π be a row vector of the state probabilities, given by

$$\Pi = [\pi_{00}, \pi_{01}, \cdots, \pi_{n_1 0}, \cdots, \pi_{n_1 m_1}], \tag{2.9}$$

and $\mathbf{P}$ denote a square matrix of transition probabilities. Because the service process satisfies the Markov process, the equilibrium probabilities can be obtained from the Markov linear equation and the boundary condition given as follows.

$$\begin{cases} \Pi = \Pi\mathbf{P} \\ \sum_{i=0}^{n_1}\sum_{j=0}^{m_1} \pi_{ij} = \mathbf{1}. \end{cases} \tag{2.10}$$

Therefore, using the equilibrium probabilities π_{ij}, the probability $P_{rob1}(i)$ that the queueing length in the buffer is i, is given by

$$P_{rob1}(i) = \sum_{j=0}^{n_1} \pi_{ji}. \tag{2.11}$$

After obtaining these probabilities, we can easily evaluate the discard probability P_{dis}.

$$P_{dis} = \frac{E_d}{E_a}, \tag{2.12}$$

where, E_d is the mean number of discarded cells and E_a is the mean number of arrivals of the priority cells in a slot. They are given as follows.

$$E_a = \sum_{j=0}^{m_1}\sum_{m=1}^{n_1} (\lambda_1 m)\pi_{mj} = \lambda_1 n_1 \frac{A}{A+B}, \tag{2.13}$$

$$E_d = \sum_{m=1}^{n_1}\sum_{j=0}^{m_1}\sum_{n=1}^{\infty} nP_r(m, m_1 - j + n, \lambda_1)\pi_{mj}. \tag{2.14}$$

2.4.2 Analysis for non-priority cells

2.4.2.1 Equilibrium state probabilities

Based on the analytical model, non-priority cells in the buffer have to wait for the service if the priority buffer is not empty.

Let b_1 be the number of priority cells in the buffer in equilibrium state and f_{ij} be the state transition probabilities of non-priority cells in the buffer where i is the queueing length of non-priority cells in the current slot and j in the next slot. No non-priority cell is served if the priority cells exist in the priority buffer. However, a non-priority cell is served in the case that no priority cell exists. So we have

$$f_{ij} = \sum_{k=1}^{m_1} P_{rob}(b_1 = k)a_{2,ij} + P_{rob}(b_1 = 0)a_{2,ij}, \quad 0 \le i, j \le m_2, \tag{2.15}$$

where

(*a*) when $b_1 \ge 1$:

$$\begin{aligned} a_{2,ij} &= 0, && \text{if } j \le i-1, \\ a_{2,ij} &= P_r(n_2, j-i, \lambda_2), && \text{if } j \ge i. \end{aligned}$$

(*b*) when b_1=0:

$$\begin{aligned} a_{2,ij} &= 0, && \text{if } j < i-1, \\ a_{2,ij} &= P_r(n_2, j, \lambda_2), && \text{if } i = 0, \\ a_{2,ij} &= P_r(n_2, j-i+1, \lambda_2), && \text{if } i \ge 1. \end{aligned}$$

Same as the analysis for the priority cells, the state of the non-priority cells at the end of the slot will make a Markov chain, so the following relationship can be obtained.

$$\begin{cases} \Omega = \Omega\mathbf{F} \\ \sum_{i=0}^{m_2} \omega_i = \mathbf{1}, \end{cases} \tag{2.16}$$

where $\mathbf{F}$ is a square matrix containing the transition probabilities f_{ij} of the non-priority cells, and Ω is a row vector of the state probabilities ω_i of the non-priority cells. They are given by

$$\mathbf{F} = \begin{bmatrix} f_{00} & f_{01} & \cdots & f_{0m_2} \\ f_{10} & f_{11} & \cdots & f_{1m_2} \\ \cdots & \cdots & \cdots & \cdots \\ f_{m_2 0} & f_{m_2 1} & \cdots & f_{m_2 m_2} \end{bmatrix}, \tag{2.17}$$

$$\Omega = \left[\omega_0 \quad \omega_1 \quad \cdots \quad \omega_{m_2}\right]. \tag{2.18}$$

After obtaining the state probabilities ω_i the relationship between the retransmission rate η and the probability that buffer is full, is easily given by

$$\eta = P_{rob2}(m_2), \tag{2.19}$$

where P_{rob2} is the state probability of the non-priority cells and it is given by

$$P_{rob2}(i) = \omega_i, \qquad i = 0, 1, 2, \cdots, m_2. \tag{2.20}$$

2.4.2.2 Average delay time T_{av}

It is very difficult to accurately analyze the average delay time for the non-priority cells because the service for the non-priority cells is strongly influenced by the priority cells, and the state 0 and state 1 of the priority source and the number of cells in the priority buffer will change in slot. Simply, we approximately evaluate the mean delay time with the method in [24] and it will be validated by computer simulation in section 2.5.

Let V be the average sum of the number of the priority cells and non-priority cells in the buffer when a non-priority cell just arrived and ν be the average arriving rate of priority sources. Thus, the mean delay T_{av} is given by

$$T_{av} = \frac{V}{1-\nu}, \tag{2.21}$$

where

$$V = \sum_{i=0}^{m_1} \sum_{j=0}^{m_2} (i+j) P_{rob1}(i) \omega_j, \tag{2.22}$$

$$\nu = E_a. \tag{2.23}$$

2.5 Performance Evaluation and Discussion

In this section, we discuss the performances of the priority cells and the non-priority cells by terms of the analytical results described in the previous sections.

Considering the characteristics of the priority source and the non-priority source, for example, the threshold of delay constraint, transmission speed of the system (e.g., 150 Mbits/s)[10, 12], we can appropriately take the priority buffer size m_1 to be 8 and m_2 to be 50 for the non-priority sources since the non-priority cell is sensitive to the discard probability. From our analytical model, the analytical performance of the priority cells can be accurately predicted and in order to validate the assumption underlying our queueing model, we compare the approximate analytical results with the performance results obtained via simulation.

Fig. 2.6 shows the discard probability of the priority cells as a function of buffer capacity for the cases of n_1=3, 4 and 5. Here, the ratio ξ is assumed to be 0.79 and the average talkspurt duration 5.56 slot units. According to the figure, we can design the optimal buffer size for a sort of source. Here we omit these natural things that the discard probability of the priority cell should become large when the arrival rate λ_1 or the number of sources increases.

However, changing our point of view, we obtained other interesting results illustrated in the following figures. In Fig. 2.7, we investigated the relationship between discard probability and the length of talkspurt when the traffic density

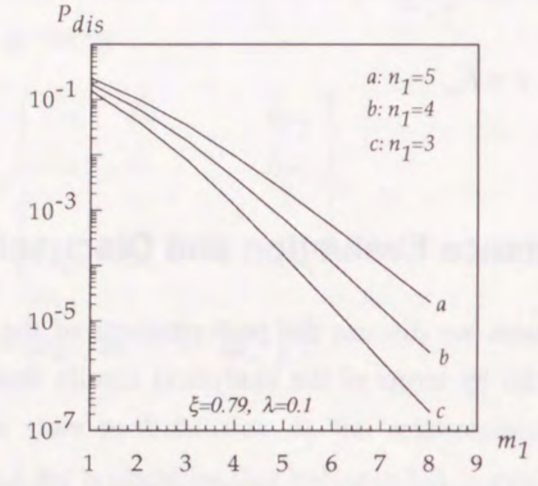


Fig. 2.6 Discard probability for priority cells vs. buffer size

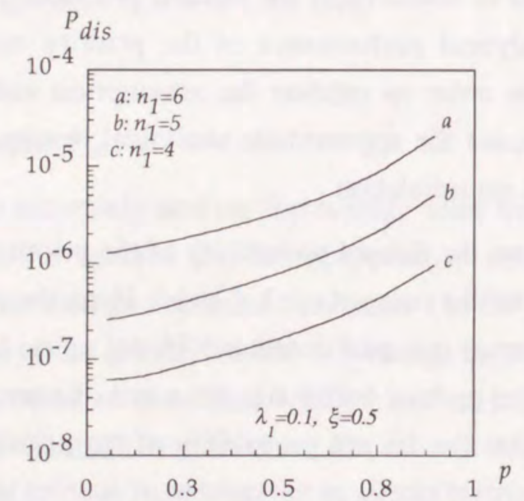


Fig. 2.7 Discard probability for priority cells vs. length of talkspurt

ρ_1 holds to be fixed 0.05. The result shows that the discard probability of the priority cells change into bigger when the period of talkspurt and silence becomes larger though the traffic density is constant.

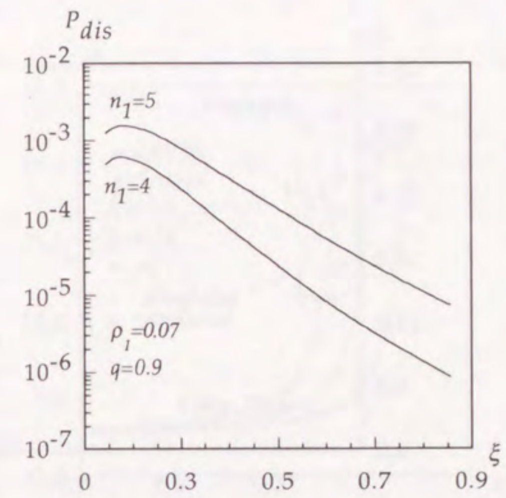


Fig. 2.8 Discard probability for priority cells vs. ratio of talkspurt

In Fig. 2.8, when the traffic density ρ_1 holds constant, the relative curve between discard probability and ratio ξ is shown. From the figure, we can apparently see that either the ratio ξ becomes larger or much lower, the discard probability becomes smaller, but at a certain point (i.e., in our example, where ξ=0.155), the discard probability has got a maximum. This phenomenon can be interpreted as follows: discarding cells occurs easily when the respective talkspurt of the priority sources pile up mutually or total arrival rate of the priority cells is larger. In this occasion, the traffic density ρ_1 is constant, while ξ becomes larger (approach 1), the superposition of talkspurt among the sources occurs easily, but the arrival rate changes into smaller, so, the discard probability of the priority cells also becomes smaller. On the other hand, when ξ becomes very smaller, the arrival rate changes into larger, but the talkspurt duration becomes very shorter, the superposition appears rarely, thus, the discard probability of the priority cells also becomes smaller.

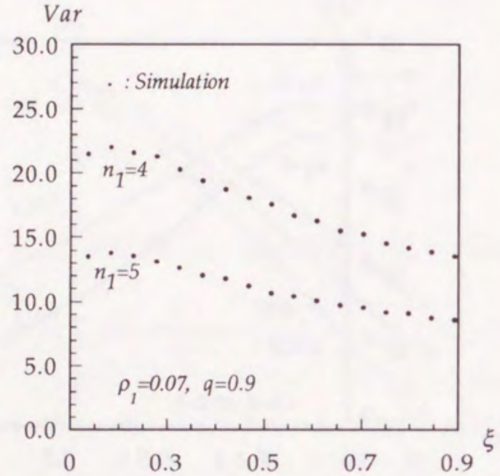


Fig. 2.9 Variance of arrival interval of priority cells versus ratio of talkspurt

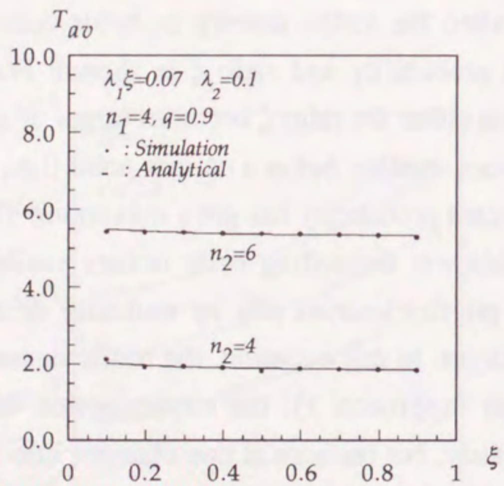


Fig. 2.10 Mean dealy of non-priority cells versus ratio of talkspurt

In order to investigate this reason, we inquired the variance of arrival interval of the priority cells in simulation. It is shown in Fig. 2.9. From the

figure, we can make out that the peak value appeared at the same point as in Fig. 2.8.

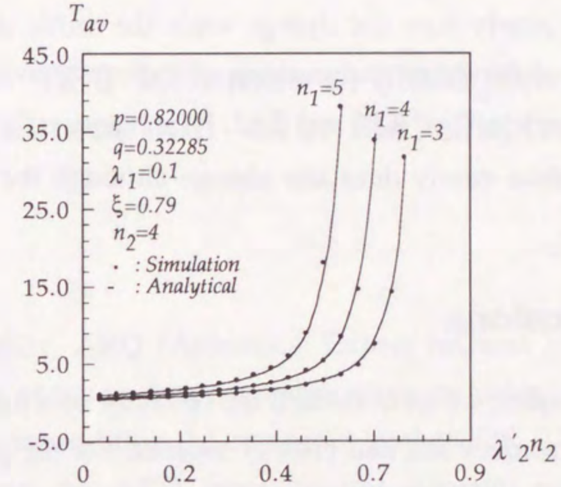


Fig. 2.11 Mean delay of non-priority cells versus arrival rate of non-priority sources

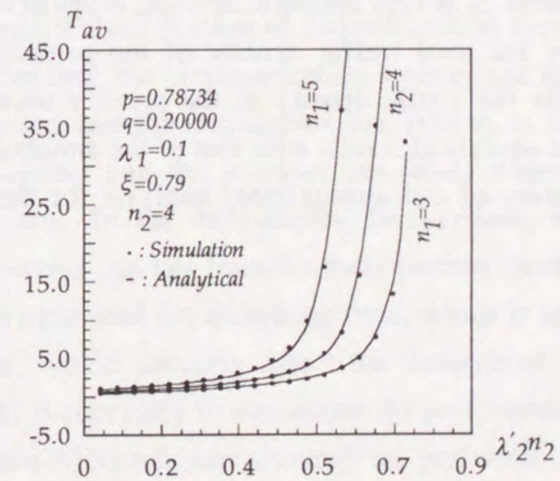


Fig. 2.12 Mean delay of non-priority cells versus arrival rate of non-priority sources

The performance of the non-priority cells under the influence of the priority cells is depicted in Fig. 2.10. The result shows the mean delay of the non-priority cells nearly does not change while the traffic density holds constant. The relation of the delay performance of the non-priority cells and its arrival rate λ_2 is shown in Figs. 2.11 and 2.12. From the two figures, it is clear that the mean delay time nearly does not change although the talkspurt and silence periods vary.

2.6 Conclusions

In this chapter, we have studied the behavior of a multiclass traffic system with priority sources and non-priority sources. For the priority cells, we focus on the discard probability, especially, the discard probability versus the variation of the arrival rate λ_1, traffic density ρ_1, talkspurt and silence periods. For the non-priority cells, we have paied our attention to the delay time. The results can be summarized as follows. First, the discard probability of the priority cells is much dependent upon talkspurt and silence durations although the traffic density ρ_1 is kept constant. Second, delay of the non-priority cells is influenced by the total traffic density of the priority cells, but it has no variation while the traffic density of the priority sources is constant. Last, comparing the analytical results with that of the simulation, it has been shown that the accuracy of our approximate analysis for the non-priority cells is sufficient.

3 ARQ Schemes in Dialogue Communication over Half-Duplex Line

As mentioned earlier, ARQ (Automatic Repeat reQuest) schemes are widely used to handle errors in data communications by using a simple error detection code and retransmitting the erroneous packets [26, 27]. From this chapter, we focus on the ARQ error control schemes under various communication environments. This chapter proposes the basic ARQ schemes in dialogue communication under half-duplex transmission line environment.

3.1 Introduction

Here, beginning from a classification of communication systems based on the form of transmission line, the communication systems can be classified as the full-duplex line system and the half-duplex line system. In the full-duplex line system, the transmitter and the receiver can freely transmit their data packets if there are any. In the half-duplex line system, however, the transmitter and the receiver can not transmit their packets freely unless they seize the transmission right, and the switching time, which is taken to across the transmission right, would strongly affect the throughput performance. These differences make it necessary to reexamine the performance of the half-duplex line under ARQ schemes although the performances of the full-duplex line system under ARQ schemes have been extensively studied [26, 27, 32-40, 69].

In the Local Area Networks (LAN), personal computer communication and mobile communication systems, under the consideration of cost, the half-duplex line systems are also used widely [30]. Unfortunately, there are few papers which studied the performance of the half-duplex line system under the ARQ schemes. Ref. [30] proposed a modified ARQ scheme in order to adapt to the variable transmission errors. Ref. [30] appears to have considered the half-duplex transmission line, however, the throughput was analyzed under the assumption that the receiver only receives the packets and does not generate any packets to send to the transmitter. In the half-duplex line system, the performance analysis is not sufficient under this assumption because the packets of the transmitter and receiver are transmitted each other over the same transmission line by controlling the transmission right. Especially in the dialogue communication environment which is considered in this chapter, this assumption is not applicable.

This chapter considers a dialogue communication system over half-duplex line under the ARQ error control schemes: SW, GBN and SR. The purposes are to analyze the throughput, to give some numerical results and clarify the influences of the switching and thinking times which exist in half-duplex line system.

The rest of this chapter is organized as follows. In Sect. 3.2, the system model and operations are described under basic ARQ schemes. In Sect. 3.3, the analytical assumptions are listed. In Sect. 3.4, we analyze the performances. For simplicity, a δ-function is introduced to get the Laplace-Transform (LT) of the distribution of the time taken to successfully transmit a message for SW and GBN schemes. For SR scheme, a set of the recursive equations is given to obtain the throughput. Some numerical results are examined in Sect. 3.5. The concluding remarks are summarized in Sect. 3.6.

3. 2 System Model and Operations

The system model under consideration consists of station A connected to station B over a half-duplex transmission line. The two stations are assumed to operate corresponding to the dialogue communication, namely, the station generates messages corresponding to the contents of the message arrived from its partner. The message is divided into some fixed-length packets. Station A (or B) does not generate any messages unless it receives all of the packets included in a message. As a station received a complete message, it will generate a corresponding message.

First let us introduce some terminology which will facilitate our discussion. Station A is said to be in *transmitting state* from the time it seizes the transmission right to the time it releases the right. On the contrary, station A is said to be in *receiving state* from the time it releases the transmission right to the time it seizes the right. The station is said to be *transmitting station* if it is in the transmitting state, and the station is said to be *receiving station* if it is in the receiving state. The station alternates between the transmitting station and the receiving station.

When a station received data packets, the transmission errors are checked in the data link layer. The error-free packets are passed to upper layer [31]. The station gets the transmission right after a switching time τ, and sends out an acknowledgment packet (AP). When a station completely received a message, it generates a corresponding message as a response. The time taken to generate a message, is called the *thinking time* (Fig. 3.1 (a)).

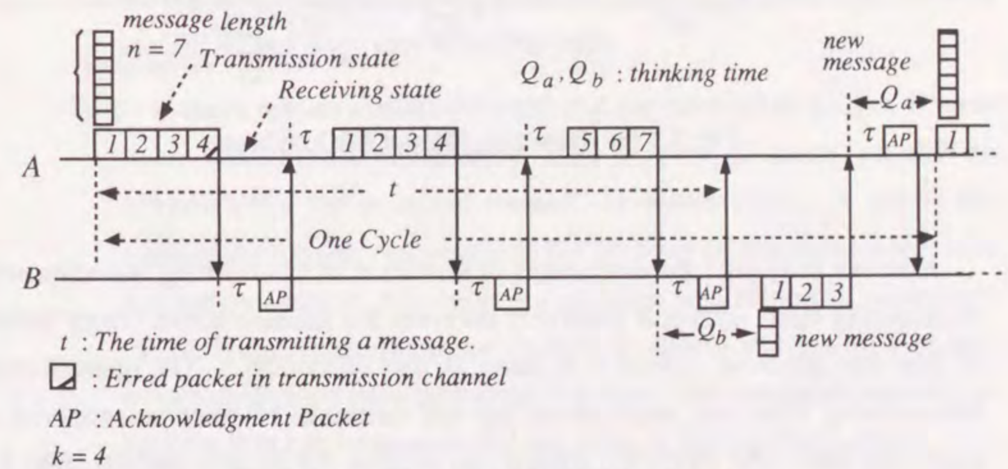


Fig. 3.1 (a) Stop-and-Wait ARQ protocol

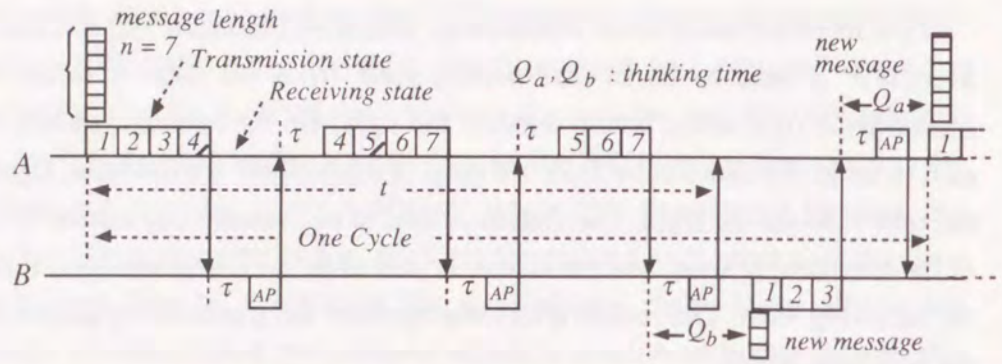


Fig. 3.1 (b) Go-Back-N ARQ protocol

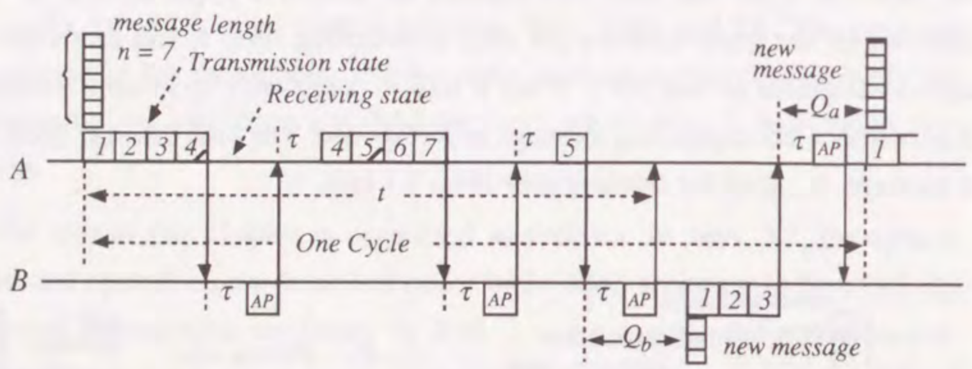


Fig. 3.1 (c) Selective-Repeat ARQ protocol

Here we only give the operations of station *A* in transmitting state because, in receiving state, station *A* passively receives the packets arrived from station *B*. The operations of station *B* is same as that of station *A*. The operations in transmitting state are determined by the contents of packets received in receiving state. The receiving station can receive AP or data packets. Fig. 3.1 shows an example of the operations of ARQ schemes, where 4 packets can be transmitted at a time. Notice that an AP contains the negative and/or positive acknowledgments for each received packet. The operations of station *A* in transmitting state are as follows.

(a) If station *A* received an AP in receiving state, according to the contents of the AP, the operation of station A is as follows;

(a-1) If the retransmission is requested, i.e. the transmission error occurred, the station *A* retransmits the erroneous packets and the other new packets which have not been transmitted if there are any, then releases the right to station *B*, and goes into receiving state.

(a-2) If no retransmissions are requested, however there exist some packets which have not been transmitted so far, then station *A* transmits the remaining packets, and then releases the right to station *B*. Station *A* goes into receiving state.

(a-3) If no retransmissions are requested and all of the packets have been transmitted, the station *A* continues to be in receiving state.

(b) If station *A* received data packets in receiving state, then station *A* checks for the errors, and operates as follows;

(b-1) If errors occurred, or if there are no errors occurred but all of the packets of station *B* have not been received, then station *A* sends out an AP to acknowledge the received packets, releases the right to station *B*, and goes into receiving state.

(b-2) If there are no errors occurred, and the packets of station *B* have been transmitted completely, then station *A* sends an AP to acknowledge the received packets. However station *A* keeps the transmission right. According to the contents of the packets received just before, station *A* generates a message as a response, packetizes the message, transmits these packets to station *B*, releases the right to station *B*, then goes into receiving state. The maximum number of packets that can be transmitted at a time, is constrained to be *k*.

Notice that the different ARQ schemes have the different retransmission rules. For SW scheme, if one of the packets transmitted at a time is received incorrectly, all the packets will be retransmitted no matter how the other packets are correct or not, e.g., in Fig. 3.1(a), all the packets 1, 2, 3 and 4 will be retransmitted although only packet 4 was received incorrectly. For GBN scheme, if the errors occurred, all the packets from the first erroneous packet will be retransmitted (Fig. 3.1 (b)). For SR scheme (Fig. 3.1 (c)), only these packets, which were received incorrectly, will be retransmitted. The resequencing buffer must be requested because the packets may be received out of order [40, 41, 47, 61, 67].

3.3 Analytical Assumptions

As a preparation of throughput analysis in the next section, the following assumptions are adopted.

A1. The packet errors occur independently with probability E and the acknowledgment packet is error-free.

A2. τ: the switching time of the half-duplex line system taken from the time that one station releases the transmission right to the time that the another station seizes the transmission right. Assume that τ is constant.

A3. A station does not generate any message unless it correctly receives all of the packets from its partner. If the message is correctly received by a station, this station generates one message as a response.

A4. Probability density function of thinking time in station A (B), denoted by $Q_a(t)$ ($Q_b(t)$), is assume to be exponential. It includes the time of packetizing message. $Q_a(t) = \alpha e^{-\alpha t}, \quad Q_b(t) = \beta e^{-\beta t}, \quad t \geq 0.$

A5. The message is divided into some fixed-length packets. The number of packets included in one message is assumed to be geometrical distribution.

$a(n)$ ($b(n)$): the probability that n packets are included in a message generated by station A (B).

$$a(n) = p_1^{n-1}(1-p_1), \quad b(n) = p_2^{n-1}(1-p_2), \quad 0 < p_1, p_2 < 1, \quad n = 1,2,3,\cdots$$

E_a (E_b): the average number of the packets included in one message generated by station A (B). $E_a = 1/(1-p_1), \quad E_b = 1/(1-p_2).$

A6. k : the maximum number of packets which can be transmitted at a time. Note that k=∞ means that the transmitting station transmits all the remaining packets at a time.

A7. The time of transmitting one packet is taken to be one unit time.

A8. The time taken to detect the packet errors, is much shorter than the switching time. Therefore, the station can transmit an AP at once when it gets the transmission right. The time of sending an AP is as long as that of sending a packet.

3.4 Performance Analysis

For convenience, we give some definitions which will facilitate our discussion.

Cycle period : The *cycle period* is defined as the time interval taken from the beginning of transmitting the first packet of a message to the beginning of transmitting the first packet of the next message in a station.

Dirac's δ-function: The Dirac's *δ-function* is defined as

$$\delta(t-t_0) = \begin{cases} \infty & \text{if} \quad t = t_0 \\ 0 & \text{otherwise}, \end{cases}$$

$$\int_{-\infty}^{\infty} \delta(t)dt = 1.$$

It has the following properties.

$$\int_{t_1}^{t_2} f(t)\delta(t-t_0)dt = f(t_0), \quad \text{if} \quad t_1 < t_0 < t_2,$$

where $f(t)$ is a function of t.

$p_n(t)$: the probability density function that the time taken to transmit a message, which contains n packets, is t, and its LT (Laplace-Transform), $\tilde{P}_n(s)$, is defined as

$$\tilde{P}_n(s) = \int_0^{\infty} p_n(t)e^{-st}dt. \tag{3.1}$$

In the system model, the message is divided into fixed-length packets, so the time of transmitting a packet can be consider as a unit time. However the switching time and the thinking time are not always a multiple of the unit time. Therefore, the system is a unit time of discrete and continuous time elements.

Using the δ-function, here the system model can be considered as a continuous time queueing model.

In this section, we derive the LT of distribution of the time taken to successfully transmit a message which contains n packets, and analyze the throughput, which is defined as the ratio of the *effective time interval* of transmitting data packets in one cycle to the time interval of one cycle, denoted by η_{SW}, η_{GBN} and η_{SR} corresponding to SW, GBN and SR protocols, respectively. If the message generated by station A (B) includes N_1 (N_2) packets, the effective time interval in the cycle is N_1+N_2 unit time.

3.4.1 Analysis for SW-ARQ scheme

The probability $a(n)$ can be rewritten as

$$a(n) = p_1^{n-1}(1-p_1), \quad n = ik+l, \quad i = 0,1,2,\cdots, \quad l = 1,2,3,\cdots,k.$$

Consider the transmission of a message which contains n (=$ik+l$) packets. If $n>k$, the message can not be transmitted completely at a time, although the first k packets are transmitted correctly. First, we get the $p_k(t)$ ($n=ik+l$), $\tilde{P}_k(s)$, and then the $\tilde{P}_l(s)$. Finally, using $\tilde{P}_k(s)$ and $\tilde{P}_l(s)$, we obtain $\tilde{P}_n(s)$.

Using the δ-function, $p_k(t)$ can be simply written as a recursive equation.

$$p_k(t) = (1-E)^k \delta(t-(k+2\tau+1)) + [1-(1-E)^k] p_k(t-(k+2\tau+1)). \tag{3.2}$$

Taking the LT with respect to t about (3.2), we get

$$\tilde{P}_k(s) = (1-E)^k e^{-s(k+2\tau+1)} + [1-(1-E)^k] e^{-s(k+2\tau+1)} \tilde{P}_k(s). \tag{3.3}$$

Thus, the $\tilde{P}_k(s)$ can be obtained as follows.

$$\tilde{P}_k(s) = \frac{(1-E)^k e^{-s(k+2\tau+1)}}{1-[1-(1-E)^k] e^{-s(k+2\tau+1)}}. \tag{3.4}$$

Now, we consider the $p_l(t)$ and $\tilde{P}_l(s)$, l=1, 2, $\cdots$, k. The message will be transmitted completely if the last l packets are transmitted correctly. Similarly, we have

$$\tilde{P}_l(s) = (1-E)^l e^{-s(l+\tau+1)} + [1-(1-E)^l] e^{-s(l+2\tau+1)} \tilde{P}_l(s).$$

and

$$\tilde{P}_l(s) = \frac{(1-E)^l e^{-s(l+\tau+1)}}{1-[1-(1-E)^l] e^{-s(l+2\tau+1)}}. \tag{3.5}$$

Based on the above preparation, $\tilde{P}_n(s)$, $n = ik+l$, is given by

$$\tilde{P}_n(s) = \left(\frac{(1-E)^k e^{-s(k+2\tau+1)}}{1-[1-(1-E)^k] e^{-s(k+2\tau+1)}} \right)^i \cdot \frac{(1-E)^l e^{-s(l+\tau+1)}}{1-[1-(1-E)^l] e^{-s(l+2\tau+1)}}. \tag{3.6}$$

Generally, we define $\tilde{P}(s)$ as $\tilde{P}(s) = \sum_{n=1}^{\infty} a(n) \tilde{P}_n(s)$. We get

$$\begin{aligned} \tilde{P}(s) &= \sum_{i=0}^{\infty} \sum_{l=1}^{k} p_1^{ik+l-1}(1-p_1) \left(\frac{(1-E)^k e^{-s(k+2\tau+1)}}{1-[1-(1-E)^k] e^{-s(k+2\tau+1)}} \right)^i \\ &\quad \cdot \frac{(1-E)^l e^{-s(l+\tau+1)}}{1-[1-(1-E)^l] e^{-s(l+2\tau+1)}} \\ &= \sum_{l=1}^{k} \frac{1-[1-(1-E)^k] e^{-s(k+2\tau+1)}}{1-[1-(1-E)^k + p_1^k (1-E)^k] e^{-s(k+2\tau+1)}} \\ &\quad \cdot \frac{(1-p_1) p_1^{l-1} (1-E)^l e^{-s(l+\tau+1)}}{1-[1-(1-E)^l] e^{-s(l+2\tau+1)}}. \end{aligned} \tag{3.7}$$

Hence, A_{SW}, the average time taken to correctly transmit a message at station A in one cycle, is given by

$$\begin{aligned} A_{SW} &= -\frac{d\tilde{P}(s)}{ds}\Big|_{s=0} = \frac{p_1^k (2\tau+k+1)}{(1-p_1^k)(1-E)^k} + \frac{1-p_1}{(1-E-p_1)(1-p_1^k)} \\ &\quad \cdot \left(\frac{(1-E)^k - p_1^k}{(1-E-p_1)(1-E)^{k-1}} - \frac{k p_1^k}{(1-E)^k} + \frac{(1-E)^k - p_1^k}{(1-E)^k}(2\tau+1) \right) - \tau. \end{aligned} \tag{3.8}$$

Similarly, B_{SW}, the average time taken to transmit a message at station B, can be obtained.

$$B_{SW} = \frac{p_2^k(2\tau+k+1)}{(1-p_2^k)(1-E)^k} + \frac{1-p_2}{(1-E-p_2)(1-p_2^k)} \cdot \left(\frac{(1-E)^k - p_2^k}{(1-E-p_2)(1-E)^{k-1}} - \frac{kp_2^k}{(1-E)^k} + \frac{(1-E)^k - p_2^k}{(1-E)^k}(2\tau+1) \right) - \tau. \tag{3.9}$$

In the special case of k=1,

$$A_{SW} = \frac{2\tau+2}{(1-p_1)(1-E)} - \tau, \quad B_{SW} = \frac{2\tau+2}{(1-p_2)(1-E)} - \tau. \tag{3.10}$$

In the special case of $k=\infty$, in equilibrium state (i.e., $p_1+E<1$ and $p_2+E<1$), we get

$$A_{SW} = (1-p_1)\left(\frac{1-E}{(1-p_1-E)^2} + \frac{2\tau+1}{1-p_1-E} \right) - \tau, \tag{3.11}$$

$$B_{SW} = (1-p_2)\left(\frac{1-E}{(1-p_2-E)^2} + \frac{2\tau+1}{1-p_2-E} \right) - \tau. \tag{3.12}$$

In the system model, new message is generated after a thinking time. So average length of one cycle, SW_{cycle}, is given by

$$\begin{aligned} SW_{cycle} &= \int_0^{\tau+1} A_{SW}\alpha e^{-\alpha t}dt + \int_{\tau+1}^{\infty}(A_{SW}+t-\tau-1)\alpha e^{-\alpha t}dt \\ &+ \int_0^{\tau+1} B_{SW}\beta e^{-\beta t}dt + \int_{\tau+1}^{\infty}(B_{SW}+t-\tau-1)\beta e^{-\beta t}dt \\ &= A_{SW} + \frac{1}{\alpha}e^{-\alpha(\tau+1)} + B_{SW} + \frac{1}{\beta}e^{-\beta(\tau+1)}. \end{aligned} \tag{3.13}$$

Therefore, the throughput η_{SW} is given by

$$\eta_{SW} = \frac{E_a + E_b}{SW_{cycle}}. \tag{3.14}$$

3.4.2 Analysis for GBN-ARQ scheme

For GBN scheme, we use the same method as used in previous to derive the throughput η_{GBN}. First, let us analyze the $p_n(t)$ and its LT, $\tilde{P}_n(s)$.

$a: 1 \le n \le k$

In this case, station A can transmit all of the n packets at a time. Simply, the recursive equation about $p_n(t)$ can be written as

$$p_n(t) = (1-E)^n \delta(t-(n+\tau+1)) + \sum_{i=0}^{n-1}(1-E)^i E p_{n-i}(t-(n+2\tau+1)). \tag{3.15}$$

So, $\tilde{P}_n(s)$, the LT of $p_n(t)$, is given by

$$\tilde{P}_n(s) = (1-E)^n e^{-s(n+\tau+1)} + \sum_{i=0}^{n-1}(1-E)^i E e^{-s(n+2\tau+1)} \tilde{P}_{n-i}(s). \tag{3.16}$$

Solving the recursive equation, we get

$$\tilde{P}_n(s) = \frac{(1-E)^n e^{-s(n+\tau+1)}}{\sum_{i=1}^{n}(1-Ee^{-s(i+2\tau+1)})}. \tag{3.17}$$

$b: n>k$

In this case, station A can not transmit all the n packets at a time. Thus the $p_n(t)$ and the $\tilde{P}_n(s)$ are given by

$$p_n(t) = (1-E)^k p_{n-k}(t-(k+2\tau+1)) + \sum_{i=0}^{k-1}(1-E)^i E p_{n-i}(t-(k+2\tau+1)), \tag{3.18}$$

$$\tilde{P}_n(s) = (1-E)^k e^{-s(k+2\tau+1)} \tilde{P}_{n-k}(s) + \sum_{i=0}^{k-1}(1-E)^i E e^{-s(k+2\tau+1)} \tilde{P}_{n-i}(s). \tag{3.19}$$

Here, we also define $\tilde{P}(s)$ as $\tilde{P}(s) = \sum_{n=1}^{\infty} a(n)\tilde{P}_n(s)$. So, according to (3.16) and (3.19), we get

$$\begin{aligned} \tilde{P}(s) &= \sum_{n=1}^{k} a(n)(1-E)^n e^{-s(n+\tau+1)} + \sum_{n=1}^{k}\sum_{i=0}^{k-1} a(n)(1-E)^i E e^{-s(n+2\tau+1)} \tilde{P}_{n-i}(s) \\ &+ \sum_{n=k+1}^{\infty} a(n)(1-E)^k e^{-s(k+2\tau+1)} \tilde{P}_{n-k}(s) \\ &+ \sum_{n=k+1}^{\infty}\sum_{i=0}^{k-1} a(n)(1-E)^i E e^{-s(k+2\tau+1)} \tilde{P}_{n-i}(s). \end{aligned}$$

$$
\begin{aligned}
&= \frac{(1-p_1)(1-E)e^{-s(\tau+2)}(1-p_1^k e^{-sk}(1-E)^k)}{1-p_1 e^{-s}(1-E)} \\
&+ \sum_{i=1}^{k} E(1-p_1)\cdot\frac{e^{-s(i+2\tau+1)}p_1^{i-1}-e^{-s(k+2\tau+1)}p_1^k(1-E)^{k-i+1}}{1-p_1 e^{-s}(1-E)}\cdot \tilde{P}_i(s) \\
&+ (1-E)^k p_1^k e^{-s(k+2\tau+1)}\tilde{P}(s) + \frac{Ee^{-s(k+2\tau+1)}(1-(1-E)^k p_1^k)}{1-p_1 e^{-s}(1-E)}\cdot \tilde{P}(s) \\
&- \sum_{i=1}^{k}\frac{1-(1-E)^{k-i+1}p_1^{k-i+1}}{1-p_1(1-E)}\cdot Ee^{-s(k+2\tau+1)}p_1^{i-1}(1-p_1)\tilde{P}_i(s).
\end{aligned} \tag{3.20}
$$

According to the properties of LT, $A_{GBN,}$ the average time taken to correctly transmit a message at station A, is given by

$$
\begin{aligned}
A_{GBN} &= -\frac{d\tilde{P}(s)}{ds}\Big|_{s=0} \\
&= (k+\tau+2) - \frac{k}{1-p_1^k(1-E)^k} + \frac{p_1(1-E)}{1-p_1(1-E)} \\
&+ \frac{p_1^k(1-E)^k(k+2\tau+1)(1-p_1(1-E))}{(1-p_1)(1-E)(1-p_1^k(1-E)^k)} \\
&+ \frac{E(k+2\tau+1)}{(1-p_1)(1-E)} + \frac{E(1-k-p_1^k)}{(1-p_1)(1-E)(1-p_1^k(1-E)^k)} \\
&+ \frac{Ep_1(1-p_1^k)}{(1-p_1)^2(1-E)(1-p_1^k(1-E)^k)} + \frac{p_1^k((1-E)^k-1)}{1-p_1^k(1-E)^k} \\
&+ \frac{Ep_1(1-p_1^k)}{(1-p_1)(1-p_1(1-E))(1-p_1^k(1-E)^k)} \\
&+ \frac{p_1^{k+1}(1-E)((1-E)^k-1)}{(1-p_1(1-E))(1-p_1^k(1-E)^k)}.
\end{aligned} \tag{3.21}
$$

Same as the previous discussion, B_{GBN}, the average time taken to correctly transmit a message at station B, can be simply given by

$$
\begin{aligned}
B_{GBN} &= (k+\tau+2) - \frac{k}{1-p_2^k(1-E)^k} + \frac{p_2(1-E)}{1-p_2(1-E)} \\
&+ \frac{p_2^k(1-E)^k(k+2\tau+1)(1-p_2(1-E))}{(1-p_2)(1-E)(1-p_2^k(1-E)^k)} \\
&+ \frac{E(k+2\tau+1)}{(1-p_2)(1-E)} + \frac{E(1-k-p_2^k)}{(1-p_2)(1-E)(1-p_2^k(1-E)^k)} \\
&+ \frac{Ep_2(1-p_2^k)}{(1-p_2)^2(1-E)(1-p_2^k(1-E)^k)} + \frac{p_2^k((1-E)^k-1)}{1-p_2^k(1-E)^k} \\
&+ \frac{Ep_2(1-p_2^k)}{(1-p_2)(1-p_2(1-E))(1-p_2^k(1-E)^k)} \\
&+ \frac{p_2^{k+1}(1-E)((1-E)^k-1)}{(1-p_2(1-E))(1-p_2^k(1-E)^k)}.
\end{aligned} \tag{3.22}
$$

In the special case of k=1, the results are same as that of SW scheme.

$$
A_{GBN} = \frac{2\tau+2}{(1-p_1)(1-E)} - \tau, \qquad B_{GBN} = \frac{2\tau+2}{(1-p_2)(1-E)} - \tau. \tag{3.23}
$$

In the special case of $k=\infty$,

$$
A_{GBN} = \frac{1}{1-p_1} + \tau + 1 + \frac{E}{1-E}\left(\frac{2\tau+1}{1-p_1} + \frac{1}{(1-p_1)^2}\right), \tag{3.24}
$$

$$
B_{GBN} = \frac{1}{1-p_2} + \tau + 1 + \frac{E}{1-E}\left(\frac{2\tau+1}{1-p_2} + \frac{1}{(1-p_2)^2}\right). \tag{3.25}
$$

So average length of one cycle, GBN_{cycle}, is given by

$$
GBN_{cycle} = A_{GBN} + \frac{1}{\alpha}e^{-\alpha(\tau+1)} + B_{GBN} + \frac{1}{\beta}e^{-\beta(\tau+1)}. \tag{3.26}
$$

Therefore, the throughput η_{GBN} is given by

$$
\eta_{GBN} = \frac{E_a + E_b}{GBN_{cycle}}. \tag{3.27}
$$

3.4.3 Analysis for SR-ARQ scheme

For SR scheme, we also use the same method to analyze the average time of transmitting a message and the throughput. We only give the recursive set of equations because it is very difficult or even impossible to derive the closed-form expressions. In Sect. 3.5, we use these recursive equations to approximately evaluate the throughput performance.

$a: 1 \le n \le k$

In the case, all of the n packets can be transmitted at a time. So the $p_n(t)$ and $\tilde{P}_n(s)$ are given as follows.

$$\begin{aligned} p_n(t) &= (1-E)^n \delta(t-(n+\tau+1)) \\ &+ \sum_{i=0}^{n-1}\binom{n}{i}(1-E)^i E^{n-i} p_{n-i}(t-(n+2\tau+1)), \end{aligned} \tag{3.28}$$

$$\tilde{P}_n(s) = (1-E)^n e^{-s(n+\tau+1)} + \sum_{i=0}^{n-1}\binom{n}{i}(1-E)^i E^{n-i} e^{-s(n+2\tau+1)} \tilde{P}_{n-i}(s), \tag{3.29}$$

$$\begin{aligned} \tilde{P}_n(s) &= \frac{1}{1-E^n e^{-s(n+2\tau+1)}} \\ &\cdot \left((1-E)^n e^{-s(n+\tau+1)} + \sum_{i=1}^{n-1}\binom{n}{i}(1-E)^i E^{n-i} e^{-s(n+2\tau+1)} \tilde{P}_{n-i}(s)\right). \end{aligned} \tag{3.30}$$

Thus, T_n, the average time taken to correctly transmit the n packets, is given by

$$\begin{aligned} T_n &= -\frac{d\tilde{P}_n(s)}{ds}\Big|_{s=0} \\ &= \frac{1}{1-E^n}\left(n+2\tau+1+\sum_{i=1}^{n-1}\binom{n}{i}(1-E)^i E^{n-i} T_{n-i}\right) - \tau. \end{aligned} \tag{3.31}$$

When n=1, $T_1 = (2\tau+2)/(1-E) - \tau$.

$b: n>k$

In this case, at most k packets will be transmitted at a time. The recursive equations of $p_n(t)$ and $\tilde{P}_n(s)$ are given by

$$p_n(t) = \sum_{i=0}^{k}\binom{k}{i}(1-E)^i E^{k-i} p_{n-i}(t-(k+2\tau+1)), \tag{3.32}$$

$$\tilde{P}_n(s) = \sum_{i=0}^{k}\binom{k}{i}(1-E)^i E^{k-i} e^{-s(k+2\tau+1)} \tilde{P}_{n-i}(s), \tag{3.33}$$

$$\tilde{P}_n(s) = \frac{1}{1-E^k e^{-s(k+2\tau+1)}} \cdot \sum_{i=1}^{k}\binom{k}{i}(1-E)^k E^{k-i} e^{-s(k+2\tau+1)} \tilde{P}_{n-i}(s). \tag{3.34}$$

So, T_n, the average time of transmitting n packets, is

$$\begin{aligned} T_n &= -\frac{d\tilde{P}_n(s)}{ds}\Big|_{s=0} \\ &= \frac{1}{1-E^k}\left(k+2\tau+1+\sum_{i=1}^{k}\binom{k}{i}(1-E)^i E^{k-i} T_{n-i}\right). \end{aligned} \tag{3.35}$$

Hence, A_{SR} (B_{SR}), the average time taken to correctly transmit a message at station A (B), can be obtained.

$$A_{SR} = \sum_{n=1}^{\infty} a(n) T_n, \tag{3.36}$$

$$B_{SR} = \sum_{n=1}^{\infty} b(n) T_n. \tag{3.37}$$

So the average length of one cycle, SR_{cycle}, is given by

$$SR_{cycle} = A_{SR} + \frac{1}{\alpha} e^{-\alpha(\tau+1)} + B_{SR} + \frac{1}{\beta} e^{-\beta(\tau+1)}. \tag{3.38}$$

Therefore, throughput η_{SR} is given by

$$\eta_{SR} = \frac{E_a + E_b}{SR_{cycle}}. \tag{3.39}$$

3.5 Numerical Examples

In this section, we discuss the throughput performance according to the analytical results described in the previous sections. The throughput is function of error rate, the switching time τ, α, β and k. For a given switching time τ and the error rate E, we give the optimal k through the numerical computation.

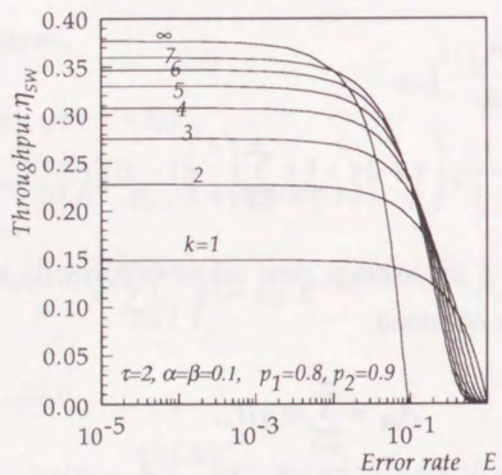


Fig. 3.2 Throughput of SW scheme

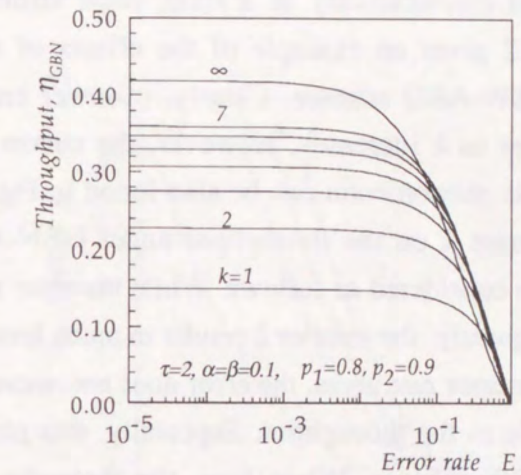


Fig. 3.3 Throughput of GBN scheme

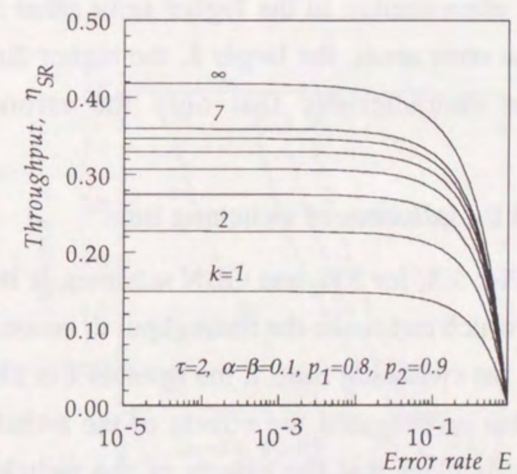


Fig. 3.4 Throughput of SR scheme

1) Influence of k on the throughput

From Eqs. (3.13), (3.26) and (3.38), k, which denotes the number of the packets transmitted consecutively at a time, must strongly influence the throughput. Fig. 3.2 gives an example of the effects of different k on the throughput under SW-ARQ scheme. Clearly, in lower error rate areas, the throughput increases as k increases. However, the curves reverse in higher error rate areas. This phenomenon can be also found in Fig. 3.3 which shows the effects of different k on the throughput under GBN-ARQ scheme. The phenomenon can be considered as follows. While the error rate is very higher, the error occurs frequently, the smaller k results in much less retransmissions at every time. In lower error rate areas, the error does not occur so frequently, the larger k is favorable to the throughput. Especially, this phenomenon is very significant under SW scheme. When $k=\infty$, the throughput of SW scheme decreases rapidly as the error rate increases. The throughput becomes zero in the high error areas $E \geq 0.1$. Note that the "$k=\infty$" means that the transmitting station transmits all the packets which have not been transmitted. Fig. 3.4 shows the effects of different k on the throughput under SR-ARQ scheme. Clearly, the reverse phenomenon in the higher error areas does not appear. It means that, in all the error areas, the larger k, the higher throughput under SR scheme due to the characteristic that only the erroneous packets are retransmitted.

2) Optimal k and the influence of switching time

In Fig. 3.2 and Fig. 3.3, for SW and GBN schemes, it is obvious that there exists an optimal k, which can make the throughput to become maximum. For a given error rate and the switching time τ, the optimal k is illustrated in Fig. 3.5 and Fig. 3.6. We also investigated the effects of the switching time τ. From Figs. 3.5 and 3.6, it is clear that the effects of the switching time τ is very remarkable. In a given error rate, the optimal k will become smaller when the switching time τ becomes smaller. It is worth noting that there has an interesting thing that the thinking time does not affect the optimal k. From analyzed results (expressions (3.13) and (3.26)), to obtain the maximal throughput, it only needs that the cycle interval is the shortest. Taking the derivative of equation (3.13) (or 3.26) about k, the term about the thinking time

will disappear. Therefore, the thinking time has no impact on the optimal k although it affects the throughput.

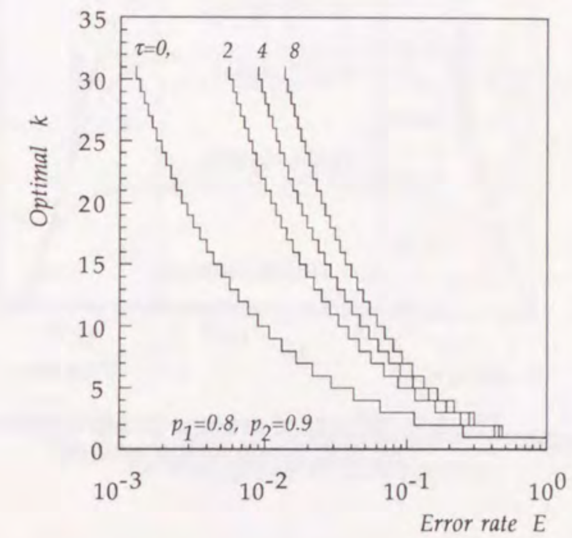


Fig. 3.5 Optimal k under SW scheme

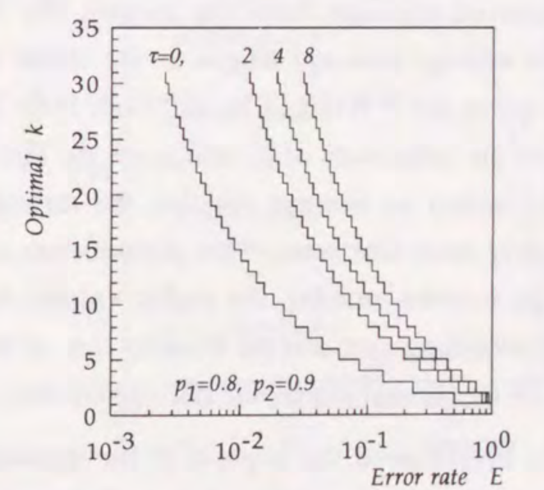


Fig. 3.6 Optimal k under GBN scheme.

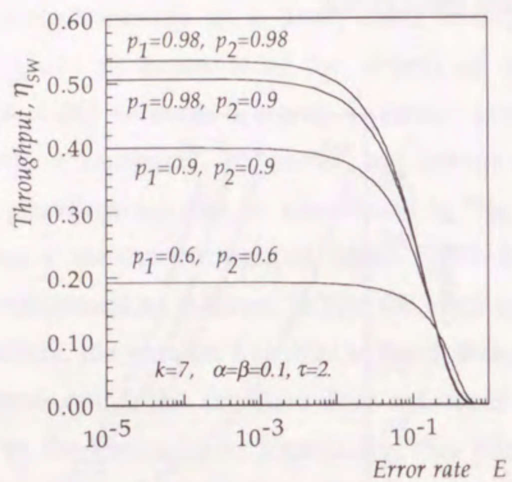


Fig. 3.7 Effects of average length of message on throughput of SW scheme.

3) Influence of the average message length

In the dialogue communication environment, the message is generated based on the received message from the partner, the throughput must be influenced by the average message length. In the above figures, the average message length (p_1 or p_2) is fixed to be identical. Here Figs. 3.7 ~ 3.9 give some examples of the influences of p_1 and p_2 on the throughput. From these figures, when p_1 and/or p_2 become smaller, the throughput also becomes smaller in the lower error rate areas. This phenomenon can be explained as, when p_1 and/or p_2 become smaller, the traffic volume becomes lower. The proportion of the switching time and the thinking time to the transmission time becomes heavy. So the throughput should become smaller.

We have also investigated the impacts of the thinking time and made a comparison among SW, GBN and SR schemes. It is sure that the throughput decreases as the thinking time increases, and that the throughputs satisfy $\eta_{SR} \geq \eta_{GBN} \geq \eta_{SW}$. Here we omitted these figures.

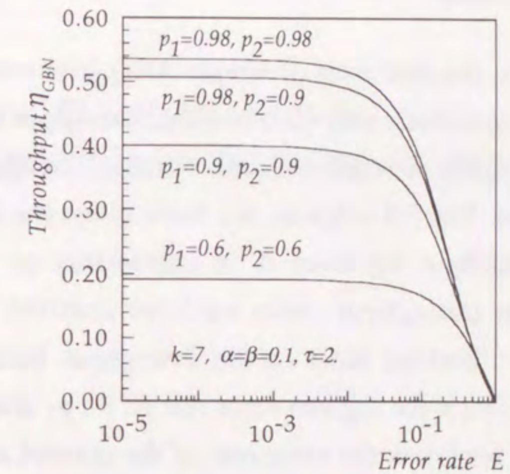


Fig. 3.8 Effects of average length of message on throughput of GBN scheme.

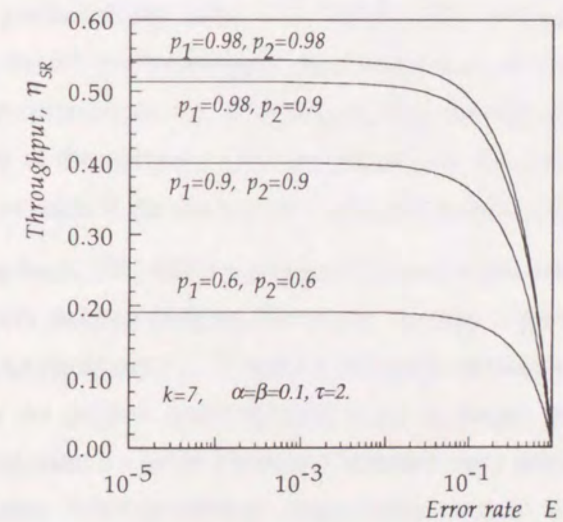


Fig. 3.9 Effects of average length of message on throughput of SR scheme.

3.6 Conclusions

In this chapter, the protocols of simple ARQ error control schemes over the half-duplex line have been proposed and the throughput has been analyzed. For SW scheme and GBN scheme, we have obtained the closed-form expressions of the throughput. For SR scheme, we have given the recursive equations to obtain the throughput because it is impossible to get the closed-form expression of the throughput. Also we have clarified the influences of the switching and the thinking times on the throughput. Based on the analysis, we have got the optimal k for a given error rate E, p_1, p_2 and the switching time τ. Therefore, when we know the error rate of the channel and the switching time τ, we can choose an optimal k to get the best efficiency of the usage of the system.

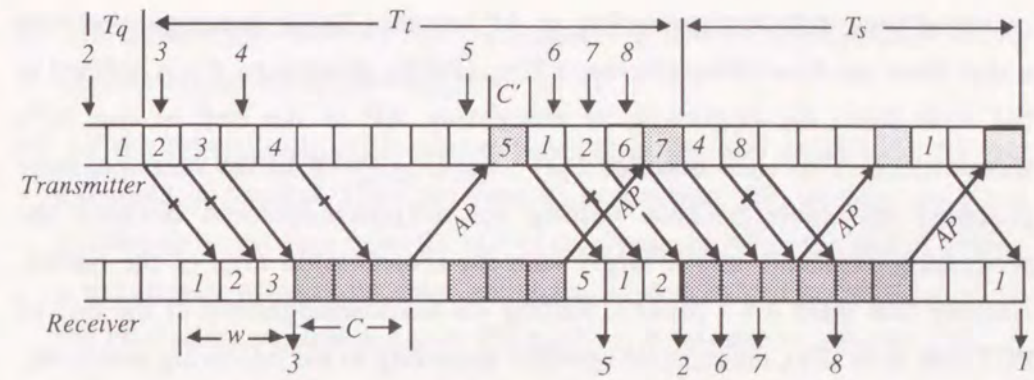
4 Block SR-ARQ Scheme in High-Speed Transmission System

4.1 Introduction

In the previous chapter, basic ARQ schemes, i.e., SW, GBN and SR, have been discussed in half-duplex line transmission system. There it is shown that the throughput of GBN scheme is better than that of SW scheme, and the throughput of SR scheme is the best one among them. In full-duplex line system, many studies about system performances, such as system throughput [38, 77, 78] and packet delay time [26, 50-54, 68], indicate that SR-ARQ scheme achieves better performances than the other two protocols at the expense of the requirement of a resequencing buffer and unavoidable resequencing delay at the receiver [35, 36, 41, 47, 48, 61, 67]. Therefore, SR-ARQ scheme is favorable if the resequencing buffer requirement is tolerable.

The idea of the basic SR-ARQ scheme is to retransmit only those packets which are negatively acknowledged. However, in very high-speed networks, such as ATM (Asynchronous Transfer Mode) networks, the ratio of propagation delay to packet transmission time is large, and the protocol processing overhead time for error recovery becomes very conspicuous [68]. In order to reduce these inferior effects, many error control schemes, such as window control schemes [78, 79], block acknowledgement schemes [80-81], are presented (an unit which consists of some packets is called a *block*) to tolerate the packet disorder and error due to the imperfect transmission channel. In Ref. [80], a scheme of erroneous or lost packet recovery has been presented, in which a group of packets is called block, and the receiver acknowledges a block instead of a single packet at a time. If one or more packets of a block are incorrectly received, the block including the erroneous packets would be retransmitted although the other packets in the block are received correctly. Obviously, it does not make the best use of SR-ARQ's advantage.

In this chapter, considering the high-speed environment, a novel SR-ARQ (called block SR-ARQ) scheme is proposed based on the principle of quickly, correctly providing the transmission service for "user", in which a single Acknowledgement Packet (AP) is used to acknowledge a block of packets. It inherits the advantage of the basic SR-ARQ scheme that is to retransmit only the erroneous packets. In Ref. [80], the receiver acknowledges a block of consecutive packets in original order. The original order means the sequence assigned while the message is packetized at the transmitting side. However, in block SR-ARQ scheme described in next section, the packets are acknowledged according to the accepted sequence at the receiver. The difference between the original order and the accepted sequence is caused by the retransmissions of the erroneous packets. The sequence of packets within an AP is called acknowledged sequence. So the acknowledged sequence is identical with the accepted sequence. The AP provides the positive or negative information for each packet in a block received by the receiver. We focus on the performance measurement of the system throughput and the average packet delay time.



$C'=1$ *AP: Acknowledgement Packet.*

× *:Erroneous Packets in Transmission Channel.*

: Processing Overhead Time at Receiver.

:Processing Overhead Time at Transmitter.

Fig. 4.1 Block selective repeat ARQ scheme

4.2 Protocols and Analytical Assumptions

The system under consideration consists of a transmitter, a transmission channel and a receiver. Time axis is divided into intervals of fixed length, called slot. A slot is taken to be the time of transmitting a packet. At the transmitting side, the transmitter retransmits the erroneous packets based on the received AP, or sends new packets which are waiting for transmission. The erred packets are superiorly retransmitted one by one in acknowledged sequence. At the receiving side, the arrival packet from transmitter is checked whether it is erroneous or not, and the results are recorded for acknowledgement. In our Block SR-ARQ scheme, a single AP is used to acknowledge a block of packets. The maximum number of packets acknowledged by an AP, denoted by w, is referred to *block size.* If a packet is received correctly, the receiver stores it temporarily in the resequencing buffer until all the packets that precede it in sequence are received correctly, then delivers them to the "user" in the original order. For instance (Fig. 4.1), packet 3 has to wait for packets 1 and 2 although it has been received correctly. In high-speed environments, the slot becomes shorter and the processing overhead time including generating an AP becomes larger in comparison with a slot. Here the *Processing Overhead Time* (POT), denoted by C, is defined as the time from the beginning of processing AP to the end of the AP's transmission. This state is called POT state (Fig. 4.1). In the receiver, there probably are some packets waiting for acknowledgement because the processing overhead time is larger than the transmission time of the packet. Assume that there are i packets waiting for acknowledgement at the end of POT (say n-th slot), the receiver operates according to the following protocols;

(1) if $w \le i$, begin to acknowledge the first w packets in the receiving buffer. The receiver enters in the POT state.

(2) if $0 < i < w$, observe the next slot ((n+1)-th slot);

(2-a) If one packet arrives during (n+1)-th slot, let the number of packets waiting for acknowledgement at the end of (n+1)-th slot be i, and $n \leftarrow n+1$, then go to (1) or (2).

(2-b) If there is no arrival packet during the (n+1)-th slot, begin to acknowledge the i packets from the beginning of (n+2)-th slot.

(3) If $i = 0$, wait for a new arrival packet. Also let the packets waiting for acknowledgement at the end of that slot in which the new packets arrived, be i, then go to (1) or (2).

Fig. 4.1 is an example using the protocols proposed above. Where s (in other literatures, s is called round-trip-delay), is defined as the sum of the propagation delay from the transmitter to receiver and its contrary delay. In Fig. 4.1, s is taken to be 4, the block size to be 3, and C to be 3 slots.

As the preparation for the analysis in the next sections, the following assumptions are used;

(1) Packet errors are assumed to be random with probability e and the feedback channel is error-free. So AP is always correctly received by the transmitter. Also, it never occurs that the packet disappears on the channel.

(2) The interarrival time of packets is distributed geometrically, namely, at most one packet arrives with the arrival rate λ_0 in a slot.

(3) The processing overhead time is assumed to be C slots regardless of the number of packets acknowledged by an AP.

(4) At the transmitting side, transmitting a packet and processing AP are executed in parallel. Here the time of processing AP, denoted by C', is defined to be the time from the end of receiving an AP to the end of getting a negative or positive message about the acknowledged packets, i.e., C' is the POT at the transmitting side. Certainly, it is suitable to assume $C' \le C$.

4.3 Performances Analysis

4.3.1 Throughput

Let's first analyze the throughput, which is defined as the maximum packet arrival rate that the system can support under the steady state, indicated by λ_{max}. The system consists of a transmitter, a transmission channel and a receiver. In order to get the system to work in the steady state, it is required that both the transmitter and the receiver are stable. So, the throughput can be obtained by considering the following two cases.

(A) $w \ge C$

In this case, the system stability is determined only by the packet arrival rate λ_0 and packet error rate e, because the processing ability of the receiver is larger than the packet arrival rate from transmitter. Thus, the system stability condition can be simply given by

$$\frac{\lambda_0}{1-e} < 1.$$

Hence, the throughput λ_{max} is given by

$$\lambda_{\max} = 1 - e. \tag{4.1}$$

(B) $w < C$

In this case, the system stability condition is determined by the three factors: the packet arrival rate λ_0, the packet error rate e, and the processing ability of the receiver. At receiver, it takes C slots to acknowledge at most w

packets. So, processing one packet at least needs C/w slots time. Reverse of C/w, the w/C (<1) can be considered as the processing ability of the receiver. If the total arrival rate satisfies $\lambda_0/(1-e)<1$, the transmitter will be stable. Thus, if $\lambda_0/(1-e)<w/C$ is satisfied, the receiver will be stable. Therefore, the system stability condition is given by

$$\frac{\lambda_0}{1-e}<\frac{w}{C}<1.$$

So the throughput λ_{max} can be obtained

$$\lambda_{\max}=\frac{w}{C}(1-e). \tag{4.2}$$

4.3.2 Packet delay

In this subsection, we analyze the average packet delay time. The packet delay time is defined from the arrival of a packet at the transmitter to correct delivery from the receiver in the original order. For convenience, the packet delay time T is divided into three parts: the queueing delay T_q, the transmission delay T_t, and the resequencing delay T_s, which are defined as follows;

T_q: the delay time from the arrival of a packet to its first transmission.

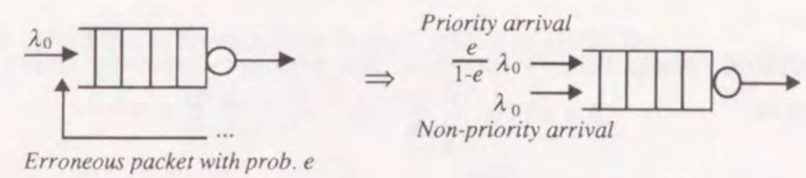
T_t: the delay time from the first transmission of a packet to its successful reception at the receiver.

T_s: the resequencing delay time in order to wait for all the packets precede itself in sequence at the receiver (Fig. 4.1).

It is very difficult to accurately analyze the packet delay time based on our block scheme, but in evaluating the system performance, it is necessary to investigate the delay time. We approximately give the average of T_q, indicated by $E[T_q]$, and the average of T_t, indicated by $E[T_t]$, which are validated by simulation. For the average of T_s, indicated by $E[T_s]$, it is measured by simulation. As we shall observe later, the effect of the average resequencing delay $E[T_s]$ on the average packet delay $E[T]$ is very heavy.

4.3.2.1 Queueing delay

The queueing delay T_q is defined to be the waiting time from the arrival of packet to its first transmission. Although the packets arrive geometrically, it is possible that the arriving packets are waiting temporarily for transmission in the queueing buffer because of the retransmissions caused by the imperfect channel. For analyzing the average queueing delay $E[T_q]$, we change the original arrival model at the transmitting side to a priority queueing model with two priority sources. Assume that the arrival of priority source λ_1 is randomly generated by the erred packets source with arrival rate $\lambda_1=\lambda_0 e/(1-e)$. The original arrival λ_0 is considered to be non-priority source (Fig. 4.2).



(a) Original arrival model (b) Analytical model for T_q

Fig. 4.2 Modeling for the analysis of queueing delay

So the average queueing delay of non-priority packet is approximately equivalent to the average queueing delay $E[T_q]$ of the original model. We observe the number of non-priority packets at the end of each slot. Let d and d_i be the number of the packets waiting for transmission and its probability at the end of a slot, respectively, i.e., d_i=$Prob(d=i)$. In the equilibrium state, the following equation can be obtained.

$$\begin{cases} d_0=d_0(1-\lambda_0)+d_0\lambda_0(1-\lambda_1)+d_1(1-\lambda_0)(1-\lambda_1), \\ d_i=d_{i-1}\lambda_0\lambda_1+d_i\left(\lambda_0(1-\lambda_1)+\lambda_1(1-\lambda_0)\right) \\ \quad +d_{i+1}(1-\lambda_0)(1-\lambda_1), \qquad i=1,2,3,\cdots. \end{cases} \tag{4.3}$$

Let $D(z)$ be the generating function of d_i, which is defined as

$$D(z)=\sum_{i=0}^{\infty}d_i z^i.$$

We get

$$D(z)=\frac{(1+\lambda_0\lambda_1-\lambda_0-\lambda_1)-(1-\lambda_0)(1-\lambda_1)z^{-1}}{1-(\lambda_0+\lambda_1-2\lambda_0\lambda_1)-\lambda_0\lambda_1 z-(1-\lambda_0)(1-\lambda_1)z^{-1}}\cdot d_0. \tag{4.4}$$

By the boundary condition, $D(1)=1$, and by using de l'Hospital's rule, d_0 is given by

$$d_0=\frac{1-\lambda_0-\lambda_1}{(1-\lambda_0)(1-\lambda_1)}.$$

So, the average number of non-priority packets, $E[S_L]$, is shown by

$$E[S_L]=D'(1)=\frac{\lambda_0\lambda_1}{1-\lambda_0-\lambda_1}. \tag{4.5}$$

Therefore, using Little's formula, the average queueing delay $E[S_L]$ is obtained as

$$E[T_q]=\frac{E[S_L]}{\lambda_0}=\frac{\lambda_1}{1-\lambda_0-\lambda_1}. \tag{4.6}$$

4.3.2.2 Transmitting delay

Unlike the basic SR-ARQ scheme where the time from the transmission of a packet to the return of its acknowledgement packet is $(s+1)$ slots, in our block scheme, the analysis becomes further complicated due to the fluctuation of the time from the transmission to its corresponding AP return. Here, we provide the approximate analytical results.

(A) $w \geq C$

First, we consider the sojourn time that a packet is in the receiving buffer waiting for its acknowledgement. In the case of $w \geq C$, the sojourn time of a packet is not longer than $(C+w)$ slots. Based on our protocols, the number of packets acknowledged by an AP can be from 1 to w. Let $P_w(i)$ be the probability that i packets are acknowledged by an AP. Assume that the packet arrival process in receiving side (i.e., the departure process in transmitting side) is also geometrical with arrival rate $\lambda=\lambda_0/(1-e)$. Let $f(i)$ be the probability that i packets consecutively arrive at the receiver , so we have

$$f(i)=\lambda^{i-1}(1-\lambda), \qquad i=1,2,3,\cdots. \tag{4.7}$$

Observe $jw+i$ packets consecutively arrived at the receiver from the transmitter. An AP acknowledges at most w packets. So the first preceding jw packets will be acknowledged by j AP's. For the last i packets, another AP is needed. Thus, considering the number of these AP's which are used to acknowledge v ($1\leq v \leq w$) packets and using $f(i)$, $P_w(i)$ is given by

$$\begin{cases} P_w(i)=\dfrac{1}{A}\displaystyle\sum_{j=0}^{\infty} f(jw+i), & i=1,2,3,\cdots \\ P_w(w)=\dfrac{1}{A}\displaystyle\sum_{j=1}^{\infty}\sum_{k=0}^{w-1} jf(jw+k). \end{cases} \tag{4.8}$$

where A is called the normalization factor, which is given by

$$\begin{aligned} A&=\sum_{i=1}^{w-1}\sum_{j=0}^{\infty} f(jw+i)+\sum_{j=1}^{\infty}\sum_{k=0}^{w-1} jf(jw+k) \\ &=\frac{1}{1-\lambda^w}. \end{aligned}$$

So $P_w(i)$ can be rewritten as

$$\begin{cases} P_w(i)=\lambda^{i-1}(1-\lambda), & i=1,2,3,\cdots \\ P_w(w)=\lambda^{w-1}. \end{cases} \tag{4.9}$$

Let $Q(k)$ denote the probability that the packet is received correctly by transmitted k times. So $Q(k)$ is simply given by

$$Q(k)=e^{k-1}(1-e), \qquad k=1,2,3,\cdots. \tag{4.10}$$

Now consider an AP which acknowledges i packets, assume that the test packet is randomly located at the l-th position in an AP, which means that there are $(l-1)$ packets at the front of the test packet. Let $E(l-1, y)$ be the probability that y packets in $(l-1)$ packets are erred, it is given as

$$E(l-1,y)=\binom{l-1}{y}e^y(1-e)^{l-1-y}, \qquad y=0,1,\cdots,l-1. \tag{4.11}$$

Assume that the packet retransmission is independent on the past history of system. So the average packet transmitting delay $E[T_t]$ is given as

$$E[T_t]=\sum_{k=1}^{\infty}(k-1)Q(k)\sum_{i=1}^{w}P_w(i)\sum_{l=1}^{i}\frac{1}{i}\sum_{y=0}^{l-1}E(l-1,y)$$
$$\cdot(s+C+C^{'}+i-l+y+1)+(s/2+1)$$
$$=\left(s+C+C^{'}+\frac{1-e}{2}\cdot\frac{1-\lambda^{w}}{1-\lambda}\right)\cdot\frac{e}{1-e}+\left(\frac{s}{2}+1\right). \quad (4.12)$$

(B) $w<C$

In the case that $w < C$, it is possible to get the packets waiting at the receiving buffer temporarily. Let $E[T\,']$ be the average waiting time (in slots) at the receiving buffer.

Define $p_C(j)$ to be the probability that j packets arrive at the receiver from the transmitter in C slots. According to the characteristics of the geometrical arrival process, $p_C(j)$ can be shown as

$$p_C(j)=\binom{C}{j}\lambda^{j}(1-\lambda)^{C-j}, \qquad j=0,1,2,\cdots,C. \quad (4.13)$$

Let $P_C(z)$ be the generating function of $p_C(j)$, so we get

$$P_C(z)=\sum_{j=0}^{C}p_C(j)z^{j}=(1-\lambda+\lambda z)^{C}. \quad (4.14)$$

Considering the relationship between C and w, we choose the end of POT as the observing point. Let $y_n(j)$ denote the state probability that there are j packets waiting for acknowledgement in the receiver at the n-th observing point. Therefore, we get the state equations as

$$\begin{cases} y_{n+1}(0)=\sum_{l=0}^{w}y_n(l)p_C(0), \\ y_{n+1}(j)=\sum_{l=0}^{w}y_n(l)p_C(j)+\sum_{k=0}^{j-1}y_n(w+j-k)p_C(k), \quad 1\le j\le C \\ y_{n+1}(j)=\sum_{i=0}^{C}y_n(j+w-i)p_C(i), \qquad j>C. \end{cases} \quad (4.15)$$

Combining the above expressions, and at the equilibrium state, $n\to\infty$, $y_{n+1}(j)\to y(j)$, we get

$$y(j)=\sum_{l=0}^{w}y(l)p_C(j)+\sum_{k=0}^{\min(j-1,C)}y(w+j-k)p_C(k). \quad (4.16)$$

Let the generating function of $y(j)$ be $Y(z)$, we have

$$Y(z)=\frac{P_C(z)\sum_{l=0}^{w}(1-z^{l-w})y(l)}{(1-P_C(z)z^{-w})}. \quad (4.17)$$

Using the boundary condition that $Y(1)=1$, we get

$$\sum_{l=0}^{w}(w-l)y(l)=w-P_C^{'}(1)=w-\lambda C. \quad (4.18)$$

The average number of packets in the receiving buffer, $E[Q_L]$, is given by

$$E[Q_L]=Y^{'}(1)$$
$$=\frac{1}{2(w-\lambda C)}\cdot$$
$$\left\{2P_C^{'}(1)+\sum_{l=0}^{w}(w(w-1)-l(l-1))y(l)+P_C^{''}-w(w-1)\right\}. \quad (4.19)$$

From Little's formula, $E[T\,']$ can be obtained

$$E[T^{'}]=E[Q_L]/\lambda. \quad (4.20)$$

In order to calculate $E[Q_L]$, we need the $y(0)$, $\cdots$, $y(w\text{-}1)$, which can be obtained through the numerical calculation (see appendix 4.A at the end of this chapter).

Hence, from $E[T\,']$, in the same way as in the case of $w\ge C$, the average transmission delay $E[T_t]$ can be obtained.

$$E[T_t] = \sum_{k=1}^{\infty}(k-1)Q(k)\sum_{i=1}^{w}P_w(i)\sum_{l=1}^{i}\frac{1}{i}\sum_{y=0}^{l-1}E(l-1,y)$$
$$\cdot(s+C+C^{'}+E[T^{'}]+i-l+y+1)+(s/2+1)$$
$$=\left(s+C+C^{'}+\frac{1-e}{2}\cdot\frac{1-\lambda^{w}}{1-\lambda}+\frac{E[Q_L]}{\lambda}\right)\cdot\frac{e}{1-e}+\left(\frac{s}{2}+1\right). \tag{4.21}$$

4.4 Performance Evaluation

In this section, we discuss the performance evaluation of the throughput and the average packet delay time (in slots) in terms of the analytical results described in the previous sections. The accuracy of our approximate analysis has been verified by simulation. Our simulation is done only according to the system protocols and assumptions (1)~(4), and the results of simulation are plotted with small dots.

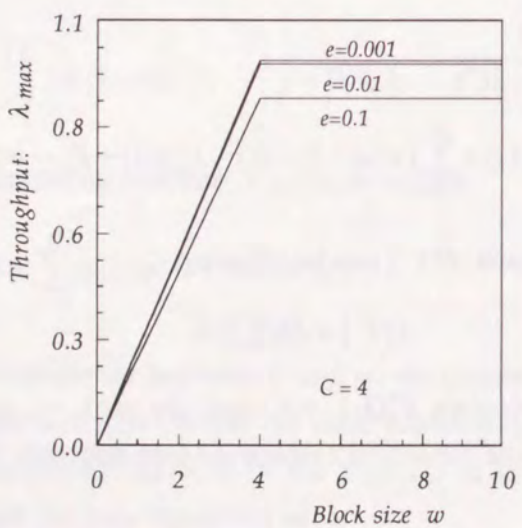


Fig. 4.3 System throughput

The throughput is a function of w, C, and the packet error rate e. Fig. 4.3 shows the effect of the value w on the throughput, where C=4. From this figure, we can plainly see that the throughput increases with the increment of block size w under the condition that $C \geq w$, when $C=w$, the throughput becomes

maximum, and while w exceeds C, i.e. $w \geq C$, the throughput reaches a constant just depending upon the packet error rate e. Therefore, we can make the conclusion that $w \geq C$ profits the throughput.

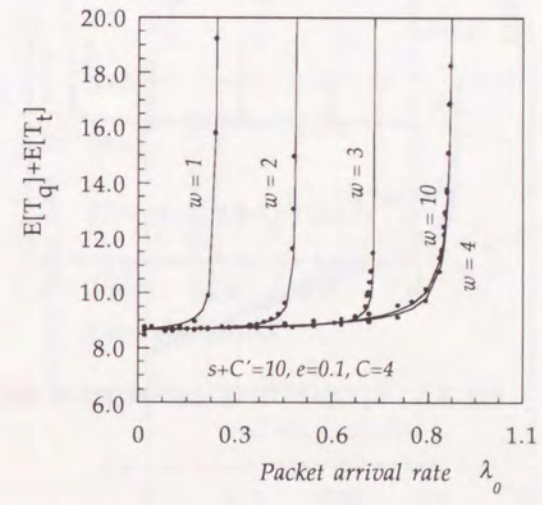


Fig. 4.4 Sum of E[Tq] and E[Tt]

From Fig. 4.4 to Fig. 4.6, we compare the analytical results (shown by consecutive curve) $E[T_q]$+$E[T_t]$ with simulation results, where e=0.1, 0.01, 0.001, respectively. From these figures, we can make out that (1) our approximate analysis has sufficient accuracy, (2) the effects of w on $E[T_q]$+$E[T_t]$ is very strong, namely when $w \leq C$, $E[T_q]$+$E[T_t]$ decreases with the increase of w, and when $w=C$, $E[T_q]$+$E[T_t]$ becomes minimum. On the contrary, when $w>C$, $E[T_q]$+$E[T_t]$ becomes larger although it is not so marked (Fig. 4.4). Especially when error rate becomes smaller, the delay performances at w≥C are almost identical. We also investigate the effects of s, when s becomes larger, the same conclusion can be gained, that is when $w=C$, the $E[T_q]$+$E[T_t]$ becomes minimum.

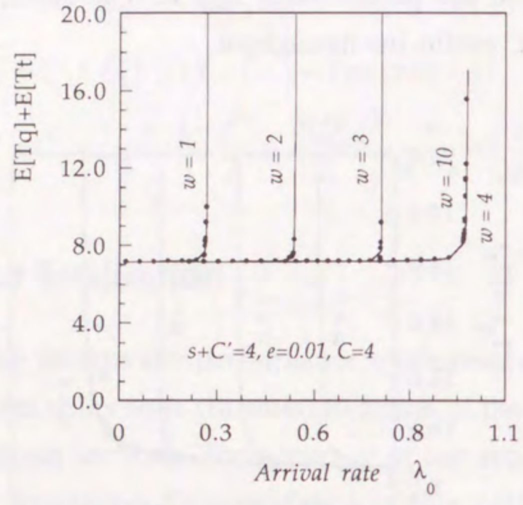


Fig. 4.5 E[Tq]+E[Tt] vs. packet arrival rate.

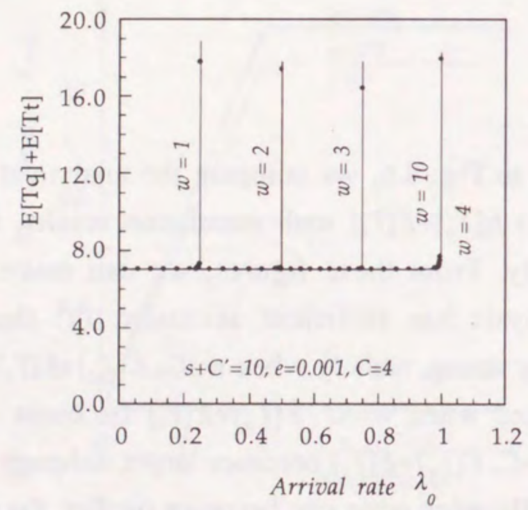


Fig. 4.6 E[Tq]+E[Tt]

In the previous sections, we have approximately analyzed $E[T_q]$ and $E[T_t]$. It is very difficult or even impossible to accurately analyze average resequencing delay $E[T_s]$. Here we only give the simulation results shown in

Figs. 4.7~ 4.9, where the average packet delay $E[T]=E[T_q]+E[T_t]+E[T_s]$. From these figures, $w=C$ is the optimal condition which gets the average packet delay to become minimum.

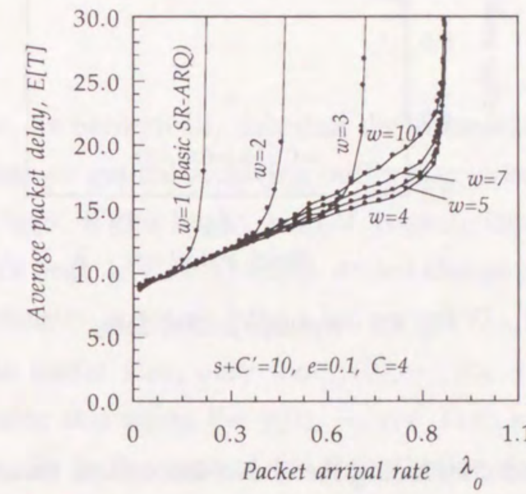


Fig. 4.7 Average packet delay

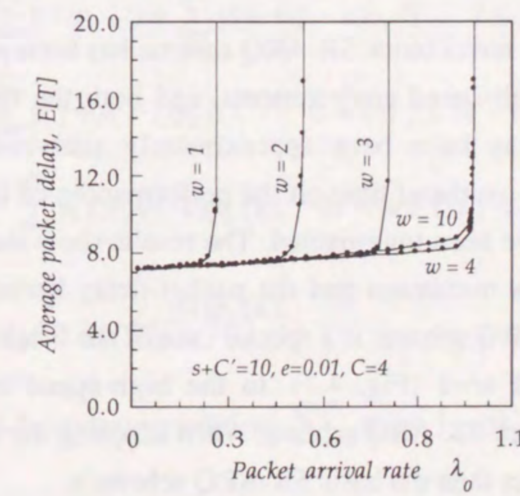


Fig. 4.8 Average packet delay E[T]

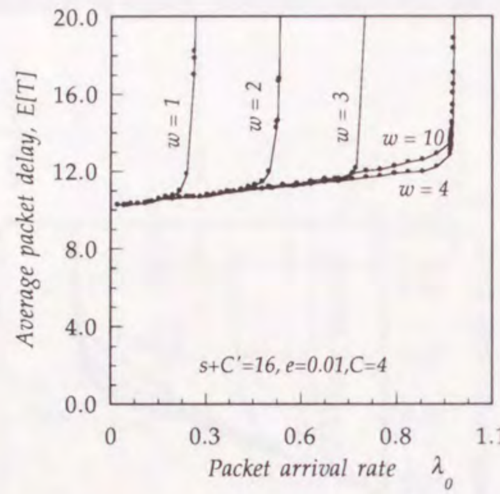


Fig. 4.9 Average packet delay

Finally, in a word concluding the above discussion, there exists an optimal block size, $w=C$, which obtains both the maximum throughput and the minimum packet delay.

4.5 Conclusions

In this chapter, a novel block SR-ARQ scheme has been proposed under the consideration of high-speed environments, and both the throughput and the average packet delay have been approximately analyzed. The effects of protocol processing overhead time on the performances of the throughput and the packet delay have been investigated. The results show that, when $w=C$, the throughput becomes maximum and the packet delay becomes minimum. In fact, the basic SR-ARQ scheme is a special case of the block SR-ARQ scheme just in the case of $w=1$ (Fig. 4.7). In the high-speed environments, the performances of block SR-ARQ scheme, when adopting the optimal block size ($w=C$), is much better than the basic SR-ARQ scheme's.

In this chapter, we have analyzed the performances of the system which consists of a transmitter, a transmission channel, and a receiver. We assumed that the arrival of new packets follows geometrical distribution and the

processing overhead time is constant regardless of the number of the packets acknowledged by an AP. As the next research, it is necessary to debate about multimedia traffic and/or bursty traffic [17, 82], and the performance while C is dependent on w.

Appendix 4.A

In this appendix, we numerically calculate the unknown probabilities $y(l)$'s, $l=0, 1, \cdots, w-1$. First, we get the receiving buffer size to be limited length M, just as observing later, if the buffer size M is sufficiently large, the state probabilities of $y(l)$'s ($l=0, 1, \cdots, w-1$) nearly do not change with the buffer size when the traffic intensity is not so large ($\lambda_0 C / w < 0.95$), furthermore, in the case of the infinite buffer size, only the $y(0), \cdots, y(w-1)$ are needed, it is sufficiently applicable that using the $y(0), \cdots, y(w-1)$ to analyze the case of infinite buffer size. So in the case of the finite buffer size, the equilibrium equations can be modified as followings;

$$\begin{cases} y(0) = \sum_{l=0}^{w} y(l) p_C(0), \\ y(j) = \sum_{l=0}^{w} y(l) p_C(j) + \sum_{k=0}^{j-1} y(w+j-k) p_C(k), & 1 \le j \le C \\ y(j) = \sum_{i=0}^{C} y(j+w-i) p_C(i), & C+1 \le j \le M-w \\ y(j) = \sum_{k=j-M+w}^{M} y(j+w-k) p_C(k), & M-w+1 \le j \le M-1 \\ y(M) = \sum_{i=M+w-C}^{M} \sum_{k=w+M-i}^{C} y(i) p_C(k). \end{cases} \tag{A.1}$$

From (A.1) and the boundary condition $\sum_{i=0}^{M} y(i) = 1$, $y(l)$, $l=0, 1, \cdots, w-1$ can be calculated.

Table 4.1 lists the numerical results of state probability $y(l)$'s where the buffer size is taken to be 30, 50, and 70. According to this table, the preceding state probabilities of $y(l)$'s (l=0, 1, ···, w-1) almost do not change when the buffer size is sufficiently large. Moreover the state probabilities $y(0)$, ···, $y(w\text{-}1)$ are required only in the case of $w<C$. So it is a reasonable approximation that we use the state probabilities $y(0)$, ···, $y(w\text{-}1)$ at the finite buffer size to analyze the case of infinite buffer size if the finite buffer size is relatively larger. In our performance evaluation, we take M=50.

Table 4.1 Numerical Results of $y(l)$'s , C =4, w =3, e =0.01

y_i	$\lambda_0 C/w$	M =30	M =50	M =70
y_0	0.133	6.530913e-1	6.530913e-1	6.530913e-1
	0.933	2.635126e-3	2.635116e-3	2.635116e-3
y_1	0.133	2.935924e-1	2.935924e-1	2.935924e-1
	0.933	2.713463e-2	2.713453e-2	2.713453e-2
y_2	0.133	4.950104e-2	4.950104e-2	4.950104e-2
	0.933	1.095432e-1	1.095428e-1	1.095428e-1
y_3	0.133	3.710817e-3	3.710817e-3	3.710817e-3
	0.933	2.185775e-1	2.185767e-1	2.185767e-1
y_4	0.133	1.044789e-4	1.044789e-4	1.044789e-4
	0.933	2.298054e-1	2.298045e-1	2.298045e-1
y_5	0.133	1.091698e-8	1.091698e-8	1.091698e-8
	0.933	1.475605e-1	1.475600e-1	1.475600e-1
y_6	0.133	1.140715e-12	1.140715e-12	1.140715e-12
	0.933	9.475023e-2	9.474987e-2	9.474987e-2
y_7	0.133	1.206640e-16	1.206473e-16	1.206471e-16
	0.933	6.084016e-2	6.083992e-2	6.083992e-2
y_8	0.133	1.461307e-18	1.466547e-18	1.466631e-18
	0.933	3.906613e-2	3.906598e-2	3.906598e-2
y_9	0.133	1.431995e-18	1.453922e-18	1.454176e-18
	0.933	2.508479e-2	2.508469e-2	2.508469e-2

5 GBN-ARQ Scheme with Channel-Grouping in Parallel-Channel -System

SR-ARQ scheme can achieve the best throughput performance at the expense of the requirement of resequencing buffer. However GBN-ARQ can also achieve very good performance if the error rate is not very high. In this chapter, GBN-ARQ scheme will be discussed in parallel-channel-system. The analysis focuses on the throughput and the channel-grouping method. An efficient Tree-Algorithm will be presented which is used to rapidly seek for the optimal channel-grouping.

5.1 Introduction

In error control schemes, generally, a specific scheme can not always achieve the best performances (e.g., the minimum packet delay, the maximum throughput, etc.) in any communication environments. Based on this reason, various diverse improved ARQ schemes are developed and studied [28, 29, 32-34, 39, 40, 63, 65, 71]. However, existing ARQ schemes are almost designed for sequential operation in which the transmitter sends a single packet at a time. In fact, data communication networks often provide multiple parallel channels between communicating entities [41]. For instance, in many packet networks, adjacent nodes are connected by multiple parallel channels, and in modern satellite communication systems, multiple frequency channels are provided between the earth terminal and the satellite.

Providing multiple parallel channels between transmitter and receiver has several benefits. *(1)* The more efficient transmission service would be provided for "user", namely, the packet delay time would be fairly decreased. *(2)* The packet error tolerance would be improved by means of usage of the multiple channels. *(3)* The throughput would be extremely improved by sending several packets or their copies at a time.

To realize the above potential of multiple channels, we have to discuss the channel-grouping methods (a grouped-channel is used to transmit a packet and its copies at a time), or the packet assignment methods. Especially, under the GBN error control scheme, the channel-grouping methods would have a very strong effect on system performance because a packet error on one channel results in retransmission of packets assigned to other channels. This phenomenon is very remarkable when the round-trip-delay is large. For example, suppose that the protocol operates over two channels, numbered 1 and 2, that have the same transmission rate and zero round-trip-delay. The time axis on the channels is partitioned into synchronized, packet-sized slots. At beginning of each slot, the transmitter sends two packets, one on each channel. Since an error in the packet with the lower (higher) sequence number requires the retransmission of two (one) packets, throughput is higher when the packet with the lower sequence number is sent over the channel with the lower error rate than the other way around. So it is necessary to discuss the packet assignment rule (in this chapter, generally called channel-grouping).

In this chapter, a transmission system is considered which consists of a transmitter and a receiver connected by multiple parallel channels on which the packet error probabilities are different and mutually independent. The transmitter sends packets to each channel in parallel at a time. The GBN-ARQ scheme is used to handle transmission error. The objective of this chapter is to evaluate the system throughput and discuss the optimal channel-grouping. First, three channel-grouping methods, called *OON*, *OOK* and *GCG*, will be proposed, and then an efficient tree-algorithm will be discussed. Finally, some numerical results will be depicted to show the efficiency of the optimal channel-grouping.

5.2 System Model and Go-Back-N Scheme

System model under consideration consists of a transmitter and a receiver connected by multiple parallel channels, numbered from 1 to N, which have different error probabilities (Fig. 5.1). The time axis is divided into packet-sized slots. The packet error probabilities of the N channels are supposed to be e_1, e_2, $\cdots$, and e_N, respectively. The round-trip-delay is assumed to be S slot time.

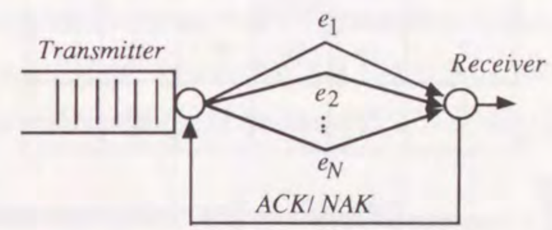


Fig. 5.1 A communication system with multiple parallel channels

The transmitter is able to send several packets to the receiver in a slot, one packet over one channel. When the channels are grouped, the transmitter concurrently sends a packet and its copies to a grouped-channel, i.e., a grouped-channel is used to transmit a packet and its copies at a time. The receiver checks for the packet errors, and sends an acknowledgment packet back to the transmitter over an feedback channel which is assumed to be error-free. An acknowledgment packet is used to acknowledge all the packets received in a slot. At the receiver, it is unnecessary to prepare the large resequencing buffer due to adopting GBN-ARQ error control scheme. The packets correctly received in order are delivered to "user" immediately.

slots	channels 1	2	3	4	5
1	1	2	3	4	5
2	6	7	8	9	10
3	11	12	13	14	15
4	4	5	6	7	8
5	9	10	11	12	13
6	14	15	16	17	18
7	19	20	21	22	23
8	11	12	13	14	15

S

: *Transmission Error.* S : *Round-Trip-Delay.*

Fig. 5.2 An example of GBN scheme used in multiple channels system

Here, we use an example, Fig. 5.2, to show the operation of GBN scheme used in the multiple channels system. In Fig. 5.2, the round-trip-delay, S, is taken to be 2 slot time and the number of channels, N, to be 5. The operation of the transmitter is shown under the very simple grouping method that takes one channel to transmit a packet at a time. In slot 1, packets from 1 to 5 are sent out. If there has no NAK coming back at the end of slot 1, the transmitter continuously sends packets. At the end of slot 3, the acknowledgment packet for these packets transmitted in slot 1, returns to the transmitter. The packets from 4 to 15 will be retransmitted immediately because of the error of packet 4. Similarly, packets from 11 to 15 will be retransmitted in slot 8.

In Fig. 5.2, it is clear that a packet error results in the retransmission of the packet of which the sequence is larger than that of the erroneous packet. Therefore, the channel-grouping methods have strong effects on the system performance. We focus on the channel-grouping methods in the next section.

5.3 Channel-Grouping Methods

5.3.1 One-to-One up to channel *N* method

The N channels, numbered from 1 to N, are connecting the transmitter with the receiver in parallel (Fig. 5.1). First, the simplest channel-grouping method is to send N packets in parallel in a slot, i.e., the smaller the packet's sequence number, the smaller sequence number is the channel to which it is assigned (Fig. 5.3 (a)). We call this channel-grouping method One-to-One up to channel N (OON) method because one packet is transmitted over one channel and all the N channels are used. Without loss of generality, hereby, we assume that $e_1 \le e_2 \le \cdots \le e_N$ is satisfied. In section 5.5.1, we will prove that, in OON method, the maximum throughput is obtained if $e_1 \le e_2 \le \cdots \le e_N$ is satisfied.

5.3.2 One-to-One up to channel *K* method

In the previous OON method, when the packet error probabilities of some channels are extremely high, the packet error over these channels would frequently occur. The packet error will result in retransmission of other packets transmitted in round-trip-delay. Therefore, using the channel of which error probability is high, conversely, is unfavorable to the system performance. Thus, the next channel-grouping method is considerable.

In one slot, only K packets are transmitted over the first K channels from 1 to K, one packet over one channel, the channels from K+1 to N are always idle, i.e. these channels are not used (Fig. 5.3 (b)). This method is called One-to-One up to channel K (OOK) method. Here we also assume that the condition, $e_1 \le e_2 \le \cdots \le e_N$, is satisfied.

Channels	1	2	3	4	5	6	7	....	*N*-1	*N*
Packet No.	*j*+1	*j*+2	*j*+3	*j*+4	*j*+5	*j*+6	*j*+7		*j*+*N*-1	*j*+*N*

(a) OON Channel-Grouping Method

Channels	1	2	3	4	5	....	*k*	....	*N*-1	*N*
Packet No.	*j*+1	*j*+2	*j*+3	*j*+4	*j*+5	......	*j*+*k*			

(b) OOK Channel-Grouping Method

Channels	1	....	*k*1	*k*1+1		*k*2	....	*kc*+1	....	*N*
Packet No.	*j*+1	...	*j*+1	*j*+2	...	*j*+2	......	*j*+*C*	...	*j*+*C*

(c) GCG Channel-Grouping Method

Fig. 5.3 Channel-grouping methods

5.3.3 General channel-grouping method

In the above *OOK* method, channels from *K*+1 to *N* are idle. It means that these channels are wasted. Hence, it is more efficient that grouping the higher error probability channels into other channels. The grouped-channel is taken to transmit one packet and its copies at a time (Fig. 5.3 (c)). This method is called General Channel-Grouping (*GCG*) method. In Fig. 5.3 (c), grouped-channel 1, which consists of K_1 channels, is taken to transmit packet *j*+1. Grouped-channel 2, which consists of K_2 channels, is used to transmit packet *j*+2, etc. Grouped-channel *C*, which consists of K_C channels, is used to transmit packet *j*+*C*. Here, $K_1 + K_2 + \cdots + K_C = N$.

5.4 Throughput Analysis

In this section, we analyze the throughput for each channel-grouping method discussed in previous section by assuming that the transmitting buffer is always in full state. The *OON* method and the *OOK* method can be considered as the specific cases of *GCG* method. Therefore, we first analyze the throughput for *GCG* method, and then specialize the results to obtain the throughput for *OOK* and *OON* methods.

Before analyzing the throughput, we, here, give some definitions which would facilitate our discussion.

Available packet: In GBN-ARQ scheme, a packet, of which the sequence number is larger than that of the erroneous packet, would be retransmitted even if it has correctly arrived at the receiver. The receiver would refuse to accept this packet, i.e., this packet would be ignored by the receiver. We call this packet unavailable packet. On the contrary, if a packet does not belong to unavailable packet, it is called *available packet*. For example, in Fig. 5.4, the packets from 1 to 4 are available packets, however packets from 5 to 15 are unavailable packets due to the error of packet 4.

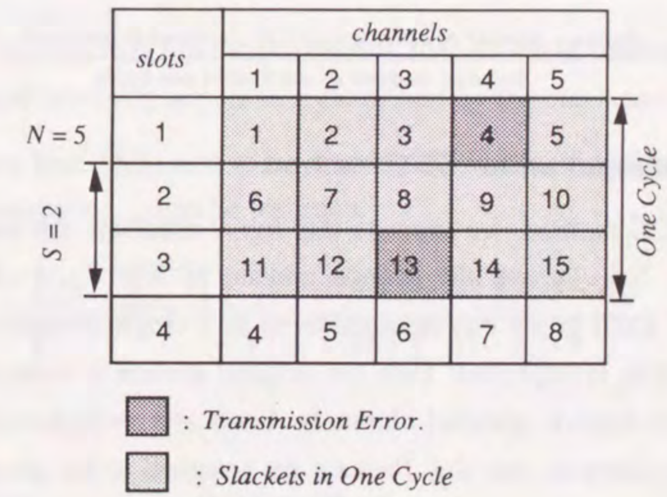


Fig. 5.4 Concepts of cycle and slacket

Available slot and Cycle: The *available slot* is defined as the slot in which at least one of the transmitted packets is available packet. The *cycle* is defined as the time interval from one available slot to the next available slot. In Fig. 5.4, packets transmitted in slot 1 can be accepted by the receiver. Packets from 6 to 15 transmitted in slots 2 and 3, however, are unavailable due to the transmission error of packet 4. Therefore, slots 1 and 4 are available and slots 2 and 3 are unavailable. The cycle interval is from the beginning of slot 1 to the end of slot 3. On the contrary, if all the packets transmitted in slot 1 are correctly received, slot 2 would be available. In this case, the cycle interval is from the beginning of slot 1 to the end of slot 1.

Slacket: The *slacket* is a fictitious quantity which is defined as the slot domain partitioned by the N channels. Hence, it only requires one slacket to send a packet. In Fig. 5.4, there are 5 slackets in one slot.

Generally, the throughput is defined as the ratio of the number of correct packets delivered to the total number of packets transmitted for a long period of time. However, using the concepts of cycle and slacket, the throughput, η, can be obtained by the following formula [42] under the assumption that the transmitting buffer is always full.

$$\eta = \frac{\text{Average number of packets correctly received in one cycle}}{\text{Average number of slackets in one cycle}}. \tag{5.1}$$

5.4.1 Throughput under *GCG* method

Under *GCG* method, we suppose that the N channels are distributed to C groups (Fig. 5.3(c)), and the groups consist of K_1, K_2, ···, K_C channels, respectively. Each group can be considered as a single channel of which the error probability is improved. Thus the original system is equivalent to a new system which have C parallel channels. Each grouped-channel is taken to transmit one packet in one slot. Packets are assigned to the grouped channels based on the packet sequence number and the error probabilities of grouped channels. The error probabilities of the C channels, denoted by E_i's (i=1, 2, ···, C) are given by

$$E_i = e_{K_{i-1}+1}e_{K_{i-1}+2}\cdots e_{K_i},\ (i=1,2,\cdots,C) \tag{5.2}$$

Analyzing the equivalent system, the throughput of original system can be obtained. Now, we observe a cycle interval to determine the average number of packets correctly received by the receiver, and the average number of slackets in one cycle, symbolized by E_P and E_S, respectively. Based on the definition of the cycle, only these packets transmitted in the first slot of the cycle are accepted and do not be ignored by receiver. If all the C packets are correctly received (with probability $\prod_{n=1}^{C}(1-E_n)$), the length of this cycle is only one slot. If packets transmitted by grouped-channel 1 to i are correctly received and the packets over the grouped-channel $i+1$ are received incorrectly, i ($1 \le i < C$) packets will be received correctly, and the length of the cycle is $S+1$ slots. Therefore, the average number of packets correctly received in one cycle, E_P, is given by

$$E_P = \sum_{i=1}^{C-1} i\prod_{j=1}^{i}(1-E_j)E_{i+1} + C\prod_{n=1}^{C}(1-E_n). \tag{5.3}$$

Meanwhile, average number of slackets in one cycle, E_S, is as follows.

$$E_S = \left(1-\prod_{i=1}^{C}(1-E_i)\right)(S+1)N + N\prod_{i=1}^{C}(1-E_i). \tag{5.4}$$

In Eq. (5.4), $\prod_{i=1}^{C}(1-E_i)$ is the probability that all the C packets transmitted in the first slot of the cycle are correctly received by the receiver.

Substituting Eqs. (5.3) and (5.4) into Eq. (5.1), the throughput for *GCG* method, denoted by η_{GCG}, can be obtained.

$$\eta_{GCG} = \frac{E_P}{E_S} = \frac{\sum_{i=1}^{C-1} i\prod_{j=1}^{i}(1-E_j)E_{i+1} + C\prod_{n=1}^{C}(1-E_n)}{(S+1)N - NS\prod_{i=1}^{C}(1-E_i)}. \tag{5.5}$$

5.4.2 Throughput under *OOK* method

For *OOK* method, K packets are transmitted over K channels, one packet over one channel. It can be considered as a special case of *GCG* method. Thus, let $C = K$, $E_i = e_i$, and substituting them into Eq. (5.5), the throughput under *OOK* method, denoted by η_{OOK}, is given by

$$\eta_{OOK} = \frac{\sum_{i=1}^{K-1} i\prod_{j=1}^{i}(1-e_j)e_{i+1} + K\prod_{n=1}^{K}(1-e_n)}{(S+1)N - NS\prod_{i=1}^{K}(1-e_i)}. \tag{5.6}$$

For a given parameters, e_i, and S, under *OOK* method, using the above expression, η_{OOK}, we can compute the optimal K which can get the η_{OOK} to become maximum.

5.4.3 Throughput under *OON* method

In the same way as in the previous, *OON* method can be considered as the specific case of *OOK* method. Let $K = N$, from Eq. (5.6), the throughput, denoted by η_{OON}, is given by

$$\eta_{OON} = \frac{\sum_{i=1}^{N-1} i\prod_{j=1}^{i}(1-e_j)e_{i+1} + N\prod_{n=1}^{N}(1-e_n)}{(S+1)N - NS\prod_{i=1}^{N}(1-e_i)}. \tag{5.7}$$

5.5 Channel-Grouping Algorithm for *GCG* Method

5.5.1 Necessary condition for optimal channel-grouping

Theorem 1 Under *OON* method, the N channels are numbered from 1 to N, of which error probabilities are e_1, e_2, $\cdots$, e_N, respectively. In comparison with other sequence of the N channels, if the condition $e_1 \le e_2 \le \cdots \le e_N$ is satisfied, the system is most efficient, i.e., the throughput, η_{OON}, is maximum.

Proof: Let the throughput be η while the error probabilities of N channels are given as e_1, e_2, $\cdots$, e_m, $\cdots$, e_n, $\cdots$, e_N. Here, without loss of generality, we focus on the channels m and n ($n>m$), and assume that $e_m < e_n$ (the error probabilities of other channels have no limitation). In this case, from Eq. (5.7), throughput, η, is given by

$$\eta = \frac{\sum_{i=1}^{N-1} i\prod_{j=1}^{i}(1-e_j)e_{i+1} + N\prod_{n=1}^{N}(1-e_n)}{(S+1)N - NS\prod_{i=1}^{N}(1-e_i)}.$$

Next, we change the channels m and n to get another usage of these channels. The error probabilities of N channels from 1 to N are e'_1, e'_2, $\cdots$, e'_m, $\cdots$, e'_n, $\cdots$, e'_N. Here, $e'_m > e'_n$, $e'_1 = e_1$, $e'_2 = e_2$, $\cdots$, $e'_{m-1} = e_{m-1}$, $e'_m = e_n$, $e'_{m+1} = e_{m+1}$, $\cdots$, $e'_n = e_m$, $\cdots$, $e'_N = e_N$. From Eq. (5.7), also, the throughput η' is as follows.

$$\eta' = \frac{\sum_{i=1}^{N-1} i\prod_{j=1}^{i}(1-e'_j)e'_{i+1} + N\prod_{n=1}^{N}(1-e'_n)}{(S+1)N - NS\prod_{i=1}^{N}(1-e'_i)}.$$

Therefore, we can calculate η-η' to determine that which of them is larger. By some simplifications, we get

$$\eta - \eta' = \begin{cases} \dfrac{\prod_{j=1}^{m-1}(1-e_j)(e_n - e_m)A(m,n)}{(S+1)N - NS\prod_{i=1}^{N}(1-e_i)}, & m \ne 1, \\[2ex] \dfrac{(e_n - e_1)\left[1 + \prod_{k=2}^{n-1}(n-2)(1-e_k) + \sum_{l=1}^{n-3} l\prod_{k=2}^{l+1}(1-e_k)e_{l+2}\right]}{(S+1)N - NS\prod_{i=1}^{N}(1-e_i)}, & m = 1. \end{cases} \tag{5.8}$$

where $A(m,n)$ is given as

$$A(m,n) = 1 + \prod_{k=m+1}^{n-1}(n-m-1)(1-e_k) + \sum_{l=1}^{n-m-2} l \prod_{k=m+1}^{l+m}(1-e_k)e_{l+m+1}.$$

In the simplification of η-η', the following equation is used.

$$e_1 + \sum_{i=1}^{k-1}\prod_{j=1}^{i}(1-e_j)e_{i+1} + \prod_{j=1}^{k}(1-e_j) = 1. \tag{5.9}$$

In above expression (5.8), $\prod_r^s(\cdot)=0$ and $\sum_r^s(\cdot)=0$, if $s<r$. From Eq. (5.8), it is clear that $\eta - \eta' \ge 0$ for $e_n > e_m$. When $e_m = e_n$, $\eta = \eta'$. Hence, the theorem has been proved.

According to the theorem, the following lemma is obtained.

Lemma In *GCG* method, assume that the channels are distributed into C groupings, and that the error probability of the grouped-channel i is E_i (i =1, 2, ···, C). The grouped-channel which has a lower sequence, is assigned to transmit the packet of which the sequence number is smaller. Under the given groupings, if $E_1 \le E_2 \le \cdots \le E_C$ is satisfied, the throughput would get to be maximum.

5.5.2 Tree-algorithm for optimal channel-grouping

According to the lemma described above, the optimal channel-grouping must be satisfied the condition $E_1 \le E_2 \le \cdots \le E_C$. Here we assume that the number of channels is N, the packet error probability of channel i is e_i for $1 \le i \le N$, and $e_1 \le e_2 \le \cdots \le e_N$. We present a efficient tree-algorithm taken to search all of the channel groupings which are satisfied the condition $E_1 \le E_2 \le \cdots \le E_C$.

Tree-algorithm: TA

Initial tree First, we make up a *initial tree* which consists of N+1 nodes numbered from 0 to N (Fig. 5.5(a)). The N+1 nodes have the relation that node i ($1 \le i \le N$) is a child [43] of node i-1. Node 0 is the root of the initial tree. Channel i ($1 \le i \le N$) is assigned to node i. Note, for simplicity, that the error probability of node i means the error probability of the grouped-channel included in node i. Therefore, the error probability of node i of the initial tree is e_i. We assume that e_0=0. Let *Num* denote the number of the nodes in the tree, i.e., *Num*=N+1.

(A) Let r represent the leaf-node in the initial tree, that is r=N.
(B) mark node r; let i= node r, j= the parent of node i and h= the parent of node j;
(C) Generate a grouped-channel x which consists of the channels included in nodes i and j; Denote its error probability by ε ;
(D) Denote the error probability of node h by ω,
if $\varepsilon<\omega$, { let h= the parent of node h, go to (D); }
if $\varepsilon \ge \omega$
{

generate a node y representing the grouped-channel x;
make node y as a child of the node h;
assign the number of *Num* to this child node; *pt* =*Num*;
execute M-copy ;
let w = the parent of node j , h= the parent of node w and j = node w;
if $j \ge 1$ go to (C);
if j =0 (the number of the root),
{ i = the parent of node i, j = the parent of node i;
h = the parent of node j;
if i>1 go to (C);
if i=1
{ if there is no un-marked leaf-node,
then {terminate the tree-algorithm *TA*; }
else { let r be one of the un-marked leaf-nodes, go to (B); }
}
}
}

M-copy
Let V = node h and u = node *pt*;
M_1: V = the child of node V which is in the path from root to the leaf-node r;
if V= node j or V= node i
{ if V = leaf-node r then { terminate the M-copy;}
else { go to *M_1*;}
}
if V is not node j and not node i,
{ *Num* = *Num*+1;
generate a node b which is a copy of node V, assign the number *Num* to node b, and make it to be a child of node u, u= node *Num*;
if V= leaf-node r then { terminate M-copy; }
else {go to *M_1*;}
}

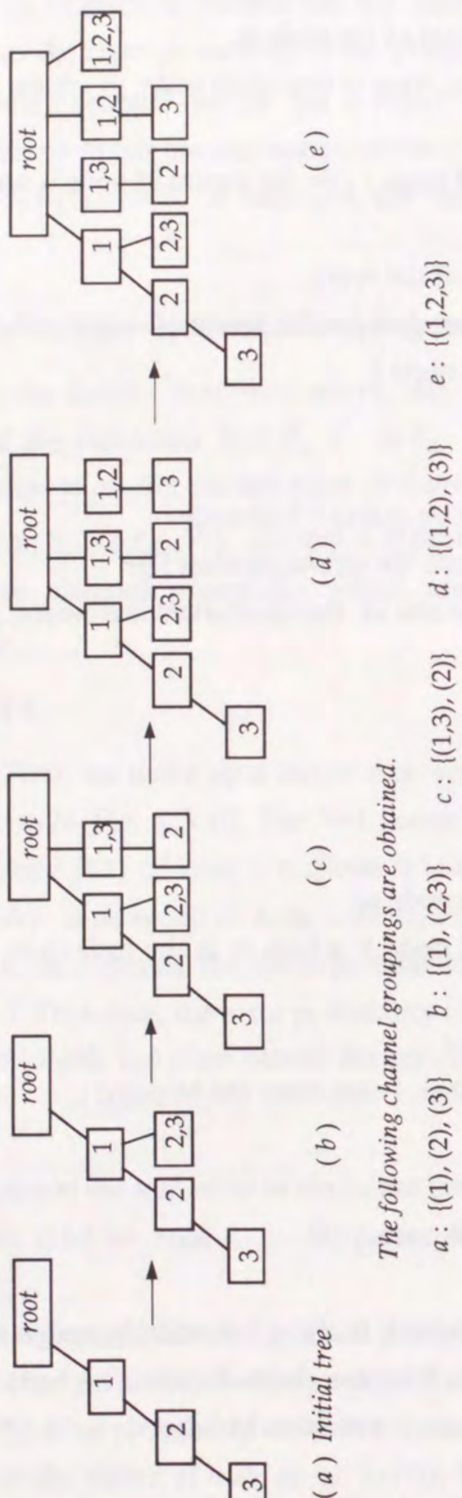


Fig. 5.5 An example of the tree-algorithm.

$N=3$, $e_1=0.0001$, $e_2=0.01$, $e_3=0.1$

Thus every path from root to leaf-nodes in the tree provides a channel-grouping. Calculating the throughput according to Eq. (5.5) for each channel-grouping, the optimal channel-grouping can be obtained. Fig. 5.5 is a searching process used the above tree-algorithm, where N=3, e_1=0.0001, e_2=0.01, e_3=0.1. When S=10, channel-grouping b is optimal one, where the channels are grouped as {(1), (2,3)}. The maximal throughput is 0.659018.

Note that in the above tree-algorithm, some channel-groupings would appear more than once. We call the channel-grouping which has appeared redundant channel-grouping. Example, in Fig. 5.5(e), from the grouping {(1,3),(2)} , grouping {(1,2,3)} can be obtained. On the other hand, from grouping {(1,2),(3)}, the same grouping {(1,2,3)} is also obtained. This problem can be solved by deleting the redundant channel-grouping from the tree.

5.6 Performance Evaluation

In this section, some numerical results are shown for different channel-grouping methods developed above. In all the results we present, N=8, S=10. In the previous discussion, error probabilities of channels are assumed to be independent. In order to clearly illustrate the performance by two-dimensional graph, we assume that the error probabilities of channels are in proportion, i.e., $p=e_i/e_{i-1}$, $(2 \le i \le N)$. Thus, the system is characterized by the number of channels, the round-trip-delay S and the error probability of channel 8.

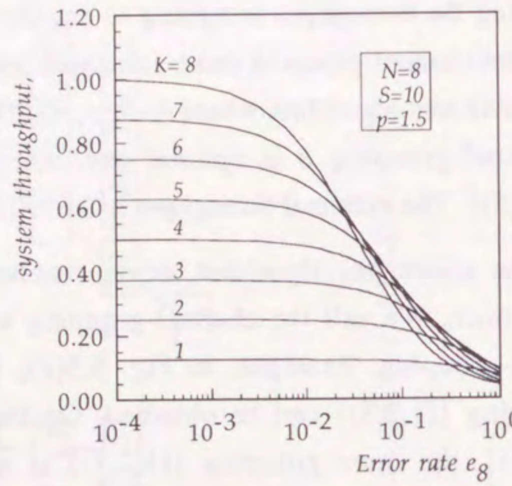


Fig. 5.6 Effects of different K under OOK method

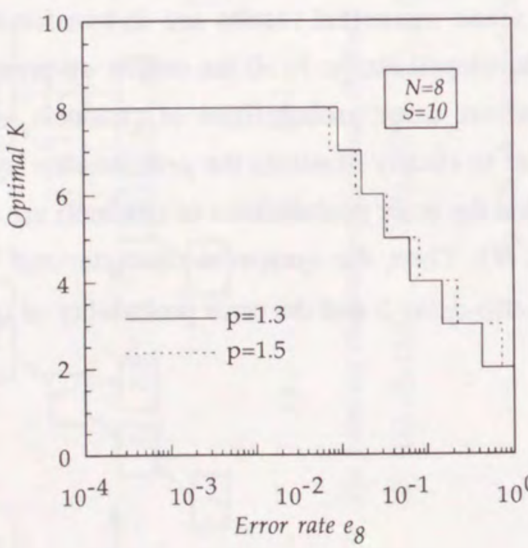


Fig. 5.7 Optimal K vs. error rate e_8

Fig. 5.6 depicts the throughput performance under *OOK* method. It shows the effects of *K* on the system throughput. When $K=N$, the *OOK* method becomes *OON* method. Based on this figure, we can see that while the error

probabilities of channels are relatively smaller, the throughput increases as *K* increases, namely, the larger *K*, the higher throughput. However, the curves reverse in the high error areas. When the error probability of channel 8, e_8, is very high, all the error probabilities of the channels become large. The reverse phenomenon means that it is not profitable to use the channels of which error probabilities are very high. Therefore, under *OOK* method, there has an optimal *K* which gets throughput to be maximum for a given error probabilities and *S*. Fig. 5.7 shows the optimal *K* when p=1.3 and 1.5.

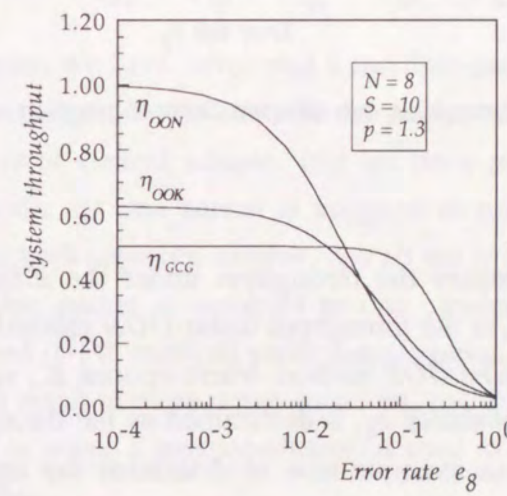


Fig. 5.8 Effects of different channel-grouping methods

Fig. 5.8 shows the comparison of throughput with different channel-grouping methods. we properly adopted K=5 for *OOK* method. For *GCG* method, the channels are grouped as {(1,2), (3,4), (5,6), (7,8)}. In this figure, the reverse phenomenon also appeared, and the throughput has been improved markedly in the higher error areas by *GCG* method. At the lower error areas, however, the throughput under *GCG* method conversely becomes much worse.

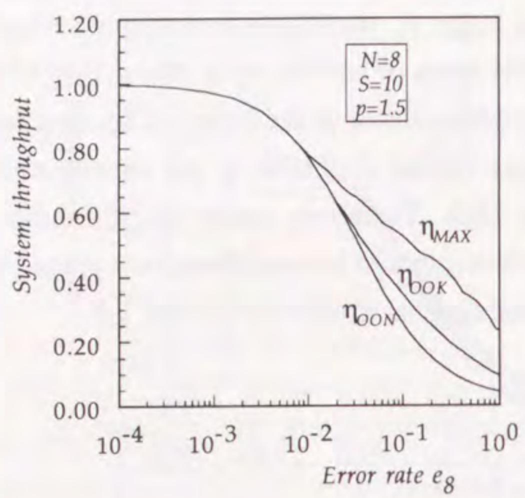


Fig. 5.9 Optimal throughput with different channel-grouping methods

In Fig. 5.9, we compare the throughput under the different channel-grouping methods. η_{OON} is the throughput under OON method. η_{OOK} shows the largest throughput under OOK method, where optimal K , varying with the conjugation of error probability e_8, is determined as the throughput gets the maximum. It gives us an example how to determine the optimal K while communication environment permits the OOK method. The η_{MAX} shows the maximal throughput by adopting the optimal channel-grouping obtained in terms of the tree-algorithm. From this figure, we see that the throughputs, η_{OON}, η_{OOK}, and η_{MAX}, are identical in the low error areas. It is true because the OON, OOK and GCG methods become as a same grouping method in lower error areas. On the contrary, in the higher error areas, the differences of throughput among the different channel-grouping methods are very significant. The throughput is extremely improved by the optimal channel-grouping method.

Here we summarize the above discussion as three points. First, the throughput is improved by adopting multiple parallel channels (Fig. 5.6). Second, there has no universal channel-grouping which can get maximal throughput in any error probabilities. Third, however, for a given error

probabilities, the optimal channel-grouping exists and can be searched by the tree-algorithm developed above.

Finally, we emphasize that the optimal channel-grouping also can be forcibly searched without our efficient tree-algorithm. But the computational time is very long. For instance, using the workstation Sun-4*LC*(12.5MIPS), when N=8, S=10, it takes about two hours to search one optimal-grouping if we forcibly search all the combinations of the N channels. However, it only takes several seconds if adopting our tree-algorithm.

5.7 Conclusions

In this chapter, we have developed a multiple-parallel-channel system to improve the throughput and to enhance the packet error tolerance under Go-Back-N ARQ error control scheme, and we have presented three channel-grouping methods: *(a)* One packet is assigned to one channel at each time based on the packet's sequence number, and all the N channels are used (OON method). *(b)* One packet is assigned to one channel but only the first K channels are used (OOK method) under consideration that usage of the worse channels would result in more retransmissions. *(c)* General-channel grouping (GCG method) in which a grouped-channel is used to transmit one packet and its copies at a time.

We have analyzed the throughput for different channel-grouping methods, and then discussed the channel-grouping methods under the measure of the throughput and given an efficient tree-algorithm to search the optimal channel-grouping.

The numerical results show that the throughput would be extremely improved, especially, when packet errors are relatively higher, and that there does not exist a universal optimal channel-grouping which can cope with any error probabilities. At a given error probabilities, however, there exists an optimal channel-grouping which can make throughput to become maximal.

The approaches developed in this paper are fascinating under the assumption that the errors of channels are observable. It can be solved by monitoring the channels to provide error probabilities to the transmitter. Based on this knowledge, the transmitter can search the optimal channel-grouping to adapt the different environments.

6 Dynamic GBN-ARQ Scheme on Two-Parallel Noisy Channels System

6.1 Introduction

In chapter 5, a multiple-parallel channels system has been proposed under GBN-ARQ error control scheme. There, the usage of the multiple channels, i.e. channel-grouping measured by the throughput, mainly has been discussed. Besides the throughput, another measurement defined as buffer occupancy (or packet delay) is also used to evaluate system performances. The buffer occupancy is usually defined as the number of packets in transmission buffer observed. Generally speaking, under the same traffic circumstances, the buffer occupancy will decrease as the throughput increases. However, the buffer occupancy under error control scheme A may be different with that under error control scheme B even if the throughput of scheme A is same as that of scheme B. This will be certified later in this chapter.

Unlike the throughput analysis, the analysis for the buffer occupancy is more difficult because it is dependent upon the packet arrival pattern as well as upon the error control scheme. In this chapter, instead of a n-parallel channels system, a two-parallel channels system will be considered because the buffer occupancy for n-parallel channels system is very difficult or even impossible. The buffer occupancy of the two-parallel channel system for both the static and dynamic packet assignment rules will be analyzed under assuming that packets arrive at the transmitter according to Poisson process.

6.2 Protocol of 2-Parallel Channels System

The system model considered in this chapter is depicted in Fig. 6.1. The transmitter is connected to the receiver over 2-parallel channels, called channel 1 and channel 2, of which the error probabilities are assumed to be e_1 and e_2, respectively. The time axis is divided into packet-sized slots and numbered by integer. Thus it needs one slot to transmit a packet. Without loss of generality, assume that $e_1 \leq e_2$, and the two-parallel channels have the same round-trip-delay, S slots. The transmitter has an infinite buffer used to temporarily store the arrival packets. The packet arrives at the transmitter according to Poisson process with arrival rate λ. Packet arrivals occur at the end of the slots. The GBN-ARQ scheme is used to handle the transmission errors. If the transmitter receives a NAK, the NAK'ed packet and the succeeding packets whose sequences are greater than that of NAK'ed packet, will be immediately retransmitted. The packets arrived at the receiver are checked for errors, and the receiver sends back the corresponding acknowledgment packet over an error-free backward channel. An acknowledgment packet is used to acknowledge all the packets received during a slot. As the packet assignment rules, the static and the dynamic packet assignment rules will be considered.

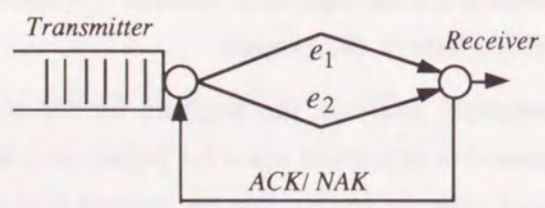


Fig. 6.1 A communication system with 2-parallel channels

6.2.1 Static packet assignment rule

The *static* packet assignment rule is to assign packets to channels based on the packet sequence number and the error rates of channels. The lower sequence packet is transmitted over higher quality channel, i.e., over channel 1.

If there is only one packet in the buffer waiting for transmission, it is transmitted over channel 1, and channel 2 is idle at this time. This operation is illustrated in Fig. 6.2(a), where packets waiting for transmission will be served based on FCFS (First-Come First-Service) basis. At the end of this slot during packet 3 arrives, only packet 3 is waiting for transmission, packet 3 is sent over channel 1, and at this time channel 2 is idle.

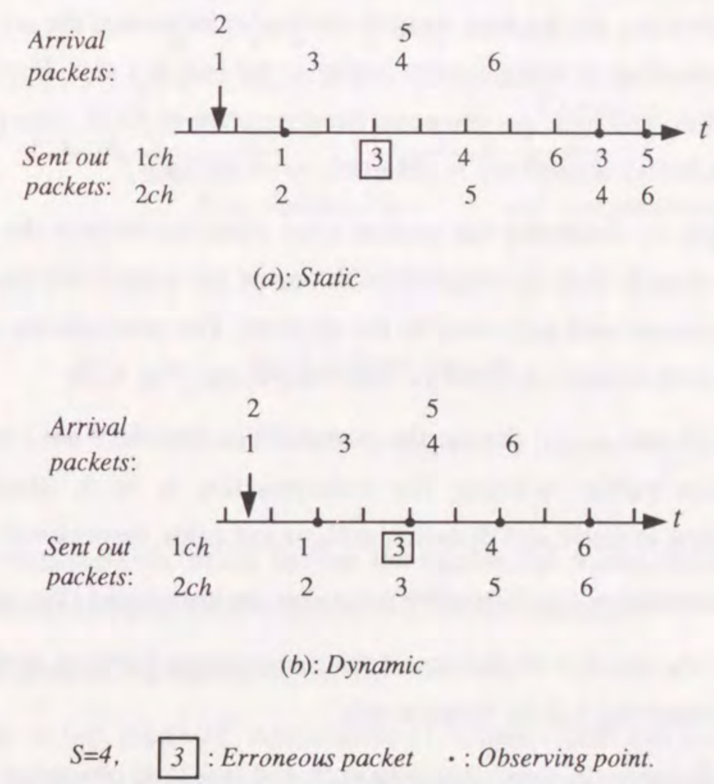


Fig. 6.2 Packet assignment rules

6.2.2 Dynamic packet assignment rule

The dynamic one is based on effectively using the idle channel when only one packet is waiting for transmission. Definitely, the transmitter transmits the packet over channel 1 and its copy over channel 2. In case that there is no packet in the buffer waiting for transmission, the transmitter does nothing, channel 1 and 2 are idle. If there are two or more packets waiting for transmission, the operation is same as that of static rule, i.e., the transmitter sends the lower sequence packet to channel 1 and the higher sequence one to channel 2. The operation is illustrated in Fig. 6.2(*b*).

6.3 Buffer Occupancy Analysis

In this section, we analyze the buffer occupancy under static and dynamic packet assignment rules. The *buffer occupancy* is defined as the number of the packets (including the packets waiting for transmission and the packets waiting for retransmission) in transmission buffer at the end of a slot. First, we set up a state equation, and then get the generating function (*g.f.*) of state probabilities. Finally, the buffer occupancy is obtained by using the *g.f.*

We begin by observing the system state. Here we observe the system state at the end of such slots in which at least one of the transmitted packets will be correctly received and delivered by the receiver. For convenience, the instant at the end of such a slot is defined as *observing point* (Fig. 6.2).

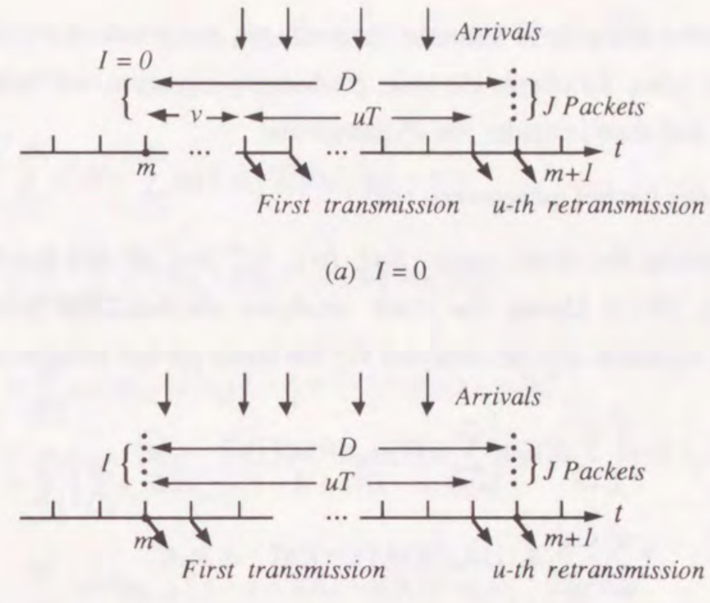
Let $p_{m,s}(i)$ and $p_{m,d}(i)$ denote the probabilities that there are i packets in the transmission buffer waiting for transmission at m-th observing point corresponding to static and dynamic assignment rules, respectively.

For convenience, the following notations are introduced (Fig. 6.3):

I and J: the number of packets in the transmission buffer at m-th and $(m+1)$-th observing points, respectively.

D: the distance (in slots) between m-th and $(m+1)$-th observing points.

v: the distance (in slots) from m-th observing point to the beginning of the next transmission . Notice that if I=0, then $v \geq 1$ (Fig. 6.3(a)), and if I>0, then v=0 (Fi.g 6.3(b)).



(*b*) $I > 0$

Fig. 6.3 State diagram

u: the retransmission times before the successful transmission, i.e., the number of retransmissions before m+1 observing point.

T=S+1. Therefore we have D=v+uT+1.

Moreover, if I=0, then $v \geq 1$. Assume v=r+1, where r indicates the number of slots during which no packets arrive at the transmitter, and "1" represents this slot in which there are some arrivals. Let the number of arrivals be l.

k: the arrivals within the next uT slots.

n: the arrivals during the last slot, i.e., during the successful transmission slot just before m+1 observing point.

y: the number of the correctly transmitted packets in last slot, i.e., y=1 or 2.

Therefore, the following relationship is clear

$$J = I + l + k + n - y. \tag{6.1}$$

The above analysis is suitable for both the static and the dynamic packet assignment rules. To obtain the state probability equation, we first consider the static rule, and then consider the dynamic one.

(A) For static packet assignment rule

Considering the three cases: I=0, I=1, I≥2 and all the conditions which satisfy Eq. (6.1). Using the state analysis method, the following state probability equation can be obtained for the static packet assignment rule;

$$
\begin{aligned}
p_{m+1,s}(j) = \Biggl[\sum_{r=0}^{\infty} a_1^r(0) \Biggl\{ & \sum_{u=0}^{\infty} a_1(1)a_{uT}(0)a_1(j)(1-e_1)e_1^u \\
& + \sum_{u=1}^{\infty}\Biggl(\sum_{k=1}^{j} a_1(1)a_{uT}(k)a_1(j-k)(1-e_1)e_2e_1^u \\
& + \sum_{k=1}^{j+1} a_1(1)a_{uT}(k)a_1(j-k+1)(1-e_1)(1-e_2)e_1^u \Biggr) \\
& + \sum_{u=0}^{\infty}\Biggl\{ \sum_{k=2}^{j+1} a_1(k)a_{1+uT}(j-k+1)(1-e_1)e_2e_1^u \\
& + \sum_{k=2}^{j+2} a_1(k)a_{1+uT}(j-k+2)(1-e_1)(1-e_2)e_1^u \Biggr\}\Biggr\}\Biggr] p_{m,s}(0) \\
& + \Biggl[\sum_{u=0}^{\infty} a_{uT}(0)a_1(j)(1-e_1)e_1^u + \sum_{u=1}^{\infty}\Biggl\{ \sum_{k=1}^{j} a_{uT}(k)a_1(j-k)(1-e_1)e_2e_1^u \\
& + \sum_{k=1}^{j+1} a_{uT}(k)a_1(j-k+1)(1-e_1)(1-e_2)e_1^u \Biggr\}\Biggr] p_{m,s}(1) \\
& + \Biggl[\sum_{u=0}^{\infty}\Biggl\{ \sum_{i=2}^{j+1} a_{1+uT}(j-i+1)(1-e_1)e_2e_1^u \\
& + \sum_{i=2}^{j+2} a_{1+uT}(j-i+2)(1-e_1)(1-e_2)e_1^u \Biggr\}\Biggr] p_{m,s}(i),
\end{aligned}
\tag{6.2}
$$

where, $a_n(k)$ denotes the probability that there are k packets arrived in n slots. Based on the Poisson arrival process, $a_n(k)$ is expressed as

$$
a_n(k) = \frac{e^{-\lambda n}(\lambda n)^k}{k!}. \tag{6.3}
$$

In equilibrium state, $m \to \infty$, $p_{m,s}(i) \to p_s(i)$, $p_{m,d}(i) \to p_d(i)$. Thus the Eq. (6.2) can be rewritten as

$$
\begin{aligned}
p_{s}(j) = \Biggl[\sum_{r=0}^{\infty} a_1^r(0) \Biggl\{ & \sum_{u=0}^{\infty} a_1(1)a_{uT}(0)a_1(j)(1-e_1)e_1^u \\
& + \sum_{u=1}^{\infty}\Biggl(\sum_{k=1}^{j} a_1(1)a_{uT}(k)a_1(j-k)(1-e_1)e_2e_1^u \\
& + \sum_{k=1}^{j+1} a_1(1)a_{uT}(k)a_1(j-k+1)(1-e_1)(1-e_2)e_1^u \Biggr) \\
& + \sum_{u=0}^{\infty}\Biggl\{ \sum_{k=2}^{j+1} a_1(k)a_{1+uT}(j-k+1)(1-e_1)e_2e_1^u \\
& + \sum_{k=2}^{j+2} a_1(k)a_{1+uT}(j-k+2)(1-e_1)(1-e_2)e_1^u \Biggr\}\Biggr\}\Biggr] p_{s}(0) \\
& + \Biggl[\sum_{u=0}^{\infty} a_{uT}(0)a_1(j)(1-e_1)e_1^u + \sum_{u=1}^{\infty}\Biggl\{ \sum_{k=1}^{j} a_{uT}(k)a_1(j-k)(1-e_1)e_2e_1^u \\
& + \sum_{k=1}^{j+1} a_{uT}(k)a_1(j-k+1)(1-e_1)(1-e_2)e_1^u \Biggr\}\Biggr] p_{s}(1) \\
& + \Biggl[\sum_{u=0}^{\infty}\Biggl\{ \sum_{i=2}^{j+1} a_{1+uT}(j-i+1)(1-e_1)e_2e_1^u \\
& + \sum_{i=2}^{j+2} a_{1+uT}(j-i+2)(1-e_1)(1-e_2)e_1^u \Biggr\}\Biggr] p_{s}(i).
\end{aligned}
\tag{6.4}
$$

(B) For dynamic packet assignment rule

For dynamic assignment rule, the difference is that when only one packet in buffer waiting for transmission, this packet is transmitted over channel 1 and its copy over channel 2 at a time. Using the same method as that for static rule, the state equation is given by

$$
\begin{aligned}
&p_d(j) = \\
&\sum_{r=0}^{\infty} a_1^r(0) \left\{ a_1(1) \sum_{u=1}^{\infty} \sum_{v=1}^{\infty} \left[\sum_{l=1}^{j} (e_1e_2)^u a_{(u-1)T}(0) a_T(l)(1-e_1)e_2 e_1^{v-1} a_{1+(v-1)T}(j-l) \right.\right. \\
&\left.\left. + \sum_{l=1}^{j+1} (e_1e_2)^u a_{(u-1)T}(0) a_T(l)(1-e_1)(1-e_2) e_1^{v-1} a_{1+(v-1)T}(j-l+1) \right] \right\} p_d(0) \\
&+ \sum_{r=0}^{\infty} a_1^r(0) \left\{ a_1(1) \sum_{u=1}^{\infty} (e_1e_2)^{u-1}(1-e_1e_2) a_{(u-1)T}(0) a_1(j) \right\} p_d(0) \\
&+ \sum_{r=0}^{\infty} a_1^r(0) \left\{ \sum_{u=0}^{\infty} \sum_{l=2}^{j+1} a_1(l)(1-e_1)e_2 e_1^u a_{1+uT}(j-l+1) \right. \\
&\left. + \sum_{u=0}^{\infty} \sum_{l=2}^{j+2} a_1(l)(1-e_1)(1-e_2) e_1^u a_{1+uT}(j-l+2) \right\} p_d(0) \\
&+ \sum_{u=1}^{\infty} \sum_{v=1}^{\infty} \left\{ \sum_{l=1}^{j} (e_1e_2)^u a_{(u-1)T}(0) a_T(l)(1-e_1)e_2 e_1^{v-1} a_{1+(v-1)T}(j-l) \right. \\
&\left. + \sum_{l=1}^{j+1} (e_1e_2)^u a_{(u-1)T}(0) a_T(l)(1-e_1)(1-e_2) e_1^{v-1} a_{1+(v-1)T}(j-l+1) \right\} p_d(1) \\
&+ \sum_{u=0}^{\infty} (e_1e_2)^u (1-e_1e_2) a_{uT}(0) a_1(j) p_d(1) \\
&+ \sum_{u=0}^{\infty} \left\{ \sum_{i=2}^{j+1} p_d(i)(1-e_1)e_2 e_1^u a_{1+uT}(j-i+1) \right. \\
&\left. + \sum_{i=2}^{j+2} p_d(i)(1-e_1)(1-e_2) e_1^u a_{1+uT}(j-i+2) \right\}.
\end{aligned}
\tag{6.4'}
$$

The generating functions $P_s(z)$ and $P_d(z)$ are defined as

$$P_s(z) = \sum_{i=0}^{\infty} p_s(i) z^i, \qquad P_d(z) = \sum_{i=0}^{\infty} p_d(i) z^i. \tag{6.5}$$

Taking the generating function about (6.4), $P_s(z)$ is given by

$$P_s(z) = \frac{\Phi_s(z)}{\Theta_s(z)}, \tag{6.6}$$

where $\Phi_s(z)$ and $\Theta_s(z)$ are given by

$$
\begin{aligned}
\Phi_s(z) = &\left\{ a_1(1) A_1(z) \frac{1-e_1}{1-e^{-\lambda T} e_1} \right. \\
&+ \left(Q_2(z) e_1 - \frac{(1-e_1) e_1 e^{-\lambda T}}{1-e^{-\lambda T} e_1} \right) a_1(1) \left((1-e_2) z^{-1} + e_2 \right) A_1(z) \\
&\left. + \left((1-e_2) z^{-2} + e_2 z^{-1} \right) \left(A_1(z) - a_1(1) z - a_1(0) \right) Q_1(z) \right\} \frac{p_s(0)}{1-a_1(0)} \\
&+ \left\{ \frac{1-e_1}{1-e^{-\lambda T} e_1} + \left((1-e_2) z^{-1} + e_2 \right) \left(Q_2(z) e_1 - \frac{(1-e_1) e_1 e^{-\lambda T}}{1-e^{-\lambda T} e_1} \right) \right\} \\
&\cdot A_1(z) p_s(1) - Q_1(z) \left(p_s(1) z + p_s(0) \right) \left((1-e_2) z^{-2} + e_2 z^{-1} \right),
\end{aligned}
\tag{6.7}
$$

$$\Theta_s(z) = 1 - \left((1-e_2) z^{-2} + e_2 z^{-1} \right) Q_1(z), \tag{6.8}$$

and, $Q_1(z)$ and $Q_2(z)$ are as follows.

$$Q_1(z) = \frac{(1-e_1) e^{-\lambda(1-z)}}{1-e^{-\lambda T(1-z)} e_1}, \tag{6.9}$$

$$Q_2(z) = \frac{(1-e_1) e^{-\lambda T(1-z)}}{1-e^{-\lambda T(1-z)} e_1}. \tag{6.10}$$

Furthermore, $A_t(z)$ denotes the generating function of $a_t(i)$. It is given by

$$A_t(z) = \sum_{i=0}^{\infty} a_t(i) z^i = e^{-\lambda t(1-z)}. \tag{6.11}$$

Taking the generating function about (6.4'), $P_d(z)$ is given by

$$P_d(z) = \frac{\Phi_d(z)}{\Theta_d(z)}, \tag{6.6'}$$

where $\Phi_d(z)$ and $\Theta_d(z)$ are given as follows;

$$\Phi_d(z) = \frac{a_1(1)}{1-a_1(0)}\left\{\frac{e_1e_2}{1-e_1e_2e^{-\lambda T}}(A_T(z)-a_T(0))Q_1(z)\left(e_2+(1-e_2)z^{-1}\right)\right.$$
$$\left.+\frac{1-e_1e_2}{1-e_1e_2e^{-\lambda T}}A_1(z)\right\}p_d(0)$$
$$+\frac{1}{1-a_1(0)}\left\{(A_1(z)-a_1(1)z-a_1(0))Q_z(z)\left(e_2z^{-1}+(1-e_2)z^{-2}\right)\right\}p_d(0)$$
$$+\frac{e_1e_2}{1-e_1e_2e^{-\lambda T}}(A_T(z)-a_T(0))Q_1(z)\left(e_2+(1-e_2)z^{-1}\right)p_d(1)$$
$$+\frac{1-e_1e_2}{1-e_1e_2e^{-\lambda T}}A_1(z)p_d(1)-\left(p_d(1)+p_d(0)\right)\left(e_2z^{-1}+(1-e_2)z^{-2}\right)Q_1(z), \tag{6.7'}$$
$$\Theta_d(z) = 1-\left((1-e_2)z^{-2}+e_2z^{-1}\right)Q_1(z). \tag{6.8'}$$

According to the properties of generating function, and using primes to denote derivatives (F' and F'' are the first and second-order derivatives of $F(z)$ with respect to z by setting z=1, respectively), the average buffer occupancy, i.e., the average number of packets in transmission buffer at the observing point in steady state, $E[L_s]$ and $E[L_d]$, can be obtained;

$$E[L_s] = \left.\frac{dP_s(z)}{dz}\right|_{z=1}, \tag{6.12}$$

$$E[L_d] = \left.\frac{dP_d(z)}{dz}\right|_{z=1}. \tag{6.12'}$$

There are two unknown $p_s(0)$ and $p_s(1)$ in expression (6.12), $p_d(0)$ and $p_d(1)$ in (6.12'). We make a set of equation to get them.

Using the boundary condition $p_s(z)\big|_{z=1}=1$, i.e., $\sum_{i=0}^{\infty}p_s(i)=1$, the following equation is obtained from (6.6).

$$\left\{\frac{(1-e_1)a_1(1)}{1-e_1e^{-\lambda T}}\cdot A_1' + Q_2'e_1a_1(1) - \left(e_1-\frac{(1-e_1)e_1e^{-\lambda T}}{1-e_1e^{-\lambda T}}\right)a_1(1)(1-e_2-A_1')\right.$$
$$\left.-\left(2(1-e_2)+e_2-Q_1'\right)\left(1-a_1(1)-a_1(0)\right)+A_1'-a_1(1)\right\}\frac{p_s(0)}{1-a_1(1)}$$
$$+\left\{Q_2'e_1-(1-e_2)\left(1-\frac{(1-e_1)e^{-\lambda T}}{1-e_1e^{-\lambda T}}\right)e_1\right\}p_s(1)+p_s(1)A_1'$$
$$+\left(2(1-e_2)+e_2\right)\left(p_s(1)+p_s(0)\right)-p_s(1)-\left(p_s(1)+p_s(0)\right)Q_1'$$
$$=2(1-e_2)+e_2-Q_1'. \tag{6.13}$$

For dynamic assignment rule, $p_d(z)\big|_{z=1}=1$ from (6.6'), we get

$$\frac{a_1(1)}{1-a_1(0)}\left[\frac{e_1e_2}{1-e_1e_2e^{-\lambda T}}\left\{A_T'+\left(1-a_T(0)\right)\left(Q_1'-(1-e_2)\right)\right\}\right.$$
$$\left.+\frac{1-e_1e_2}{1-e_1e_2e^{-\lambda T}}A_1'\right]p_d(0)+\frac{1}{1-a_1(0)}$$
$$\cdot\left[(A_1'-a_1(1))+\left\{1-a_1(1)-a_1(0)\right\}\left\{Q_1'-e_2-2(1-e_2)\right\}\right]p_d(0)$$
$$+\frac{e_1e_2}{1-e_1e_2e^{-\lambda T}}\left\{A_T'+\left(1-a_T(0)\right)\left(Q_1'-(1-e_2)\right)\right\}p_d(1)$$
$$+\frac{1-e_1e_2}{1-e_1e_2e^{-\lambda T}}A_1'p_d(1)-p_d(1)-\left(p_d(1)+p_d(0)\right)\left\{Q_1'-e_2-2(1-e_2)\right\}$$
$$=2(1-e_2)+e_2-Q_1'. \tag{6.13'}$$

To obtain $p_s(0)$, $p_s(1)$, $p_d(0)$ and $p_d(1)$, we need other equations besides equations (6.13) and (6.13'). In expressions (6.6) and (6.6'), the $P_s(z)$ and $P_d(z)$ are two generating function of $p_s(i)$ and $p_d(i)$ (i=0,1, ⋯, ∞), respectively. Thus, $P_s(z)$ and $P_d(z)$ must be analytical functions of z in the unit circle of complex plane, i.e., the z-plane [72]. Therefore, the zero points of $\Theta_s(z)$ or $\Theta_d(z)$ in the unit circle must be the zero points of $\Phi_s(z)$ or $\Phi_d(z)$. It is clear that z=1 is one zero points of $\Theta_s(z)$ or $\Theta_d(z)$. Let z_1 be another zero point of $\Theta_s(z)$ or $\Theta_d(z)$ in the unit circle, and z_1 must be a zero point of $\Phi_s(z)$ or $\Phi_d(z)$. From (6.7) and (6.8), we get the following equations.

$$1-\left((1-e_2)z_1^{-2}+e_2z_1^{-1}\right)Q_1(z_1)=0, \tag{6.14}$$

$$\begin{aligned}&\left\{a_1(1)A_1(z_1)\frac{1-e_1}{1-e^{-\lambda T}e_1}\right.\\&\quad+\left(Q_2(z_1)e_1-\frac{(1-e_1)e_1e^{-\lambda T}}{1-e^{-\lambda T}e_1}\right)a_1(1)\left((1-e_2)z_1^{-1}+e_2\right)A_1(z_1)\\&\quad\left.+\left((1-e_2)z_1^{-2}+e_2z_1^{-1}\right)\left(A_1(z_1)-a_1(1)z_1-a_1(0)\right)Q_1(z_1)\right\}\frac{p_s(0)}{1-a_1(0)}\\&+\left\{\frac{1-e_1}{1-e^{-\lambda T}e_1}+\left((1-e_2)z_1^{-1}+e_2\right)\left(Q_2(z_1)e_1-\frac{(1-e_1)e_1e^{-\lambda T}}{1-e^{-\lambda T}e_1}\right)\right\}A_1(z_1)p_s(1)\\&-Q_1(z_1)\left(p_s(1)z_1+p_s(0)\right)\left((1-e_2)z_1^{-2}+e_2z_1^{-1}\right)=0.\end{aligned} \tag{6.15}$$

For dynamic assignment rule, $\Phi_d(z_1)=0$, from (6.7') and (6.8'), we get

$$\begin{aligned}&\frac{a_1(1)}{1-a_1(0)}\left\{\frac{e_1e_2}{1-e_1e_2e^{-\lambda T}}\left(A_T(z_1)-a_T(0)\right)Q_1(z_1)\left(e_2+(1-e_2)z_1^{-1}\right)\right.\\&\qquad\left.+\frac{1-e_1e_2}{1-e_1e_2e^{-\lambda T}}A_1(z_1)\right\}p_d(0)\\&+\frac{1}{1-a_1(0)}\left\{\left(A_1(z_1)-a_1(1)z_1-a_1(0)\right)Q_2(z_1)\left(e_2z_1^{-1}+(1-e_2)z_1^{-2}\right)\right\}p_d(0)\\&+\frac{e_1e_2}{1-e_1e_2e^{-\lambda T}}\left(A_T(z_1)-a_T(0)\right)Q_1(z_1)\left(e_2+(1-e_2)z_1^{-1}\right)p_d(1)\\&+\frac{1-e_1e_2}{1-e_1e_2e^{-\lambda T}}A_1(z_1)p_d(1)-\left(p_d(1)+p_d(0)\right)\left(e_2z_1^{-1}+(1-e_2)z_1^{-2}\right)Q_1(z_1)=0.\end{aligned} \tag{6.15'}$$

Therefore, $p_s(0)$, $p_s(1)$, $p_d(0)$ and $p_d(1)$ are found by solving Eqs. (6.13), ~, (6.15), (6.13') and (6.15'). The average buffer occupancy can be obtained by (6.12) and (6.12').

6.4 Performance Evaluation

In this section, we first gives the throughput, and then discuss the performance of the buffer occupancy for static and dynamic rules by terms of the analytical results described above.

6.4.1 System throughput

To compare the static rule with dynamic one, we first give the system throughput. Generally, the system throughput represents the system's ability of supporting traffic volume. The dynamic rule can be considered as an improved one of the static rule, i.e., the dynamic rule efficiently uses the idle channel appeared in the static rule. Thus if assuming that the transmission buffer is always full, the throughput of dynamic rule is same as that of static one. From (5.7) of chapter 5, for both static and dynamic rules, the throughput, η, is easily given by

$$\eta=\frac{(1-e_1)e_2+2(1-e_1)(1-e_2)}{2(S+1)-2S(1-e_1)(1-e_2)}. \tag{6.16}$$

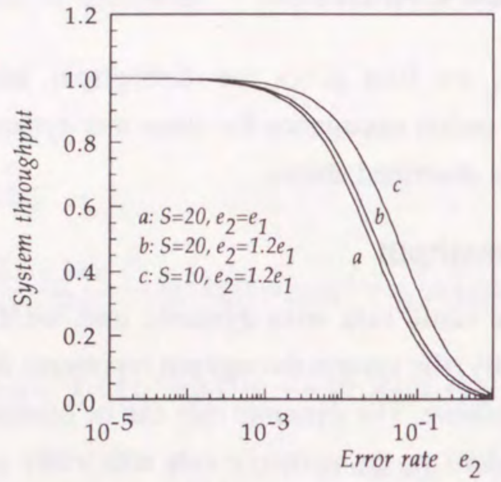


Fig. 6.4 System throughput

Note that $e_1 + e_2 \geq e_1 e_2$ due to $0 \leq e_1, e_2 \leq 1$. Based on Eq. (6.16), we can easily prove that the throughput η decreases as round-trip-delay S and/or error rate e_1 (e_2) increase, These properties are illustrated in Fig. 6.4.

6.4.2 Buffer occupancy

Fig. 6.5 shows the average buffer occupancy for both static and dynamic rules as a function of the packet arrival rate. Here we assumed that the round-trip-delay S to be 20. According to this figure, the average buffer occupancy increases as packet arrival rate λ increases, and the buffer occupancy performance of dynamic rule is better than that of the static one. Fig. 6.6 gives the comparison of the static and the dynamic rules, where $E[L_d]/E[L_s]$ denotes the proportion of the average buffer occupancy under the dynamic rule to that under the static rule. From the two figures, it is clear that the buffer occupancy is remarkably improved by the dynamic rule (surely the packet delay, which is defined as the time from the arrival of a packet at the transmitter to the correct acceptance at the receiver, also is improved remarkably). The $E[L_d]/E[L_s]$ approaches to 1 as arrival rate λ becomes very high. It is true because, when λ

is very high, channel 2 is seldom idle, the dynamic rule is equivalent to the static one.

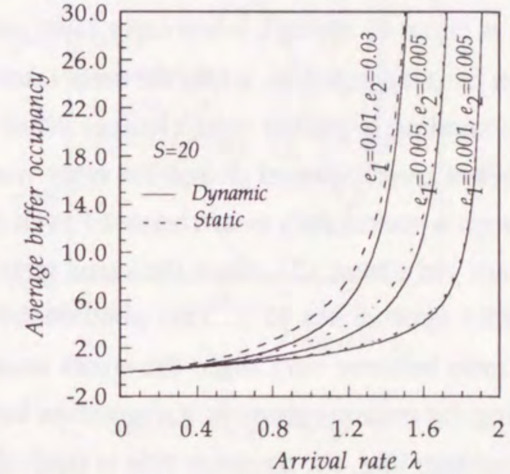


Fig. 6.5 Average buffer occupancy

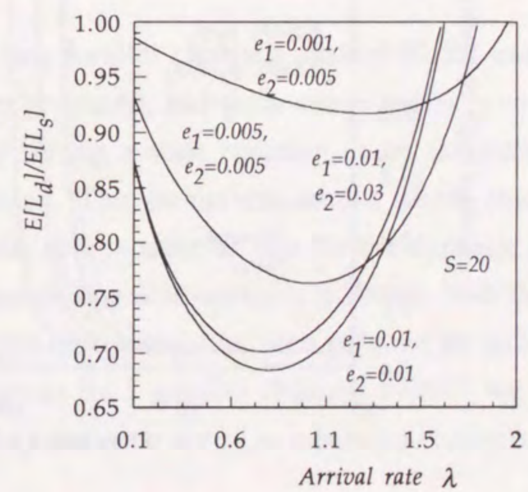


Fig. 6.6 $E[L_d]/E[L_s]$

Fig. 6.7 shows the average buffer occupancy as a function of error rate e_2. Clearly, the buffer occupancy increases as the ronud-trip-delay and/or error rate e_2 increase. Fig. 6.8 shows the improvement of the buffer occupancy performance by using the dynamic rule. From the two figures, it is clear that: (1) $E[L_d]/E[L_s]$ is equal to about 1 when error rates are very low ($<10^{-4}$). This phenomenon can be interpreted as, while the error rates are very low, the errors rarely occur, so sending a packet over channel 1 has the same efficiency as sending this packet over channel 1 and its copy over channel 2, i.e., it is efficient to transmit a packet only over channel 1 even if channel 2 is idle when the error rates are very low; (2) when the error rates are very high ($>10^{-2}$), $E[L_d]/E[L_s]$ rapidly approaches to 1. This phenomenon can be interpreted as while the error rates become very high, the errors occur frequently, the length of packets waiting for transmissions in transmission buffer becomes very long, so channel 2 is seldom idle, the dynamic rule is equivalent to the static one; (3) the buffer occupancy is improved by the dynamic rule when the error rates are not so low and not so high.

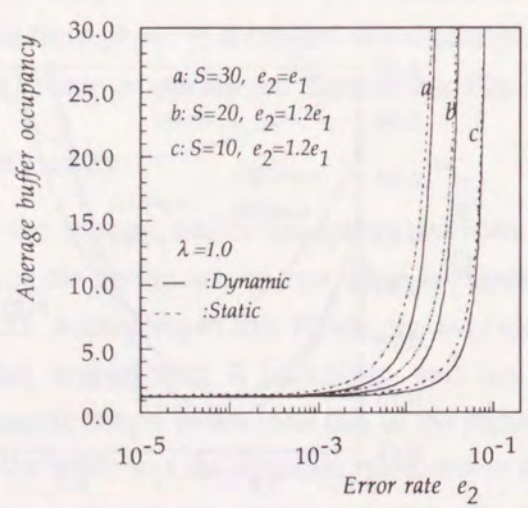


Fig. 6.7 Average buffer occupancy

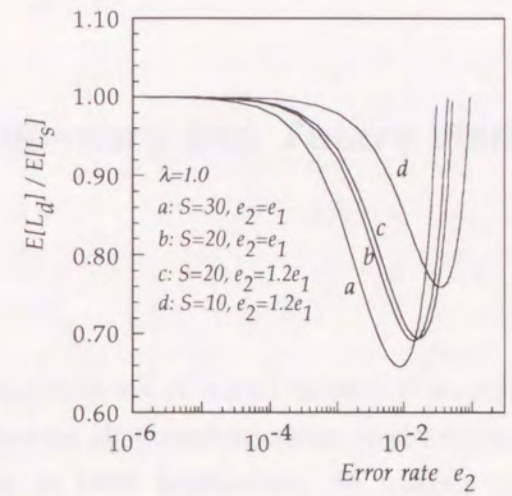


Fig. 6.8 $E[L_d] / E[L_s]$

6.5 Conclusions

In this chapter, a two-parallel channels system model under Go-Back-N ARQ scheme has been proposed, and the average buffer occupancy has been analyzed by means of setting a state equation under the static and dynamic packet assignment rules. The numerical results show that although the throughput for the static rule is same as that for the dynamic rule, the buffer occupancy for the dynamic one is remarkably different from that for the static one. The average buffer occupancy has been reduced by using the dynamic rule. Based on our analysis for 2-parallel channels system, we believe that the analysis method and the results are useful to n-parallel channels system ($n>2$).

7 Summary and Future Works

In communication systems, it would be ideal if an uniform control scheme could be applied across all communication environments. However, this is extremely difficult or even impossible. So, taking into account various communication environments, in this thesis, a priority control for multiclass traffic and some novel ARQ error control schemes (i.e., block SR-ARQ scheme in high-speed communication environments, ARQ schemes in dialogue communication and GBN-ARQ scheme in multiple-parallel channels system) which had not been settled so far, have been studied. By analyzing the system performances, the following results and new knowledges have been obtained.

(1) In multiclass traffic system, the influence of the burstiness of the arrivals from the priority source is very strong on the system performances such as the cell delay and the discard probability. Chapter 2 shows that the cell discard probability for the priority sources (bursty sources) and the cell delay for the non-priority sources increase as the traffic density increases. Especially, the discard probability changes with the bursty duration although the traffic density is constant; and the cell delay of the non-priority is not affected by the variation of the bursty duration if the traffic density is constant.

(2) In Chapter 4, a block SR-ARQ error control scheme has proposed to reduce the defective influences of protocol processing overhead time which occurs in high-speed communication environments. The results shows that the performances (the system throughput and packet delay) of our block SR-ARQ are much better than that of basic SR-ARQ.

(3) In Chapter 3, the influences of the switching and thinking times on the throughput in dialogue communication system over half-duplex transmission

line have clarified. It has been shown that there exists an optimal k which enables to maximize the throughput, Also, the optimal k has been obtained.

(4) In chapters 5 and 6, the throughput of a parallel-channel system has been analyzed, the channel-grouping method has been discussed, and a tree-algorithm used to seek out the optimal channel-grouping has been proposed. The results shows that by using the optimal channel-grouping, the system throughput is much better than that by using the existing channel-grouping(i.e., *OON* method). Furthermore, in chapter 6, a dynamic packet assignment rule has been proposed to efficiently use the idle channels in static packet assignment rule. Numerical results show that the different packet assignment rules would strongly influence the buffer occupancy even if their throughputs are mutually equivalent.

In this thesis, we have analyzed a priority control discipline and implemented the ARQ schemes to some communication systems. The common assumption in ARQ schemes which have been considered in this thesis, is that the packet errors are mutually independent. This is suitable in many cases because the errors are caused by many unpredictable reasons [87]. Accurately modeling for errors in transmission channel is very difficult. Considering the correlation among errors, some related error models have been presented [87], such as Gilbert's model [69], McCullough's model. Modeling for errors in high-speed transmission channel should be future works. In the system considered in this thesis, there are no others node (or multiplexer) between the transmitter and the receiver. The error control scheme in such systems is called *link-by-link* scheme in other literatures [68, 86]. If the transmitter and the receiver are not adjacent, i.e., the transmitter does not directly connect to the receiver, but connected each other through other nodes. The error control scheme in such systems is called *end-to-end* scheme. Recently, the high-speed communication networks, such as ATM, have been receiving great attention for transmitting various traffic. The end-to-end error and traffic controls should be the next interesting works.

References

[1] H. ichikawa, M. Aoki and T. Uchiyama, "High-Speed Packet Switching Systems for Multimedia Communications," *IEEE Select. Areas Commun.*, Vol. SAC-5, pp. 1336-1345, Oct. 1987.

[2] M. Murata and H. Miyahara, "Recent developments on Traffic Theories for Communication Networks," *IEICE* Vol. 74, No. 2, pp.142-145, Feb. 1991.

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

[4] H. Saito, "Teletraffic Studies in ATM Network," *IEICE* Vol. 75, No. 5, pp. 496-502, May 1992.

[5] T. Saito, "Broadband ISDN," IEICE Vol. 75, No. 11, pp. 1193-1199, Nov. 1992.

[6] P. Ye and S. Richard, "Telecommunications in China," *IEEE Commun. Mag.*, Vol. 31, No. 7, pp. 14-16, Jul. 1993.

[7] N. Miyaho et al., "ATM Switching System Technologies," *IEEE Denshi Tokyo*, No. 31, pp. 29-37, 1992.

[8] T. Oouchi and A. Hatono, "ATM Cell Transfer Controls for B-ISDN," *IEEE Denshi Tokyo*, No. 31, pp. 38-43, 1992.

[9] S. Shioda and H. Uose, "Virtual Path Bandwidth Control Method for ATM Networks," *IEICE Trans.*, Vol. J 75-B-1, No. 5, pp. 333-342, May, 1992.

[10] C. Yuan and J. A. Silverster, "Queueing Analysis of Delay Constrained Voice in a Packet Switching System," *IEEE J. Select. Areas Commun.*, Vol. 7, No. 5, pp. 729-738, June 1989.

[11] Y. Lim and J. Kobza, "Analysis of a Delay-Dependent Priority Discipline in an Integrated Multiclass Traffic Fast Packet," *IEEE Trans. Commun.*, Vol. 38, No. 5, pp. 659-665, May 1990.

[12] J. N. Daigle and J. D. Langford, "Models for Analysis of Packet Voice Communication Systems," *IEEE J. Select. Areas Commun.*, Vol. SAC-4, pp. 847-855, Sept. 1986.

[13] I. Ide, "Superposition of Interrupted Poisson Processes and Its Application to Packetized Voice Multiplexers," in *Proc. 12th Int. Teletraffic Cong.*, Torino, Italy, 1988.

[14] T. M. Chen et al., "Dynamic Priority Protocols for Packet Voice," *IEEE J. Selec. Areas Commun.*, Vol. 7, No. 5, pp. 632-643, June 1989.

[15] A. Y. Lin and J. A. Silvester, "Priority Queueing Strategies and Buffer Allocation Protocols for Traffic Control at an ATM Integrated Broadband Switching System," *IEEE J. Selec. Areas Commun.*, Vol. 9, No. 9, pp. 1524-1536, Dec. 1991.

[16] H. Saito and K. Shiomoto, "Dynamic Call Admission Control in ATM Networks," *IEEE J. Selec. Areas Commun.*, Vol. 9, No. 7, pp. 982-989, Sept. 1991.

[17] M. Murata et al., "Analysis of a Discrete-Time Single-Server Queue with Bursty Inputs for Traffic Control in ATM Networks," *IEEE J. Selec. Areas Commun.*, Vol. 8, No. 3, pp. 447-458, Apr. 1990.

[18] H. Yoshino, "An Approximation Method for Queuing Networks with Priority and Its Applications to ATM Networks," *IEICE Technical Report*, IN 88-66.

[19] H. Arai et al., "An Analysis of Call Admission Control in the ATM Network," *IEICE Trans.* Vol. J 72-B-1, No. 11, pp. 1000-1007, Nov. 1989.

[20] S. Q. Li, "Study of Information Loss in Packet Voice Systems," *IEEE Trans. Commun.*, Vol. 37, No. 11, pp. 1192-1202, Nov. 1989.

[21] Y. H. Ohba and M. Y. Murata, " An Analysis of Interdeparture Process for Bursty Traffic in ATM Networks," *IEICE Technical Report*, SSE 89-169, pp. 37-42, 1990.

[22] H. Ohta and T. Kitami, "A Technique to Detect and Compensate Consecutive Cell Loss in ATM Networks," *IEEE Proc. Infocom'91*, pp. 781-790.

[23] Y. A. Tanaka and M. Akiyama, "A Priority Control Method in ATM Switching System," *IEICE Technical Report*, IN 90-36, pp. 31-36, 1990.

[24] S. I. Kuribayashi and Y. T. Takahashi, "Priority Queues with Synchronous Service and Its Application to Packet Switching System," *IEICE Trans.*, Vol. J70-B, No. 10, pp. 1118-1123, Oct. 1987.

[25] K. J. Kubota and M. Y. Murata, "Congestion Control for Bursty Video Traffic in ATM Networks," *IEICE Technical Report*, SSE 89-168, pp. 31-36, 1990.

[26] M. E. Anagnostou and E. N. Portonotarious, "Performance Analysis of the Selective Repeat ARQ Protocol," *IEEE Trans. Commun.*, Vol. COM-34, No.2, pp.127-135, Feb. 1986.

[27] S. R. Chandran and S. Lin, "Selective-Repeat-ARQ Schemes for Broadcast Links," *IEEE Trans. Commun.*, Vol. 40, No. 1, pp.12-19, Jan. 1992.

[28] R. Lau and S. Lei, "Forward Error Correction for BISDN: Implementation and Analysis," in *Proc. IEEE Globecom'91*, pp.1638-1643, 1991.

[29] A. G. Wicker, "Adaptive Rate Error Control Through the Use of Diversity Combining and Majority-Logic Decoding in a Hybird-ARQ protocol," *IEEE Trans. Commun.*, Vol. 39, No. 3, pp. 380-385, Mar. 1991.

[30] Y. Ishibashi and A. Iwabuchi, "Performance Evaluation of Adaptive ARQ Schemes over Half Duplex Transmission Line," in *Proc. IEEE Infocom'89*, pp.564-573, 1989.

[31] T. Umemoto and Y. Ohno, "Data communication Networks," CQ Company, Chap. 7, 1990.

[32] S. Lin, S. J. Costello and M. J. Miller, "Automatic-Repeat-Request Error Control Scheme," *IEEE Commun. Magazine*, Vol. 22, No. 12, pp. 5-17, Dec. 1984.

[33] D. Towsley and J. K. Wolf, "On the Statistical Analysis of Queue Lengths and Waiting Times for Statistical Multiplexers with ARQ Retransmission Schemes," *IEEE Trans. Commun.*, Vol. COM-27, NO. 4, pp. 693-702, April 1979.

[34] M. Moeneclaey and H. Bruneel, "Throughput Optimization for a generalized Stop-and-Wait ARQ Scheme," *IEEE Trans. Commun.*, Vol. COM-34, No. 2, pp. 206-207, Feb. 1986.

[35] Z. Rosberg and N. Shacham, "Resequencing Delay and Buffer Occupancy under the Selective Repeat ARQ," *IEEE Trans. Inform. Theory*, Vol. IT-35, pp. 166-173, Jan. 1989.

[36] M. Komatsu, "Resequencing Delay in the Selective-Repeat ARQ Scheme," *Trans. IEICE* Vol. E73-6, pp. 867-869, June 1990.

[37] D. Towsley, "The Stutter Go Back-N ARQ Protocol," *IEEE Trans. Commun.*, Vol. COM-27, No. 6, pp. 869-875, Jun. 1979.

[38] H. Bruneel and M. Moeneclaey, "On throughput Performance of Some Continuous ARQ Strategies with Repeated Transmissions", *IEEE Trans. Commun.*, Vol. COM-34, No. 3, pp. 244-249 [Mar. 1979].

[39] C. Chen, M. Komatsu and K. Kinoshita, "Performance Evaluation of ARQ Schemes in Dialogue Communication over Half-Duplex Line," *Proc. IEICE SSE'92*, Vol. 92, No. 349, pp. 1-6, 1992.

[40] C. Chen, M. Komatsu and K. Kinoshita, "Performance Evaluation of Block SR-ARQ Scheme in High-Speed Communication Environments," *IEICE Trans. Commun.*, Vol. E75-B-12, No. 12, pp. 1338-1345, Dec. 1992.

[41] N. Schacham and B. C. Shin, "A Selective-Repeat-ARQ Protocol for Parallel Channels and Its Resequencing Analysis," *IEEE Trans. Commun.*, Vol. 40, No. 4, pp. 773-782, April 1992.

[42] C. H. Leung, Y. Kikumoto and A. Sorensen, "The Throughput Efficiency of the Go-Back-N ARQ Scheme under Markov and Related Error Structures," *IEEE Trans. Commun.*, Vol. 36, No. 2, pp. 231-234, Feb. 1988.

[43] A. V. Aho, J. E. Hopcroft and J. D. Ullman, "*Data Structures and Algorithms*," Addison-Wesley Publishing Company, Chap. 3 and 7, 1987.

[44] J. F. Chang and T. H. Yang, "Multichannel ARQ Protocols," *IEEE Trans. Commun.*, Vol. 41, No. 4, pp. 592-598, Apr. 1993.

[45] D. Towsley, " An Analysis of a Point-to-Multipoint Channel Using a Go-Back-N Error Control Protocol," *IEEE Trans. Commun.*, Vol. COM.33, pp.282-285, Mar. 1985.

[46] I. S. Gopal and J. M. Jaffe, "Point-to-Multipoint Communication over Broadcast," *IEEE Trans. Commun.*, Com.32, No. 9, pp. 1034-1044, Sept. 1984.

[47] M. J. Miller and S. Lin, "The Analysis of Some Selective-Repeat ARQ Schemes with Finite Receiver Buffer," *IEEE Trans. Commun.*, Com-29, No. 9, pp.1307-1315, Sept. 1981.

[48] P. S. Yu and S. Lin, "An Efficient Selective-Repeat ARQ Scheme for Satellite Channels and Throughput Analysis," *IEEE Trans. Commun.*, Com-29, No. 3, pp. 353-363, Mar. 1981.

[49] M. Kusano, J. Taniguchi and T. Takahashi, "A Study of Multiple-Channel Communication Scheme," *IEICE Technical Report*, In92-111, pp. 19-24, Feb. 1993.

[50] Y. Hayashida, S. Fujii and Y. Tezuka, "Delay performances of a Continuous ARQ System with Copy-Transmissions," *IEICE Trans.* Vol. J 70-B, No. 10, pp. 1111-1117, Oct. 1987.

[51] S. Fujii, Y. Hayashida and Y. Tezuka, "Delay performances of Some ARQ Systems with Service Interruptions," *IEICE Trans.* Vol. J 71-B, No. 4, pp. 516-522, Apr. 1988.

[52] Y. Hayashida, N. Sugimachi, M. Komatsu and Y. Yoshida, "Delay Performance of Go-Back-N ARQ Protocol with Limited Retransmissions," *IEICE Trans.* Vol. J 71-B, No.12, pp. 1486-1491, Dec. 1988.

[53] Y. Hayashida, N. Sugimachi and Y. Yoshida, "Throughput Analysis of Tandem Go-back-N ARQ Scheme," *IEICE Trans.* Vol. J 72-B-1, No.3, pp. 219-225, Mar. 1989.

[54] Y. Hayashida, N. Sugimachi and Y. Yoshida, "Delay Performances of Tandem Go-back-N ARQ Scheme," *IEICE Trans.* Vol. J 72-B-1, No.6, pp. 486-491, June 1989.

[55] M. Nakamura and Y. Takada, "ARQ Scheme Reinforced with Past Acknowledgement Signals," *IEICE Trans.* Vol. J 72-B-1, No.1, pp. 51-57, Jan. 1989.

[56] H. Inai et al., "Effect of Timeout Retransmissions on End-to-End Delay in Packet Switching Networks," *IEICE Trans.* Vol. J 74-B-1, No. 4, pp. 275-282, Apr. 1991.

[57] M. B. Pursley and S. D. Sandberg, "Variable-Rate Hybrid ARQ for Meteor-Burst Communications," *IEEE Trans. Commun.*, Vol. 40, No. 1, pp. 60-73, Jan. 1992.

[58] S. R. Kim and C. K. Un, "Throughput Analysis for Two ARQ Schemes Using Combined Transition Matrix," *IEEE Trans. Commun.*, Vol. 40, No. 11, pp. 1679-1683, Nov. 1992.

[59] T. H. Lee, "The Throughput Efficiency of Go-Back-N ARQ Scheme for Burst-Error Channels," *IEEE Proc. Infocom'91*, pp. 773-780.

[60] G. R. Pieris and G. H. Sasaki, "The Performance of Simple Error Control Protocols under Correlated Packet Losses," *IEEE Proc. Infocom'91*, pp. 764-772.

[61] S. Kallel and C. Leung, "An Adaptive Incremental Redundancy Selective-Rpeat ARQ Scheme for Finite Buffer Receivers," *IEEE Proc. Infocom'91*, pp. 791-796.

[62] M. C. Lin and M. Y. Guu, "The Performance Analysis of a Concatenated ARQ Scheme Using Parity Retransmissions," *IEEE Trans. Commun.*, Vol. 39, No. 12, pp. 1869-1874, Dec. 1991.

[63] A. Shiozaki et al., "A Hybrid ARQ Scheme with Adaptive Forward Error Correction for Satellite Communications," *IEEE Trans. Commun.*, Vol. 39, No. 4, pp. 482-484, Apr. 1991.

[64] S. Kallel and D. Haccoun, "Sequential Decoding with an Efficient Partial Retransmission ARQ Strategy," *IEEE Trans. Commun.*, Vol. 39, No. 2, pp. 208-213, Feb. 1991.

[65] R. Fantacci, "Performance Evaluation of Some ARQ Schemes Using Efficient Modulation Techniques and Noncoherent Detection," *IEEE Trans. Commun.*, Vol. 39, No. 3, pp. 445-451, Mar. 1991.

[66] Y. J. Cho and C. K. Un, "Window Flow Control with Error-Checking Scheme in Quasi-Cut-Through Switching Network with Noisy Channels," *IEEE Trans. Commun.*, Vol. 39, No. 3, pp. 394-397, Mar. 1991.

[67] N. Shacham and D. Towsley, "Resequencing Delay and Buffer Occupancy in Selective Repeat ARQ with Multiple Receivers," *IEEE Trans. Commun.*, Vol. 39, No. 6, pp. 928-937, June 1991.

[68] J. J. Bae et al., "Evaluation of the Effects of Protocol Processing Overhead in Error Recovery Schemes for a High-Speed Packet Switched Network: Link-by-Link versus Edge-to-Edge Schemes," *IEEE J.Selec. Areas Commun.*, Vol. 9, No. 9, pp. 1496-1509, Dec. 1991.

[69] M. Komatsu, C.-X. Chen, and K. Kinoshita, "Traffic Analysis of the Stop-and-Wait ARQ over a Markov Error Channel," Submitted to *IEICE Trans. Commun.*

[70] T. Takenaka et al., "Characteristics and Approximation Methods of a Non-Preemptive $\Sigma M_i/G/1[\Sigma N_i]$ System," *IEICE Trans.* Vol. J 70-B, No. 11, pp. 1338-1345, Nov. 1987.

[71] G. Genelli, "A New Method for the Integration of Modulation and Channel Coding in an ARQ Protocol," *IEEE Trans. Commun.*, Vol. 40, No. 10, pp. 1594-1606, Oct. 1992.

[72] H. Bruneel and B. G. Kim, *Discrete-Time Models for Communication Systems Including ATM*, by Kluwer Academic Publishers, Boston, 1993.

[73] V. K. Oduol and S. D. Morgera, "Performance Evaluation of the Generalized Type-II Hybrid ARQ Scheme with Noisy Feedback on Markov Channels," *IEEE Trans. Commun.*, Vol. 41, No. 1, pp. 32-40, Jan. 1993.

[74] G. R. Pieris and G. H. Sasaki, "Performance of Go-Back-∞ Protocol under Correlated Packet Losses," *IEEE Trans. Commun.*, Vol. 41, No. 5, pp. 660-663, May 1993.

[75] B. T. Doshi et al., "Error and Flow Control Performance of a High Speed Protocol," *IEEE Trans. Commun.*, Vol. 41, No. 5, pp. 707-720, May 1993.

[76] D.-L. Lu and J. -F. Chang, "Performance of ARQ Protocols in Nonindependent Channel Errors," *IEEE Trans. Commun.*, Vol. 41, No. 5, pp. 721-730, May 1993.

[77] H. Ahmadi, P.Kermani and P. Tran-gia, "Throughput Analysis of a Class of Selective Repeat Protocols in High-Speed Environments," in *Proc. IEEE* GLOBECOM'89, pp.26.4.1-26.4.9.

[78] D. D. Clark, et al., "NETBLT: a High Throughput Transport Protocol," in *Proc.* ACM SIGCOMM'87, pp. 353-359.

[79] W. Zwaenepoel, "Protocol for Large Data Transfers over Local Networks," in *Proc.* ACM SIGCOMM'85, pp. 22-32.

[80] K. Sabnani and A. Netravali, "A High Speed Transport Protocol for Datagram/Virtual Circuit Networks," in *Proc.* ACM SIGCOMM'89. pp.146-157.

[81] G. M. Brown et al., "Block Acknowledgement: Redesigning the Window Protocol," in *Proc.* ACM SIGCOMM'89. pp.128-135.

[82] H. Heffes and D. M. Lucantoni, "A Markov Modulated Characterization of Packetized Voice and Data Traffic and Related Statistical Multiplexer Performances," *IEEE J. Select. Areas Commun.*, Vol. SAC-4, pp. 856-878, Sept. 1986.

[83] R. Chipalkatti et al., "Scheduling Policies for Real-Time and Nonreal-Time Traffic in a Statistical Multiplexer," in *Proc. IEEE INFOCOM'89*, pp.774-783.

[84] P. Bhattacharya and A. Ephremides, "Optimal Scheduling with Strict Deadlines," *IEEE Trans. Automat. Contr.*, Vol. 34, pp. 721-728, July 1989.

[85] S. Panwar et al., "Optimal Scheduling Policies for a class of Queues with Customer Deadlines to the Beginning of Service," J. Ass. Comput. Mach., Vol. 35, No. 4, pp. 832-844, Oct. 1988.

[86] A. Bhargava et al., "Performance Comparison of Error Control Schemes in High-Speed Computer Communication Networks," *IEEE J.Selec. Areas Commun.*, Vol. 6, No. 9, pp. 1565-1575, Dec. 1988.

[87] L. Kanal and A. R. Sastry, "Models for Channels with Memory and Their Applications to Error Control," *IEEE J.Selec. Areas Commun.*, Vol. 6, No. 9, pp. 1565-1575, Dec. 1988.

[88] W. Turin and M. M. Sondhi, "Modeling Error Soures in Digital Channels," *IEEE J.Selec. Areas Commun.*, Vol. 11, No. 3, pp. 340-347, Dec. 1993.

[89] A. Gersht and K. J. Lee, "A congestion Control Framework for ATM Networks," in *Proc. IEEE INFOCOM'89*, pp. 701-710.

[90] G. M. Woodruff et al., "A Congestion Control Framework for High-Speed Intergrated Packetized Transport," in *Proc. IEEE GLOBECOM'88*, pp. 7.1.1-7.1.5.

List of Author's Publications

Papers

1. C.-X. Chen and M. Komatsu, M. Kawakatsu and M. Kawaguchi,
"Performance Analysis of Multiclass Traffic in ATM Networks," *IEEE Proc. GLOBECOM'91*, pp.0244-0248, Dec. 1991 (Chapter 2).

2. C.-X. Chen, M. Komatsu and K. Kinoshita,
"Performance Evaluation of Block SR-ARQ Scheme in High-Speed Communication Environments," *IEICE Trans. Commun.*, Vol. E-75-B, No. 12, pp. 1338-1345, Dec. 1992 (Chapter 4).

3. C.-X. Chen, M. Komatsu and K. Kinoshita,
"Channel-Grouping Methods on Go-Back-N ARQ Scheme in Multiple-Parallel-Channel System," *IEICE Trans. Commun.*(accepted for publication) (Chapter 5).

4. C.-X. Chen, M. Komatsu and K. Kinoshita,
"Throughput Analysis of ARQ Schemes in Dialogue Communication over Half-Duplex Line," *IEICE Trans. Commun.* (accepted for publication) (Chapter 3).

5. M. Komatsu, C.-X. Chen, and K. Kinoshita,
"Traffic Analysis of the Stop-and-Wait ARQ over a Markov Error Channel," *IEICE Trans. Commun.* (accepted for publication).

Technical Reports

6. C.-X. Chen, M. Komatsu, M. Kawakatsu and M. Kawaguchi,
"Performance Analysis of Multiclass Traffic in ATM Networks," *SSE* Technical Report of *IEICE*, Mar. 1991.

7. C.-X. Chen, M. Komatsu and K. Kinoshita,
"Performance Analysis of Block SR-ARQ Scheme," Technical Report of *IEICE*, *SSE92-17*, May 1992 (in Japanese).

☆ *IEEE: The Institute of Electrical and Electronics Engineers, U. S. A.*

† *IEICE: The Institute of Electronics, Information and Communication Engineers, Japan.*

8. C.-X. Chen, M. Komatsu and K. Kinoshita,
"Performance Analysis of Block SR-ARQ Scheme for High-Speed Packet Networks," *27-th FTC* Technical Report, Jul. 1992.

9. C.-X. Chen, M. Komatsu and K. Kinoshita,
"Performance Evaluation of ARQ Schemes in Dialogue Communication over Half-Duplex Line," Technical Report of *IEICE, SSE92-93,* Nov. 1992.

10. C.-X. Chen, M. Komatsu and K. Kinoshita,
"Throughput Analysis of Go-Back-N ARQ Scheme over the Multiple-Parallel-Channel System," Technical Report of *IEICE, SSE92-122,* Dec. 1992 (in Japanese).