PACKET SWITCHING
IN COMPUTER COMMUNICATION NETWORKS

By

HIROMI OKADA

DECEMBER 1974

PACKET SWITCHING
IN COMPUTER COMMUNICATION NETWORKS

By

HIROMI OKADA

DECEMBER 1974

ACKNOWLEDGEMENTS

The Auther would like to express his appreciation to Professor Yoshikazu Tezuka for his continuing encouragement and guidance, for his useful discussion, and for his numerous suggestions, throughout this investigation.

The auther is pleased to acknowledge the patient guidances of Honorary Professor Yoshiroo Kasahara, Honorary Professor Kenji Aoyagi, and Honorary Professor Kanemasa Katoo.

The auther also would like to take this opportunity to express his thanks to Professor Kioyasu Itakura, Professor Toshihiko Namekawa, Professor Nobuaki Kumagai, Professor Yoshiroo Nakanishi, Professor Osamu Kakushio, Professor Zenichi Kitamura, Professor Hiroshi Ozaki, and Professor Shinzoo Kodama, for their patient guidances and instructions.

The auther wishes to thank Associate Professor Masao Kasahara for his helpful suggestions and observations. The auther also thanks Professor Toshiharu Hasegawa of Kyoto University and Associate Professor Minoru Akiyama of Tokyo University for their constructive advices in the development of this investigation

The auther is indebted to Dr. Hidehiko Sanada Dr. Hikaru Nakanishi, Dr. Shundo Lyou, Dr. Seiichi Uchinami, Dr. Hideo Miyahara, Dr. Tutomu Asabe, and Dr. Junichiroo Shiomi for their useful suggestions and discussions. Thanks are also due to many colleagues with whom the auther had and/or has discussed on this investigation, among them, Mr. Tadashi Hirao, Mr. Kooji Hookawa, Mr. Norio Hamada, and Mr. Yasuo Iwashita for their useful suggestions, Mr. Teiji Nishizawa, Mr. Yukio Yamasaki, and Mr.Masakazu Suzuki for their contributions in the development of some of the analyses, Mr. Michio Naito and

Mr. Tatsuo Sunouchi for their patient assistances.

Finally, the auther is indebted to the member of Toyo Information System Ltd., especially President Osamu Tanaka, Director Yoshito Miyata, Mr. Noboru Yamada, and Mr. Seiji Maenishi for their helpful advices and assistances in the computer simulation of this research.

ABSTRACT

This thesis discusses a packet switching technique which can be applied to computer-communication networks. Present study has two main purposes. One is to develop mathematical tools for evaluating and optimizing the system performance. The other is to exploit useful techniques for packet switching.

In Chapter 1, fundamental aspects of packet-switching computer-communication networks are presented. A review of the previous researches and the problems studied in this thesis are given.

In Chapter 2, several packeting methods for dividing messages into standardized packets are described. First, four types of packeting methods (i.e. Single, Sequential, Exponential, and Complex packeting methods) are proposed and mathematically described. Second, packet transmission efficiency η_T is defined as the ratio of the message bits to transmitted bits. Finally, using this η_T as the evaluation measure, the packeting methods are analyzed and compared.

In Chapter 3, delivering behavior of a packet-sequence of the same message in the network is represented by a model and analyzed. Packet-intervening action and dropping action are defined. Packet-intervening number is also defined and is chosen for representing the state of the system. Considering the Markovian property of this packet-intervening number, probability density function of packet-intervening number are calculated, both in transient state and in steady state.

In Chapter 4, the prediction of average message traverse time is described. The message traverse time consists of nodal delay at each node on the path of the message in the network and reassemble delay at the destination node. The former part of this chapter (i.e.

Sec.4.2) deals with the nodal delay. Introducing the Independence Assumption, the communication network can be regarded as a network of queues. Further, each node is composed of a tandem of two types of queues (i.e. queue with switching service of the CPU and output queues with channel transmission) Therefore, by solving these queueing models, nodal delay in the network is obtained Sec.4.3 describes the analysis of reassemble delay Given the probability density function of intervening number reassemble delay is easily obtained. Therefore, the message traverse time is obtained by summing up these elementary delays.

In Chapter 5, the acknowledgement (ACK) control techniques for packet retransmission are described. In packet switching, selective repeat ARQ is generally used for keeping packet from missing. The chapter deals with the packet-ACK methods on this ARQ. Various packet-ACK methods are classified, and two types of multi-ACK control techniques are proposed. As performance measures, packet response time and transmission efficiency can be used. The various ACK methods are analyzed and compared regarding these measures. As a result of this analysis, it is insisted that the multi-ACK method with time-control and piggy-back fashion is the best ACK method in the computer-communication networks.

In Chapter 6, Isarithmic flow control method is studied. This method prevents the network from over-congestion by forcing a finite number of packet carriers in the whole network. Permitting process of external packet in Isarithmic network is represented by a model of duplicated queue with permit and external packet. Analyzing this queueing model, the effectiveness of flow control and excess control effect are disclosed. In order to enhence the throughput of Isarithmic method, some improvements are introduced and their advantages are shown.

In Chapter 7, the block switching method is presented. This method is a hybrid switching which has both line-switching mode and packet-switching mode. Basic principle of the method and control behavior are described. Then, comparing three types of switching by throughput, superiority of this method is shown.

In Chapter 8, the conclusions obtained in this dissertation are summarized.

CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

With the extensive advance of computer science and communication technology, the more surpassing infomation processing systems with high accuracy, effectiveness, rapidity and potential ability have been pursued. The computer networks have been the most important objectives in the last decade.

For the purpose of resource sharing, load sharing, and huge potentiality, not a few computer networks, that range from miniature size of the inhouse-network to the gigantic international-network, have been realizing. But, most of them are constructed as the experimental computer network. Thus, the development of the computer network must be still at the first step. In the pursuit of higher performance with more potential ability, it is quite necessary to investigate the problems of the analysis, design and implementation of the computer network.

The computer network is composed of three abilities from aspects of the basic function. Those are:

1) Data processing (or computational) ability
2) Data storing ability
3) Data communicating ability

The first and the second are the essential functions of the computer. On connecting to the remote-located computers, the third ability is indispensable for the computer network. Further, it is noted that the third function is made more powerful by utilizing the computer itself. Thus, the computer network is not only the network of locally distributed computers, but also is the systematical combination of these three functions.

This dissertation is primarily concerned with the third function of the computer network, i.e. data communicating ability, and discusses on a packet switching technique applicable to the computer network. This technique enables rapid delivering of the message, efficient sharing of the transmission channel, and efficient utilization of the network capacity. Concerning to this packet switching, the major objectives of this thesis are described as follows.

1) To develop the mathematical tools for evaluating and optimizing the system performance
2) To exploit useful techniques for packet switching

1.1 Aspects of Computer Network

There are various types of the computer system which are called as computer network. Thus, it is appropriate to define the computer network. G.D.Cole[1] defines the computer network as follows.

" A computer network is a set of interconnected processors which can be utilized jointly in a procedure manner, but which normally are controlled by separate operating systems, and can perform in an autonomous manner."

With this definition of the computer network, we consider some network systems and experiments.

There are several interesting systems in the earliest attempts to connect a number of computers. Those are the SAGE[2] air defence systems, the SABRE[3] reservation system for the American Airlines and so on. Then, the reliable data transmission systems with computers are required. The examples of these systems include the AUTODIN[4], the CYBERNET of CDC Corp., the DATRAN etc.

A star network has several advantages regarding to simplicity and cost. The examples of this type are the COINS, the Lawrence Radiation Laboratory OCTPUS[5], and the IBM computer network[6]. A loop network is another type of simple and economical network.

An example of loop-network is the DCS of Univ. of California.

The examples of the computer networks which are utilizing the packet switching technique are the NPL network[7,8,9] and the ARPANET [10,11,12,13]. The former is the experimental computer network in National Physical Laboratory in England. The latter is one of the largest and the most ambitious networks in the various systems. This network is composed of a large number of node computers which are called as interface message processor(IMP). The large computer (called as HOST) is connected to each IMP. These IMPs are linked together by up to 200 Kbits/sec transmission channel.

There are numerous other existing networks and the networks in the design stages. In this section, we describe only the typical example of computer networks.

1.2 Packet-Switching Computer Network

For the computer network, three switching methods are considered. Those are:

1) Line switching
2) Message switching
3) Packet switching

The first method is sometimes called as circuit switching or direct switching. This method is the same as the dial-up system. Thus, a network user must establish the path (or link) of his message, before he put it into the network.

On the other hand, a network user need not establish this link, in the second and the third method. In the message switching, a message is transfered in the network on Store-and-Forward (S/F) way, with header informations. In the packet switching, a message is divided into several standardized packets, and these packets are transfered in the network on S/F way.

Since users of computer network use large computers in the fashion of the time-sharing, remote-access, and on-line, we can assume that the traffic of the computer network has the following characteristics.

1) Messages have the short average length

2) Messages occur frequently

For these characteristics of the messages, the packet switching can be considered to be more advantageous than other two methods.

We represent the model of the packet-switching computer-network as Fig.1.1. This network is composed of two subnetworks. Those are:

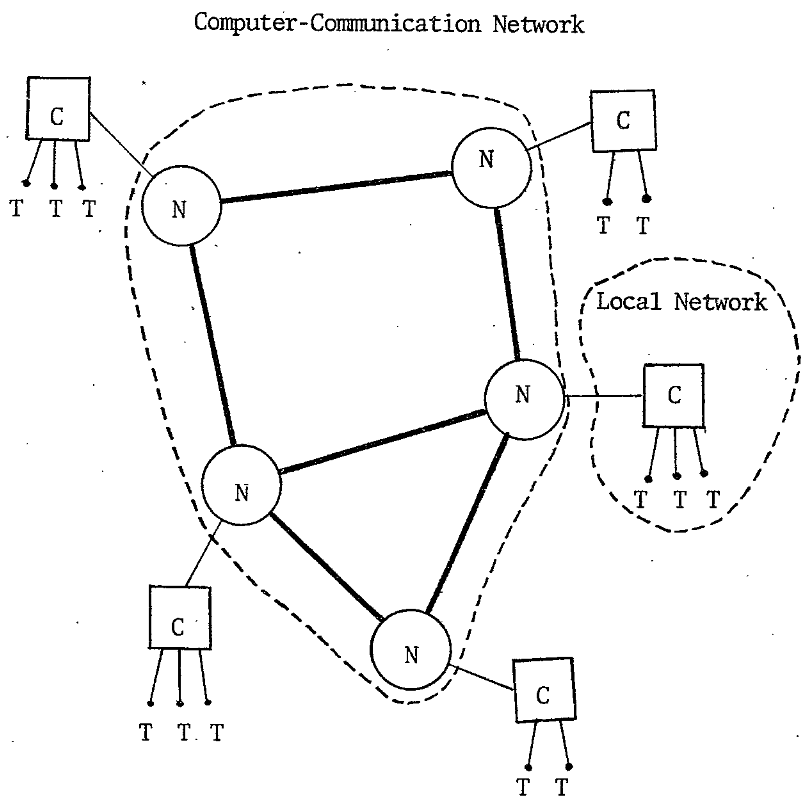


Fig. 1.1 Computer network

1) High level network: The network of node computers (N) connected by large capacity data channels. We define this as *computer-communication network*.

2) Low level network: The network of large computers (C) and terminals (T) connected by small capacity channels as tree-shape. We define this as local communication network.

This thesis deals with the former, i.e. computer-communication network, and we regard the latter as the source and the sink of messages.

We will describe the basic concept of packet switching in the following. When a terminal user (or a large computer) intends to transfer a message to another terminal through the network, he sends the message to switching node which his terminal belongs to. If over-congestions occur in the network, the node rejects this message by the flow control function and the terminal must halt this message till the over-congestions are dissolved. Otherwise, the node can receive this message and can transfer to the destination through the network. We call the former as the source node and the latter, the destination node.

At the source node, the message is first divided into several packets. Then each packet of the same message is independently transfered through the network to the destination node on the Store-and-Forward way. This means that each packet chooses its own path according to the routing procedure and is transmitted to the next node as a transmission unit. To obtain high reliability, the adjacent nodes in the network exchange the acknowledgement at each transmission of a packet. When the packet is correctly received, then it is transmitted to the next node.

When a packet reaches its destination node, it is stored in the reassemble buffer till all the packets of the same message arrive.

Then the destination node reassembles these packets to the original message and transfers this message to the destination terminal. Further, the destination sends the acknowledgement to the source node that the message is completely received, and requests for the next message.

1.3 Design Problem on Packet-Switching Computer-Communication Networks

In the design and operation of a packet-switching computer-communication networks, there are various problems to be investigated. They are classified into three classes. Those are:

A) Design of network structure

B) Design of switching node

C) Decision of switching system

The first class consists of the following.

A-1) Design of network topology

A-2) Channel capacity assignment

A-3) Resource allocation

The second class consists of the following.

B-1) Design of the CPU ability of the switching node (processing speed)

B-2) Design of buffer memory size

B-3) Decision of the functions of the switching node

The third class contains the following problems.

C-1) Design of packet structure (packeting method)

C-2) Flow-control method

C-3) Routing procedure

C-4) Acknowledgement (ACK) method for packet retransmission due to channel error

C-5) Priority discipline of packet

For these design problems, the performance measures to be established are:

D-1) Total communication cost

D-2) Throughput

D-3) System efficiency

D-4) Message traverse time (message delay)

D-5) Reliability

These three classes of design problems and the measures are related systematically each other. But, it is quite difficult to design the optimal packet-switching computer-communication network, concerning to all these. Thus, when we want to model, analyze, estimate, and design the packet switching network, it is indispensable for us to establish the problems to be solved.

1.4 Review of the Previous Researches

During the last decade, there have been the numerous investigations in the fields of computer-communication network and packet-switching or Store-and-Forward network. In this section, the brief review of the previous researches on these fields are presented.

Concerning to the Store-and-Forward networks(S/F network), the earliest and the most basic contributions were given by L.Kleinrock [14] and P.Baran[15]. Kleinrock represented the S/F network as a network of queues and introduced the results of P.J.Burk[16] and J.R.Jackson[17] to this queueing network. Further, Kleinrock derived the Independence Assumption and secured the validity of this assumption by computer simulation. By introducing this assumption, the queueing network can be regarded as a network of independent queues. Baran dealt with the distributed communication network for high reliability and presented the basic concept of the packet switching technique.

We will summarize some of the representative contributions in the area of each problem classified in Sec.1.3. First, concerning to the network structure problems, Kleinrock[14] analyzed capacity assignment, and H.Frank, et al[18] considered the network topology and capacity assignment. On the resource allocation, W.W.Chu[19], P.P.S.Chen[20] investigated the optimization problem of file allocation.

Second, in regard to the switching node problems, analysis of buffering technique and design of buffer size are the most interesting and important problems. Chu[21,22,23] gave the many and important contributions, utilizing the queueing theory. J.H.Chang[24] studied the buffer of teleprocessing system and J.F.Zeigler[25] analyzed the nodal blocking.

The analysis of message traverse time (message delay) is one of the most important problems in the S/F network, and is quite related to the buffer behaviors of the nodal processing and the channel transmission. Kleinrock[14] gave the basic analysis of message delay in the message switching network. G.L.Fultz[26] analyzed the message delay in the packet switching, utilizing the results of Kleinrock. He dealt with the priority of packets regarding to the packet length.

Third, there are many technical problems concerning to the decision of the switching system. The design of packet structure (including the design of block length and format) was investigated by F.B.Wood[27], Chu[28]. They dealt with the optimization of fixed length of signle block. R.L.Kirlin[29] studied the type of variable length block.

The routing procedure is one of the most interesting and important problems and there are many contributions in this area. R.T. Prosser investigated the random routing[30] and directory routing[31].

B.W.Bohehm[32] and Fultz[33] examined some adaptive routing and obtained the interesting results. Flow control is the important technique to prevent an S/F network from over-congestion as well as routing procedure. R.E.Kahn[34] showed the method in the ARPA-NET. D.W.Davies[35] presented the Isarithmic method. According to this Isarithmic network, W.L.Price[36,37] analyzed the behavior by computer simulation and showed the interesting results.

The retransmission of data network was investigated by R.J. Henice[38,39]. A.G.Gatfield[50] studied the ARQ technique on the satellite channel. The packet switching with this satellite channel is the interesting technique. Kleinrock[40], L.G.Roberts[41] investigated the access methods in the satellite packet switching.

1.5 Research Problems

Most of the contributions shown in Sec.1.4 dealt with the message switching network and there exist only a few works concerning to the packet switching. The results of the message switching (i.e. model, analysis, measures, and technology) may be approximately applied to the packet switching, but can not cover all the problems of the packet switching with sufficient accuracy.

In this thesis, we investigate some of these problems and attack to analyze the problems which are unsolved and/or unexploited. We summarize them as follow.

1) Packeting methods: This means the transforming algorithm from a message to several packets. We present four types of packeting methods and represent mathematical models of them. Defining the packet transmission efficiency and packet error rate as performance measures, we analyze four packeting methods.
2) Analysis of delivering behavior of packet-sequence of the same message: Under the fixed routing, all packets of the same

message pass the same route. These packet-sequences are intervened by many packets of other messages on the traversing network. We model and analyze these processes.

3) The prediction of message traverse time: Regarding to this problem, there exist several contributions. We discuss on this problem from the aspects of two unique points. One is the general representation of packet composition (packet format). The other is that reassemble delay is derived from the analysis of packet intervening process.

4) Packet-ACK method: On packet retransmission, there exist few contributions that dealt with acknowledgement (ACK) method. We classify these ACK method and define two measures. Further, we present two multi-ACK control methods. Regarding to two measures, we analyze these ACK methods.

5) The analysis of the Isarithmic flow control method: Price investigated this problem by computer simulation, but there exists no theoretical analysis. We represent one switching node in the Isarithmic network as the model of duplicated queueing system and analyze it. Furthermore, we present some improvements and show their advantages.

6) The block switching method: Concerning to the various and wide characteristics of the traffic in the computer-communication network, the packet switching has some disadvatages. To improve these, we present the block switching method. This method is one of the hybrid switchings and based upon the packet switching. We describe the concept of this method and show the comparison of the throughputs of three switching methods.

CHAPTER 2

PACKETING METHODS

2.1 Introduction

In packet switching network, the messages are divided into standardized packets at the source node. We call this process *packeting*. For packeting, basically, two methods are considered. Those are:

1) Fixed length packeting

2) Variable length packeting

In the first case, only a few types of packets are permitted in the network. Since the number of types of packets is small, the switching process of packet is very easy in the switching node. But, on composing packet, dummy bits (null bits) are added to make the packet length constant.

In the second case, only the maximum length of packet is determined. So each packet has a different length according to the original message, and does not need any dummy bits. When a packet is stored in the buffer-memory of switching node, it is settled in the buffer block of fixed size which is equal to the maximum length of packets. Further, in this case, switching process is more complicated than the case of fixed length packeting. Thus, this thesis discusses on the fixed length packeting.

In packeting a message, each packet is composed of a certain number of segments (or blocks). A segment has a fixed length. It is a transmission unit and is equal to the size of the buffer-memory unit. The number of segments in the packet is determined by the packeting method. The first segment of the packet has the header which contains housekeeping bits. Each segment has check bits for channel error detection. In most case, the last segment has dummy bits to make the segment length constant. The examples of packet are shown in Fig.2.1.

This chapter presents four types of packeting methods and also represents them mathematically. To evaluate these methods, transmission efficiency η_T is defined. Through packet error rate, expected number of retransmission, and η_T, packeting methods are analyzed and compared.

To analyze these, the following assumptions are introduced.

2.1) The message length has an exponential distribution with mean m.

2.2) Channel error during transmission is independent of each other and bit error rate is defined as q.

2.3) On receiving a packet, any channel error can be detected, and when channel error occurs, the same packet is retransmitted.

2.2 Packeting Methods and Mathematical Models

Packeting method is the algorithm to compose a certain number of segments into packet and can be mathematically represented by the following four items.

1) The set of packets defined as P, whose elements are the number of segments that a packet can have

2) $P_{pm}(i)$: Probability that i packets occur when a message is packeted

3) $P_{sp}(i)$: Probability that a packet contains i segments

4) $E_{sm}(i)$: Expectation that i segment-packet occurs per one message

From the definition, $P_{sp}(i)$ is denoted by $E_{sm}(i)$ as follows.

$$P_{sp}(i) = \frac{E_{sm}(i)}{\sum_{j \varepsilon P} E_{sm}(j)} \qquad (2.1)$$

On the following four packeting methods, P, $P_{pm}(i)$, and $E_{sm}(i)$ are derived and represented.

2.2.1 Single Packeting

Each packet is composed of one segment-packet. The set of packet is

$$P = \{ 1 \} \tag{2.2}$$

The probability $P_{pm}(i)$ and the expectation $E_{sm}(i)$ are

$$P_{pm}(i) = F\{im(L_1-L_2-L_3)\}-F\{(i-1)m(L_1-L_2-L_3)\} \tag{2.3}$$

$$E_{sm}(i) \begin{cases} = \sum_{j=1}^{\infty} jP_{pm}(j) & i=1 \\ = 0 & i \geq 2 \end{cases} \tag{2.4}$$

An example of this packeting is shown in Fig.2.1 (a).

2.2.2 Sequential Packeting

The messages are packeted into one-segment-packet to N-segment-packet according to their lengths. If the message length x is so long and cannot be included in N-segment-packet, i.e.

$$jm(NL_1-L_2-NL_3)+m[(i-1)L_1-L_2+\delta(i-1)L_2-(i-1)L_3]<x$$

$$\leq jm(NL_1-L_2-NL_3)+m(iL_1-L_2-iL_3) \tag{2.5}$$

where (j=1,2,....: i=1,2,....,N-1)

$$\delta(i-1) = 1 \qquad i=1$$

$$= 0 \qquad i \neq 1$$

the message is divided into j packets of N-segment-packet and one i-segment-packet. The set of packets P, $P_{pm}(i)$, and $E_{sm}(i)$ are

$$P = \{1,2,3,.....,N\} \tag{2.6}$$

$$P_{pm}(i) = F\{im(NL_1-L_2-NL_3)\}-F\{(i-1)m(NL_1-L_2-NL_3)\} \tag{2.7}$$

$$E_{sm}(i)\begin{cases} =\sum_{j=0}^{\infty}[F\{jm(NL_1-L_2-NL_3)+m(iL_1-L_2-iL_3)\}-F\{jm(NL_1-L_2-NL_3) \\ \quad +m(<i-1>L_1-L_2-<i-1>L_3)+\delta(i-1)mL_2\}] \quad i\varepsilon P, \quad i\neq N \\ =\sum_{j=0}^{\infty}[F\{(j+1)m(NL_1-L_2-NL_3)\}-F\{jm(NL_1-L_2-NL_3) \\ \quad +m(<N-1>L_1-L_2-<N-1>L_3)\}+jP_{pm}(i+1)] \quad i=N \end{cases} \qquad (2.8)$$

An example of this packeting method is shown in Fig.2.1 (b).

2.2.3 Exponential Packeting

The types of packets of this method are 2^i-segment-packet (i=1, 2,..,N), then the set of packets P is

$$P = \{1,2,4,.......,2^N\} \qquad (2.9)$$

According to the length of message, packets are composed in order to minimize the non-information bits, i.e. dummy bits which occur in packeting, and housekeeping bits. Considering a phase with the cycle of the length $\Delta m=m(2^NL_1-L_2-2^NL_3)$ and the states $(1,2,3,..,2^N)$ in each phase, the range where the message with length x becomes j phase k state is

$$X(j,k-1) < x \leq X(j,k) \qquad (2.10)$$

where

$$X(j,k) = jm(2^NL_1-L_2-2^NL_3)+m(kL_1-w<k>L_2-kL_3)$$

$$(j = 0,1,2,...... \quad ; k = 1,2,4,.....,2^N)$$

$$X(j,0) = X(j-1,2^N)$$

$$w(k) = \sum_{i=0}^{N} a_{ki}$$

$$k = a_{k0}+a_{k1}\cdot 2+a_{k2}\cdot 2^2+.........+a_{kN}\cdot 2^N \qquad (a_{ki} \text{ is 0 or 1})$$

The message which belongs to j phase k state is packeted to $(j+a_{kN})$ packets of 2^N-segment-packet and a_{ki} packet of 2^i-segment-packet (i=1,2,.,N-1). The probability $P_{Pm}(i)$ and the expectation $E_{sm}(i)$ are given by

$$P_{pm}(i) = \sum_{j=0}^{i} \sum_{k=1}^{2^N} \delta\{i-j-w(k)\}\cdot[F\{X(j,k)\}-F\{X(j,k-1)\}] \qquad (2.11)$$

$$E_{sm}(i) = \sum_{j=0}^{i} \sum_{k=1}^{2^N} a_{ki}\cdot[F\{X(j,k)\}-F\{X(j,k-1)\}] \qquad i\epsilon P,\ i\neq 2^N$$

$$= \sum_{j=0}^{\infty} [F\{X(j,2^N)\}-F\{X(j,2^N-1)\}+j\{F(X<j+1,0>)$$

$$-F(X<j,0>)\}] \qquad i=2^N \qquad (2.12)$$

An example of this packeting is shown in Fig.2.1 (c).

2.2.4 Complex Packeting

The set of packets P of this method is

$$P = \{1,2,3,\ldots,2^M-1,2^M,2^{M+1},2^{M+2},\ldots,2^N\} \qquad (2.13)$$

As with Exponential packeting, considering the phase with the cycle of the length $\Delta m=m(2^N L_1-L_2-2^N L_3)$ and the states $(1,2,\ldots,2^N)$ in each phase. The range where the message with length x becomes j phase k state is

$$X(j,k-1)<x\leq X(j,k) \qquad (2.14)$$

where ($j=1,2,\ldots;\ k=1,2,\ldots,2^N$)

$$X(j,k) = jm(2^N L_1-L_2-2^N L_3)+mkL_1-\{w(k)+1+\delta(A<k>)\}mL_2-mkL_3$$

$$w(k) = \sum_{j=M}^{N} a_{kj} \qquad A(k)\equiv k \qquad (\text{mod } 2^M)$$

$$k = a_{k0}+a_{k1}\cdot 2+a_{k2}\cdot 2^2+\cdots\cdots+a_{kN}\cdot 2^N$$

The message which belongs to j phase k state is divided into $(j+a_{kN})$ packets of 2^N-segment-packet, $\delta[A(k)-i]$ packet of i-segment-packet $(i=1,2,\ldots,2^M-1)$ and a_{ki} packet of 2^i-segment-packet $(i=M,M+1,\ldots,N-1)$. The probability $P_{pm}(i)$ and the expectation $E_{sm}(i)$ are represented as follows.

$$P_{pm}(i) = \sum_{j=0}^{\infty} \sum_{k=1}^{2^N} \delta\{k-j-w(k)-1+\delta(A<k>-2^N)\} \cdot [F\{X(j,k)\} - F\{X(j,k-1)\}] \quad (2.15)$$

$$E_{sm}(i) = \begin{cases} \sum_{j=0}^{\infty} \sum_{k=1}^{2^N} \delta\{A(k)-i\} \cdot [F\{X(j,k)\}-F\{X(j,k-1)\}] & 1 \leqq i \leqq 2^M-1 \\ \sum_{j=0}^{\infty} \sum_{k=1}^{2^N} a_{ki} \cdot [F\{X(j,k)\}-F\{X(j,k-1)\}] & i \varepsilon P,\ 2^M \leqq i \leqq 2^{N-1} \\ \sum_{j=0}^{\infty} [F\{X(j,2^N)\}-F\{X(j,2^{N-1})\}+j\{F(X<j+1,0>) - F(X<j,0>)\}] & i=2^N \end{cases} \quad (2.16)$$

An example of this packeting method is shown in Fig.2.1 (d).

2.3 Packet Retransmission due to Channel Error

On transmitting a packet to the adjacent node, the probability α_1 that one segment is transmitted without any error is

$$\alpha_1 = (1-q)^{mL_1} \quad (2.17)$$

where q is a bit error rate.

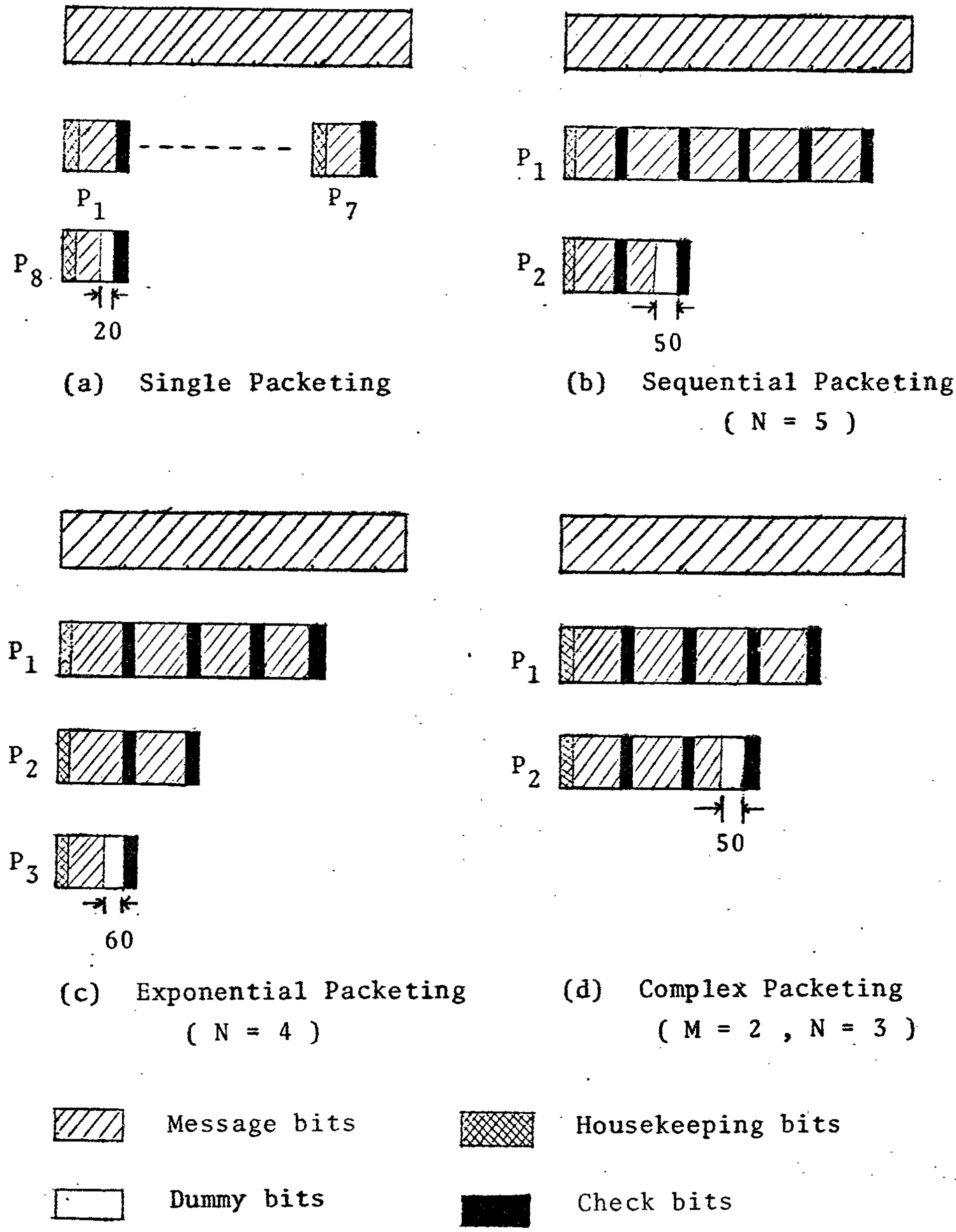
Similarly, the probability α_i that i-segment-packet is transmitted without any error is

$$\alpha_i = (1-q)^{imL_1} \quad (2.18)$$

So, the probability ε_i that some channel errors occur on transmitting i-segment-packet is given by

$$\varepsilon_i = 1-\alpha_i = 1-(1-q)^{imL_1} \quad (2.19)$$

Consequently, the probability ε, defined as packet error rate, that



$m = 500$ $L_1 = 0.2$ $L_2 = 0.04$ $L_3 = 0.02$ $x = 540$

Fig. 2.1 Example of packeting methods

the packet is erroneously transmitted becomes

$$\varepsilon = \sum_{i\varepsilon P} P_{sp}(i)\cdot\varepsilon_i = 1-\sum_{i\varepsilon P} P_{sp}(i)\cdot\alpha_i \tag{2.20}$$

Then, utilizing these probability, the expected number of retransmission can be obtained. According to the assumption, any channel errors during transmission can be detected at the receiving node and a retransmission is required to the sending node. The expected number of the retransmission of packet is derived as follows.

Using ε_i and α_i, the mean number of transmission per i-segment-packet, γ_i is

$$\gamma_i = \alpha_i + 2\cdot\varepsilon_i\cdot\alpha_i + 3\cdot\varepsilon_i^2\cdot\alpha_i + \cdots\cdots\cdots + n\cdot\varepsilon_i^{n-1}\cdot\alpha_i + \cdots\cdots$$

$$= \frac{\alpha_i}{(1-\varepsilon_i)^2} = \frac{1}{1-\varepsilon_i} \tag{2.21}$$

Then, on every packeting method, the mean number of transmission per one packet, γ is

$$\gamma = \sum_{i\varepsilon P} P_{sp}(i)\cdot\gamma_i = \sum_{i\varepsilon P} P_{sp}(i)\cdot\frac{1}{1-\varepsilon_i} \tag{2.22}$$

Hence, the expected number of retransmission per one packet, ω is

$$\omega = \gamma - 1 = \sum_{i\varepsilon P} P_{sp}(i)\cdot\frac{\varepsilon_i}{1-\varepsilon_i} \tag{2.23}$$

2.4 Transmission Efficiency

On determining the packeting method, or further on designing the channel capacity of packet-switching computer networks, the efficiency of transmission is one of the most essential evaluation measures. In order to measure this quantitatively, the transmission efficiency η_T is defined as the ratio of the message bits to the all transmitted bits required to transmit one message to the adjacent node perfectly.

Generally, all transmitted bits between any pair of node are composed of the following factors.

1) Message bits (information bits)
2) Non-information bits consisted of the housekeeping bits, check bits, and dummy bits
3) Retransmitted packet bits due to a channel error
4) Packet acknowledgement bits (between neighbouring two nodes)
5) Message acknowledgement bits (between the source node and the destination node)
6) Control information bits for channel control, communication-buffer control and so on
7) Route information bits

The first and the second indicate the bits that compose packets, and the third is the retransmission packet. The fourth is the acknowledgement information that the receiving node (N_R) transmits to the sending node (N_T) for informing the receipt-state. Classifying roughly, there are three methods about the packet acknowledgement.†

i) Positive acknowledgement (P-ACK) ; Only when N_R receives packet correctly, it transmits ACK to N_T.

ii) Negative acknowledgement (N-ACK) : Only when N_R receives packet faultly, it transmits ACK to N_T.

iii) All acknowledgement (A-ACK) : Whenever N_R receives packet, it informs to N_T about receipt-state.

The fifth concerns with the acknowledgement that the destination node (N_D) transmits to the source node (N_S), when N_D receives all packets of the same message. If the route of packets changes according to

---

† In Chapter 5, the packet-ACK methods are discussed in detail.

time, it is quite difficult to take (5) into consideration on packet transmission efficiency precisely. So, if acting time is long enough, it may be assumed that the message-ACK relates with its message almost one to one. About (6) and (7), characteristics of transmitted bits are very fluctuated according to their methods. Otherwise, they may be processed by subchannel. Hence, we do not take (6) and (7) into consideration.

Above all, considering (1)-(5), the transmission efficiency η_T is represented as follows.

$$\eta_T = \frac{m}{L_1 \cdot m \cdot S_T} \tag{2.24}$$

where S_T is the sum of segments consisted of packets, retransmitted packets, packet-ACK, and message-ACK and is derived as follows.

$$S_T = L_4^* + \sum_{i \varepsilon P} E_{sm}(i) \cdot \{i\gamma_i + \beta L_5^*\} \tag{2.25}$$

Here, L_4^*, L_5^* are the rate of message-ACK and packet-ACK to one segment length. And β is classified according to ACK methods as follows.

$$\beta \begin{cases} = 1 & \text{where P-ACK} \\ = \gamma_i - 1 & \text{where N-ACK} \\ = \gamma_i & \text{where A-ACK} \end{cases}$$

Consequently, η_T is

$$\eta_T = \frac{1}{L_1 \cdot [L_4^* + \sum_{i \varepsilon P} E_{sm}(i) \cdot \{i \cdot \gamma_i + \beta \cdot L_5^*\}]} \tag{2.26}$$

2.5 Calculated Results and Considerations

On the four packeting methods, packet error rate ε, the expected number of retransmission ω and the transmission efficiency η_T are calculated. These results are shown and discussed below. On calculating, the average message length m is chosen as 10^3.

Fig.2.2 represents q-ε characteristics. When the bit error rate q is in the range from 10^{-7} to 10^{-4}, ε_i is small enough, so the difference between ω and ε is

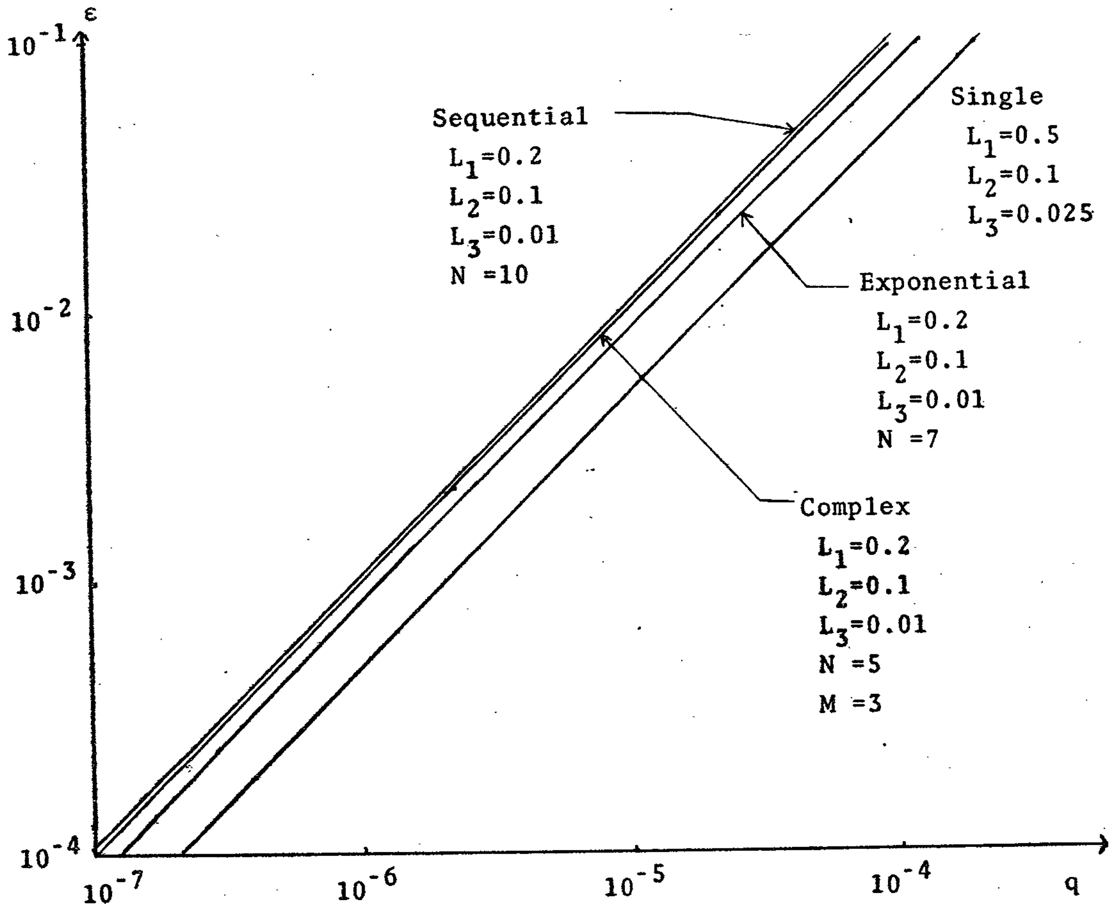


Fig. 2.2 Packet error rate versus bit error rate (m=1000)

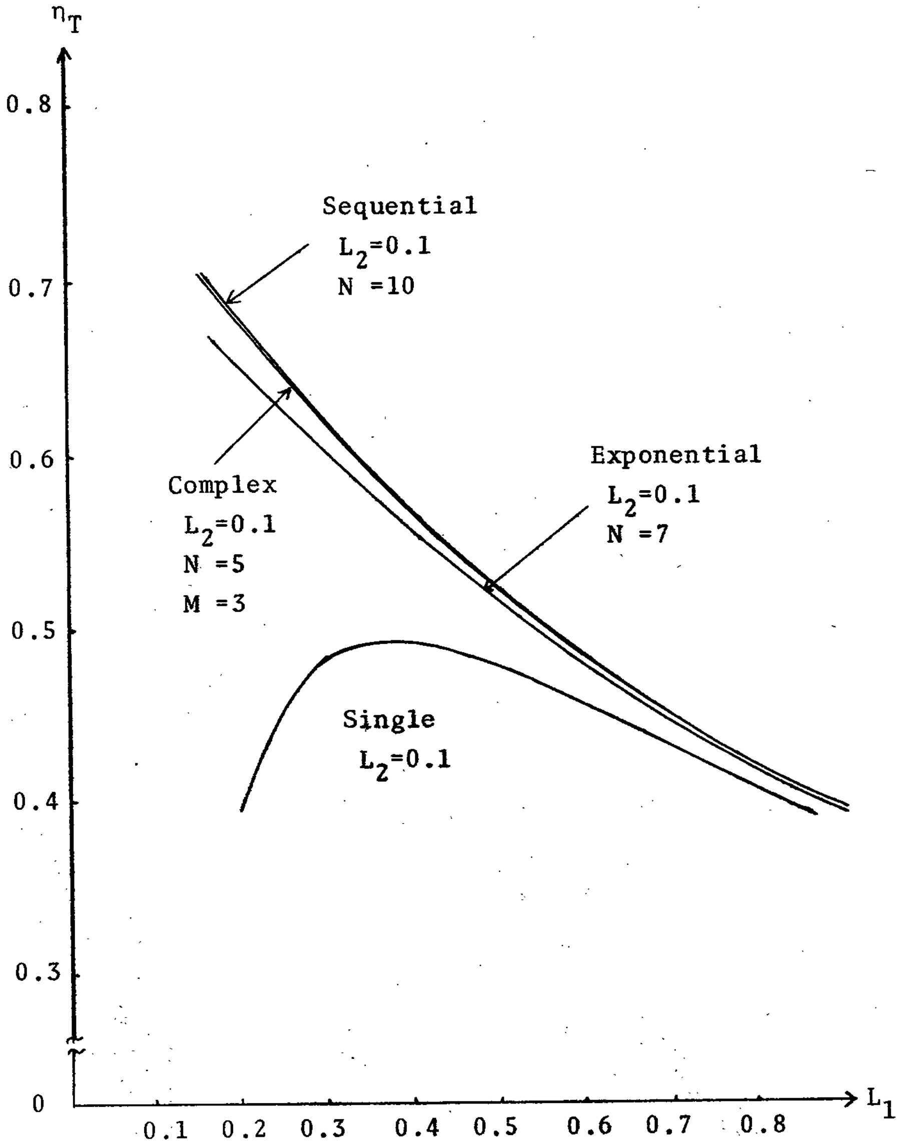


Fig. 2.3 Transmission efficiency versus rate of segment length to message length L_1, in the case that check bits length varies in proportion to segment length ($L_3 = 0.05 \times L_1$, m = 1000, q = 10^{-6})

$$\omega\text{-}\varepsilon = \sum_{i\varepsilon P} P_{sp}(i)\cdot \frac{\varepsilon_i^2}{1-\varepsilon_i} \simeq \sum_{i\varepsilon P} P_{sp}(i)\cdot\varepsilon_i^2\cdot(1+\varepsilon_i) \qquad (2.27)$$

Therefore, q-ε characteristics and q-ω characteristics can be considered as the same characteristics. From Fig.2.2, it is obvious that ε and ω linearly change as q changes.

Fig.2.3 and Fig.2.4 represent L_1-η_T characteristics about N-ACK of four packeting, where $L_3=0.05\times L_1$ on Fig.2.3 and $L_3=0.05$ on Fig.2.4. In both cases, Complex packeting and Sequential packeting have almost similar characteristics and higher efficiency than other two packetings. It is remarkable that Single packeting has low efficiency.

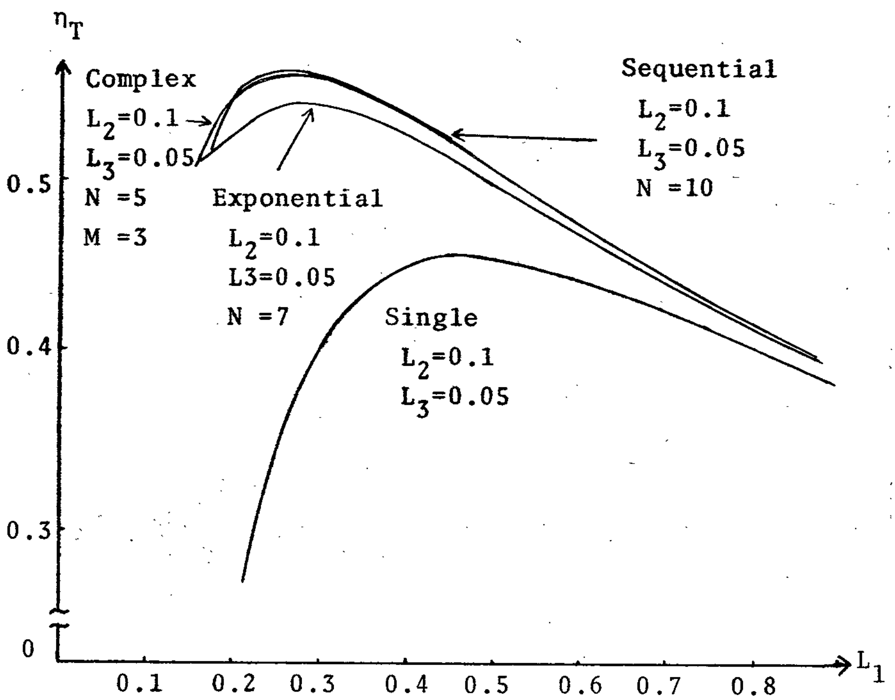


Fig. 2.4 Transmission efficiency versus L_1, in the case that check bits length is fixed ($L_3 = 0.05$, m=1000, $q=10^{-6}$)

Fig.2.5 represents L_1-η_T characteristics about the three ACK methods of Sequential packeting. The transmission efficiency of N-ACK is 20% lower than the one of ACK-free, and P-ACK and A-ACK are 15% down to N-ACK. Here, an ACK-packet has the same length as the one-segment-packet. Hence, if L_4^* and L_5^* are smaller than this, the transmission efficiency of the three ACK methods will be improved.

Fig.2.6 represents q-η_T characteristics. The retransmitted packets affect greatly η_T, where q is larger than 10^{-4}, but if q is smaller than 10^{-6}, the effect is negligible.

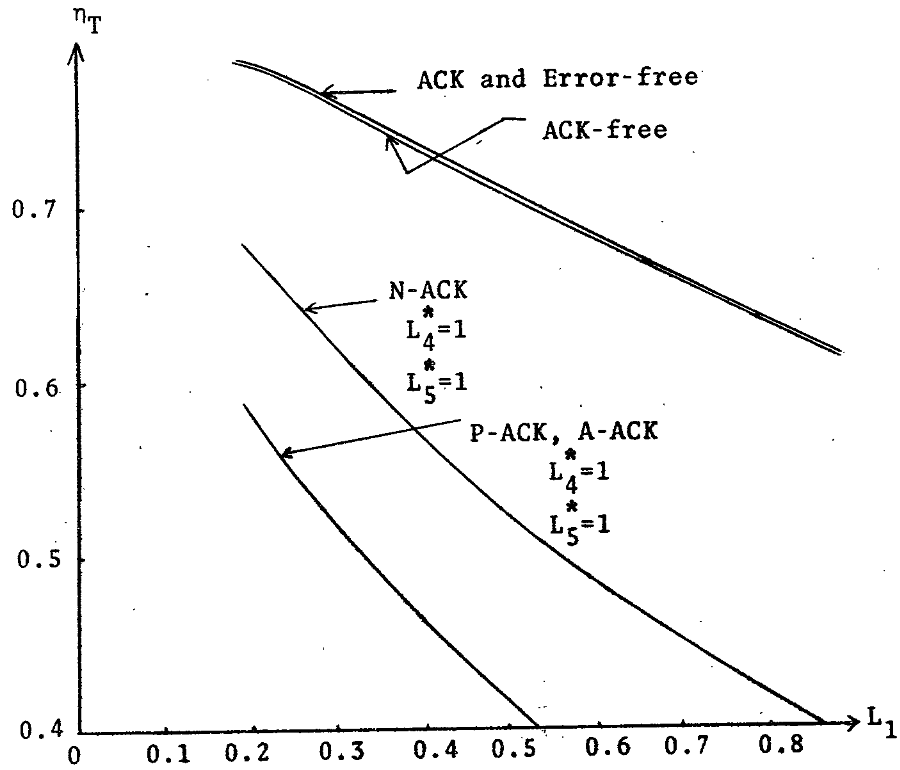


Fig Fig. 2.5 Transmission efficiency for three ACK methods ($m=1000$, $q=10^{-6}$)

2.6 Conclusion

This chapter presents four types of packeting methods and mathematically represents them. Further, as the evaluation measures, the packet error rate ε, the expected number of retransmission ω, and the transmission efficiency η_T are defined. Utilizing these, the packeting methods are analyzed and compared. As the results, the following are obtained.

1) Regarding ε and ω, the Single packeting is the best.
2) Regarding η_T, the Sequential and the Complex packeting show almost similar characteristics and have the highest efficiency.
3) The ε and ω are linearly changed as q changes.
4) On condition that q is smaller than 10^{-5}, the effects of retransmission packets on η_T are negligible.
5) Non-information bits in packet and acknowledgement bits for retransmission extremely affect η_T, so these cannot be neglected.
6) As L_1-η_T characteristics has the maximum value, so it is possible to design the optimal packet format on η_T.

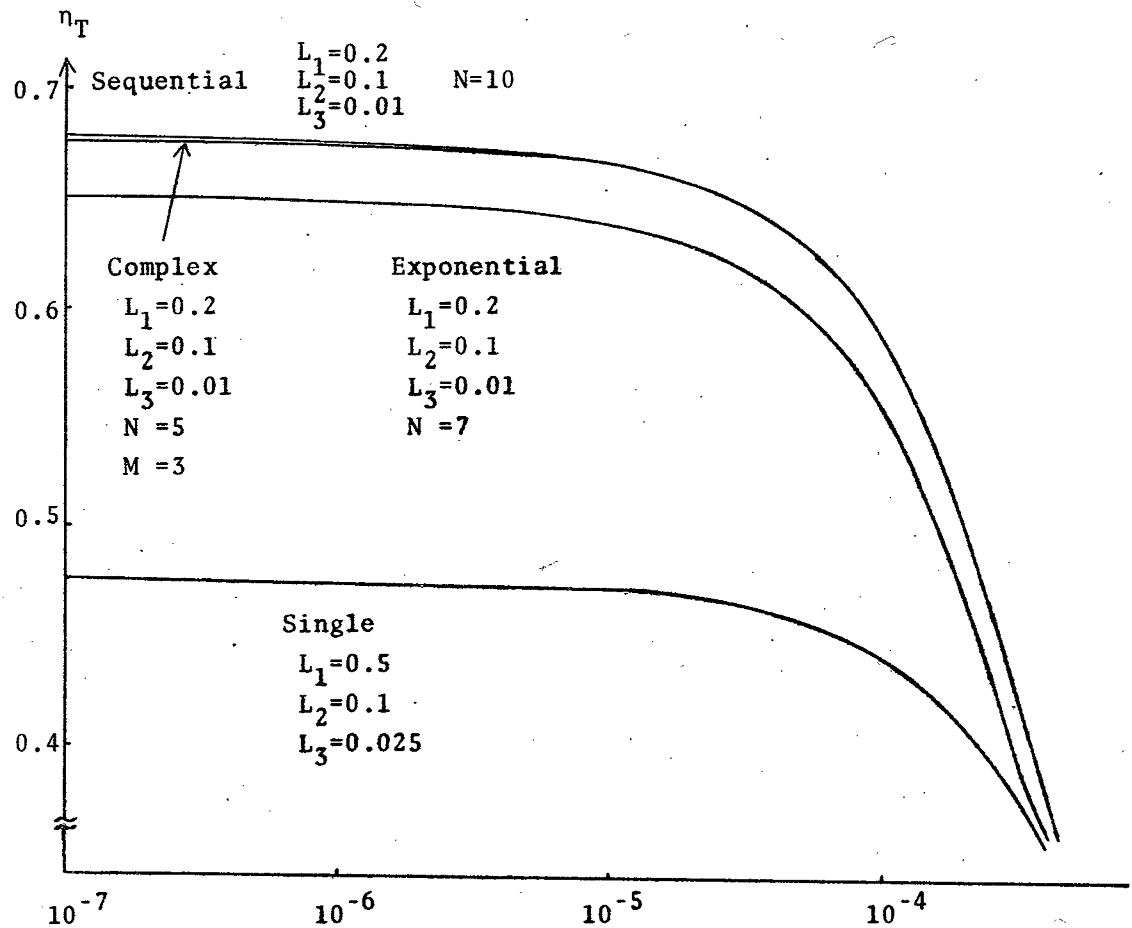


Fig. 2.6 Transmission efficiency versus bit error rate

CHAPTER 3

ANALYSIS OF DELIVERING BEHAVIOR OF PACKET SEQUENCE IN THE NETWORK

3.1 Introduction

In evaluating and designing packet-switching computer-communication network, message traverse time is one of the most important measures. The message traverse time consists of the transmission delay and the reassemble delay. As discussed in the next chapter, the former is analyzed on the basis of queueing problems. The latter is, however, unique to packet switching system. We define this as the time difference between the arrival of the first packet and that of the final packet at the destination, that belong to the same message. Then, the reassemble delay greatly depends on the number of intervening packet between the packet-sequence of the same message.

Furthermore packet switching can be regarded as a kind of time division multiplexing transmission system. In this switching, messages are generally divided into several packets, and these packets are transmitted independently. When we watch at a specific transmission channel, packets of the same messages do not always continue, but frequently packets, which belong to the various messages, are mixed together. Thus, the effect of this TDM is dynamically or naturally given. It is interesting to investigate the grade of mixture among the packets of the various messages.

This chapter discusses on the probability density function of the packet-intervening number between packet-sequence. The validity of the analysis adopted in this chapter is confirmed by the simulation on Single packeting. The mean number of intervening packet and the expectation of intervening bit rates are derived for the various packeting methods.

3.2 Delivering Model

As described in Chapter 1, the analysis of this thesis is limited to computer-communication networks (i.e. high-level networks) and local networks (i.e. low-level networks), that consist of large computers and terminals, are assumed to be the message source and sink.

Further, we adopt a fixed routing procedure or a directory routing procedure, for the algorithm that decides the outgoing channel. In the former case, the overall path of packets is uniquely determined corresponding to the pair of the nodes of the source and the destination. In the latter case, the packets can select the best path at that time among some fixed routings. In both cases, packets of the same message take the same fixed routing. Therefore, for packet-sequence that belongs to the same message, computer-communi-

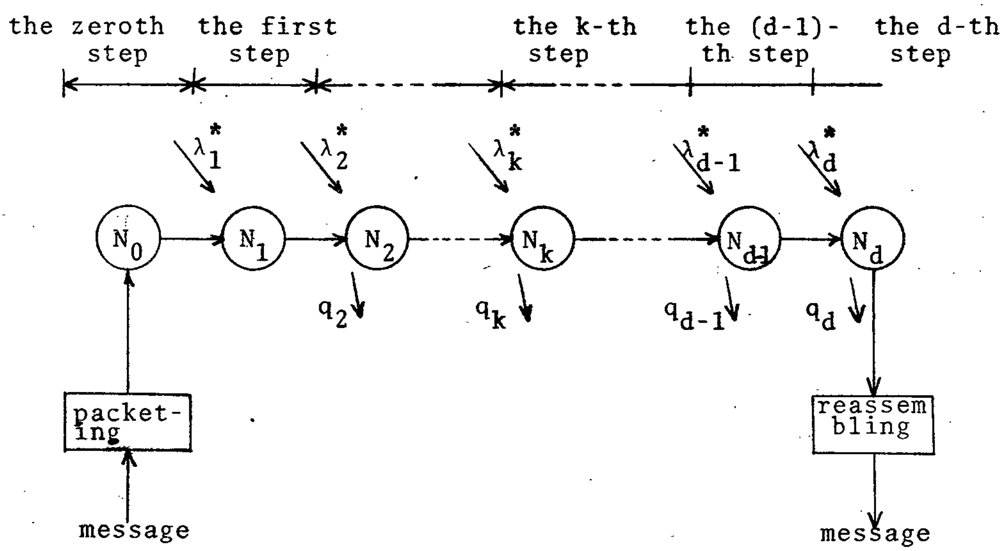


Fig. 3.1 Network model

cation network can be transformed to a series-type network shown in Fig.3.1.

As shown in Fig.3.1, the messages are packeted at the node N_0, and then packet-sequence is transmitted to the destination through several nodes. In this transmitting process, some packets of other messages are intervening packet-sequence. At the destination node N_D, packets are restored untill all packets of the same message arrive and then they are reassembled to the original message.

Each switching node in the network is represented by a model shown in Fig.3.2. The input and the output control buffers are taken as one-segment length. Similarly, transmission and reception between switching nodes are performed, taking a segment as a unit length.

When all the segments in one packet are received, the channel is opened and another packet can utilize it. The received packets

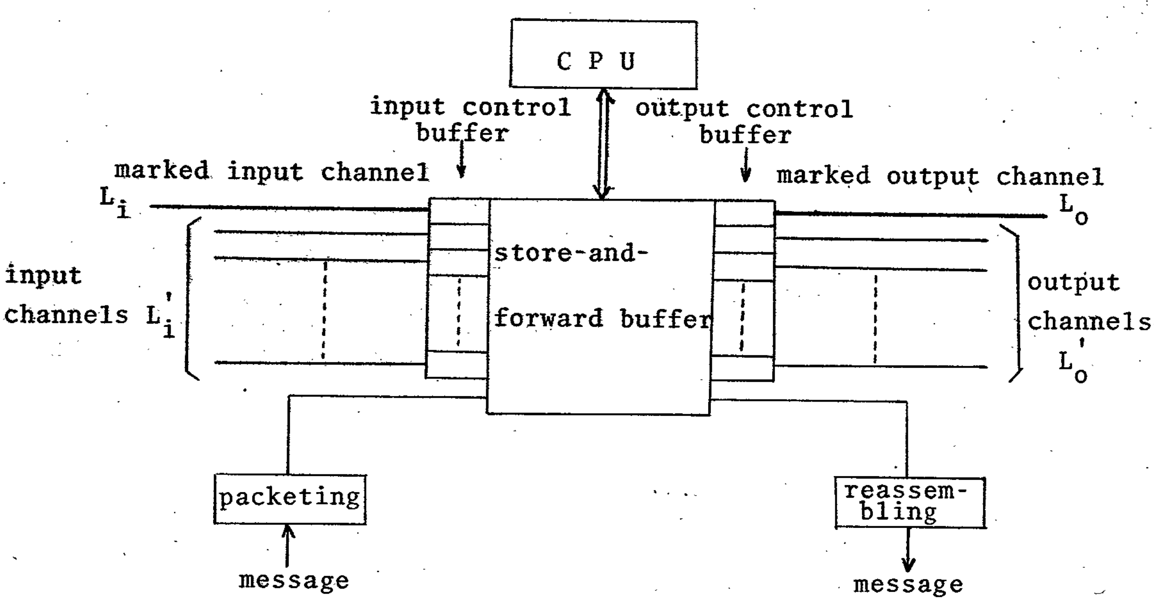


Fig. 3.2 Switching node model

are stored in the communication buffer and subjected to be switched.

Now, we define the source node and the output channel to the first node as the zero-th step as shown in Fig.3.1. Similarly, the k-th node and the output channel to the (k+1)-th node are defined as the k-th step.

Furthermore, the following assumptions are introduced for analysis.

3.1) The average number of packets that arrive from channel i and are transfered through channel j is denoted by $r_{ij}^{(k)}$. So the input traffic at a node N_k is given by the traffic matrix

$$\tilde{R}_k = [r_{ij}^{(k)}]$$

3.2) The time required for switching process (i.e. error detection, output channel control etc.) at switching node is much smaller than the transmission time and can be neglected.

3.3) The transfer of messages from the communication buffer to the reassemble buffer is so fast that messages cannot compose queue.

3.4) A utilization factor of each channel is sufficiently large and null-transmission time between two packets in the same message is negligible.

3.5) In Fig.3.2, the packet generated at node N can be considered to be transmitted through an input channel. Similarly, the packet terminated at node N can be assumed to go out through an output channel.

3.3 Definition of Packet Intervention

In the packet switching, switching tasks at each node and transmission are performed by taking a packet as a unit. However, transfer control and reception control at the nodes are performed by taking a segment as a unit. Therefore, sometimes several packets are received and/or sended through different channels simultaneously. When a packet-sequence of the same message traverses in the network, packets of different messages may intervene between this sequence. Then, the definition of packet intervening, dropping of intervening packets, and the intervening number of packet between two packets in the same message, are given in the following.

<u>Definition 3.1</u> In Fig.3.2, let us denote the packet-sequence by $(\ldots.P_1, P_2, \ldots.)$, which arrives at the node sequentially from an input channel L_o. Now, we consider that, during the reception of packet P_2 after the completion of the reception of the preceding packet P_1, packet P_0', whose output channel is L_1 as P_1 and P_2, is completely received from one of the input channels L_i'. Then, the packet P_o' intervenes between packet P_1 and P_2. So, the packet-sequence which is transmitted to the next node through channel L_o is denoted by $(\ldots. P_1, P_0', P_2', \ldots.)$. We define this as packet intervening.

<u>Definition 3.2</u> In the packet-sequence $(P_{01}', P_{02}', \ldots., P_{0j}')$ which intervenes between P_1 and P_2 at the reception from an input channel L_i, some packets (for example $P_{02}', \ldots., P_{0j}'$) may be transfered through an output channel L_o', which differs from the output channel L_o of packet P_1 and P_2. Since the packets do not pass channel L_o, they do not intervene between P_1 and P_2 on transmitting via channel L_o. Therefore, these packets are considered to be withdrawn from the intervention between P_1 and P_2 at the instance when the reception of these packets is completed at that node. The remainder (i.e. P_{01}') still intervenes between P_1 and P_2 at this step.

<u>Definition 3.3</u> The intervening number n_k at the k-th step between P_1 and P_2 on the output channel at the k-th step. It is assumed here, that P_1 and P_2 are adjacent packets belonging to the same message, and that the intervening packet between them belongs to a different message.

3.4 Markov Property of Intervening Number

To analyze the probability distribution of the intervening number, we introduce the following two approximations.

Approximation 3.1   The number of segments in the packet is given by the probability density function $P_{sp}(i)$.†

Approximation 3.2   The dropping of intervening packets at any step occurs independently and randomly.

Considering these approximations, the intervening number n_k at the k-th step is only dependent on the intervening number n_{k-1} at the (k-1)-th step. So, we represent n_k as follows.

$$n_k = n_{k-1} + n_{int,k-1} - n_{drp,k-1} \tag{3.1}$$

where $n_{int,k-1}$ is the intervening number of packets that are generated at the (k-1)-th step, and $n_{drp,k-1}$ is the dropping number of the intervening packets which are withdrawn at the (k-1)-th step.

Now, we denote by S_n the state that the intervening number is n. Further, we express by X_k the state at the k-th step. So, $X_k = S_n$ means that the state at the k-th step is S_n. Since X_k is only dependent on X_{k-1}, it is clear that the state sequence (X_k) forms a simple Markov chain containing enumerable infinite number of the states.

We define the state probability vector $\bar{S}_k$ as follows.

$$\bar{S}_k = \{ P_{k0}, P_{k1}, \cdots\cdot P_{ki}, \cdots\cdot \} \tag{3.2}$$

where P_{ki} means the probability that the intervening number at the k-th step is i.

At the zero-th step, the message is generated and packeted within a much shorter time than a packet transmission time, and then packet-sequence is transmitted sequentially to the next node. There-

---

† This probability function is described in Chapter 2.

fore, no intervening takes place at this step and the state probability vector has the following initial vector $\bar{S}_0$.

$$\bar{S}_0 = \{ 1, 0, 0, \ldots\ldots \} \tag{3.3}$$

For obtaining $\bar{S}_k$ at a node k, we need $\bar{S}_{k-1}$ and the state transition probability matrix $\bar{T}_{k-1,k}$. The element $t_{ij}^{(k-1)}$ of the (i+1)-th row and the (j+1)-th column in this matrix $\bar{T}_{k-1,k}$ represents the state transition probability from state S_j at the (k-1)-th step to state S_i at the k-th step. Utilizing these matrices, the following relations are established.

$$\begin{aligned}\bar{S}_k &= \bar{S}_{k-1} \cdot \bar{T}_{k-1,k} \\ &= \bar{S}_0 \cdot \bar{T}_{0,1} \cdot \bar{T}_{1,2} \cdots \bar{T}_{k-1,1}\end{aligned} \tag{3.4}$$

Here, the state transition probability $t_{ij}^{(k)}$ is represented as follows.

$$t_{ij}^{(k)} = P\{n_k = j / n_{k-1} = i\} \tag{3.5}$$

The intervention-generating number and the dropping number are independent each other. Thus, Eq.(3.5) is rewritten as follows using Eq.(3.1).

$$\begin{aligned}t_{ij}^{(k)} &= P\{n_{int,k} - n_{drp,k} = j-i \ / \ n_{k-1} = i\} \\ &= \sum_{x_i - x_d = j-i} [P\{n_{int,k} = x_i \ / \ n_{k-1} = i\} \\ &\qquad \times P\{n_{drp,k} = x_d \ / \ n_{k-1} = i\}]\end{aligned} \tag{3.6}$$

where x_i and x_d are non-negative integers. In the next section, we obtain this state transition probability and further the state probability vector $\bar{S}_k$, by analyzing these dropping probability and the intervention-generating probability.

3.5 Probability Density Function of Intervening Number

In the previous section, it is shown that the intervening number at any step forms a simple Markov chian and the initial state probability vector is given. At this point, we need only to know the state transition probability matrix $\bar{T}_{k-1,k}$. The element of this matrix (i.e. $t_{ij}^{(k-1)}=P\{n_k=j/n_{k-1}=i\}$) consists of the dropping probability and the intervention-generating probability. In this section we derive these probabilities and mention the limit distribution of intervening number for infinite step number.

3.5.1 Intervention-generating Probability

We have defined intervening-generating probability as the probability that a packet of other messages intervenes between adjacent packet P_1 and P_2 of the same message.

Because of a large number of input channels, arrival of packets at each node is random. Hence, the generation of intervening packets is also random. In one-step transmission, the number of packets that newly intervenes between P_1 and P_2 depends on the time separation of the reception of P_1 and P_2 at the node of this step, and the average intervening number per unit time, i.e. the intervening rate. At the k-th step, we denote the former by d_k and the latter by r_k^*.

The time separation d_k depends on the intervening number n_{k-1} at the preceding step, the number of each intervening packet and the transmission time of one segment i.e. u_k.† From Approximation 3.1, the total number of segments which are transmitted on the (k-1)-th channel from the receipt time of P_1 till P_2 is completely received at the k-th node, is given by the probability density function $P_{seg}(i)$

---

† We define this u_k as transmission-unit-time.

$$P_{seg}(i/n_{k-1}) \begin{cases} = \sum_{j_1+j_2+\cdots+j_{(n_{k-1}+1)}=i} \{\prod_{L=1}^{n_{k-1}+1} P_{sp}(j_L)\} & i>n_{k-1} \\ = 0 & 0\leq i\leq n_{k-1} \end{cases} \tag{3.7}$$

Letting the channel capacity at the (k-1)-th step be C_{k-1}, the time required for transmitting one segment, u_{k-1}, is given by

$$u_{k-1}=\frac{m\cdot L_1}{C_{k-1}} \tag{3.8}$$

where m is the average message length and L_1 is the ratio of one segment length to m. From Assumption 3.4, there is no null time on transmitting channel between P_1 and P_2. Thus, the time difference d_k is given by

$$\text{Prob}\{d_k=i\cdot u_{k-1}\} = P_{seg}(i/n_{k-1}) \tag{3.9}$$

The average intervening number r_k^* means the average number of packets which arrive from channel L_i' besides L_i and are transfered through L_o. Now, r_k^* is given by

$$r_k^* = \sum_{i\in\{L_0'\}} r_{i,L_0}^{(k)} \tag{3.10}$$

So, the intervening rate λ_k^*, which is the average intervening number per transmission-unit-time, is given by

$$\lambda_k^* = u_{k-1}\cdot r_k^* \tag{3.11}$$

Therefore, when the intervening number of the (k-1)-th step is equal to n, the intervention-generating probability $P_{int}(i/n)$ that i intervening packets occur at the k-th step is given by

$$P_{int}(i/n) = \sum_{j=n+1}^{(n+1)\cdot\max\{P\}} [P_{seg}(j/n)\cdot\frac{\{j\lambda_k^*\}^i\exp(-j\lambda_k^*)}{j!}] \tag{3.12}$$

3.5.2. Dropping Probability of Intervening Packets

The packets which drop at the k-th step belong to the intervening packets at the (k-1)-th step. Thus, the dropping number of the intervening packet at the k-th step depends on the intervening number n_{k-1} at the (k-1)-th step and the dropping rate q_k at this step.

The dropping rate q_k is defined as the ratio of the traffic which transfers through L_0' besides L_0, to all input traffic from channel L_i. So, q_k is given by

$$q_k = \frac{\sum_{j \in L_0'} r^{(k)}_{L_i,j}}{\sum_{j \in \{L_0\ L_0'\}} r^{(k)}_{L_i,j}} \tag{3.13}$$

From Approximation 3.2, dropping of each intervening packet occurs independently with probability q_k. Thus, it is clear that the number of dropping packets at each step obeys the binominal distribution with rate q_k If the intervening number n_{k-1} at the (k-1)-th step is equal to n, the probability $P_{drp}(i/n)$ that i packets are dropping at the k-th step among n packets is given by

$$P_{drp}(i/n) \begin{cases} = {}_nC_i\ q_k^i \cdot (1-q_k)^{n-i} & n \geq 1 \\ = 1 & n=0,\ i=0 \end{cases} \tag{3.14}$$

3.5.3 State Transition Probability

As represented in Eq.(3.5), the state transition probability $t^{(k)}_{ij}$ is the probability that the intervening number n_k of the k-th step is j, under the condition that n_{k-1} is equal to i. Adopting $P_{int}(i/n)$ and $P_{drp}(i/n)$, Eq.(3.6) is rewritten as follows.

$$t_{ij}^{(k)} = \sum_{x_i - x_d = j-i} [P\{n_{int,k} = x_i / n_{k-1} = i\} \times P\{n_{drp,k} = x_d / n_{k-1} = i\}]$$

$$= \sum_{x_i - x_d = j-i} [P_{int}\{x_i/i\} \times P_{drp}\{x_d/i\}] \tag{3.15}$$

Therefore, from Eqs. (3.12), (3.14), and (3.15), $t_{ij}^{(k)}$ is represented as follows.

$$P_{tran}^{(k)}(S_i, S_j) = \sum_{l=0}^{i} \{ {}_iC_l \, q_k^l \cdot (1-q_k)^{l-i} \times \sum_{n=i+1}^{(i+1)\cdot \max\{P\}} [\{ \sum_{j_1+j_2+\cdots+j_{i+1}=n} (\prod_{r=1}^{n+1} P_{sp}\langle j_r \rangle)\} \times \frac{(n\cdot\lambda_k^*)^{l+j-i} \exp(-n\cdot\lambda_k^*)}{(l+j-i)!}]\} \qquad i \leq j$$

$$P_{tran}^{(k)}(S_i, S_j) = \sum_{l=0}^{j} \{ {}_iC_{l+i-j} \, q_k^{l+i-j} \cdot (1-q_k)^{j-l} \times \sum_{n=i+1}^{(i+1)\cdot \max\{P\}} [\sum_{j_1+j_2+\cdots+j_{i+1}=n} (\prod_{r=1}^{n+1} P_{sp}\langle j_r \rangle)\} \times \frac{(n\cdot\lambda_k^*)^{l} \exp(-n\cdot\lambda_k^*)}{l!}]\} \qquad i > j \tag{3.16}$$

where $j_r \varepsilon P$

3.5.4 State Probability and the Limit Distribution

Given channel capacity C_k and traffic matrix $\bar{R}_k$ at each step we can obtain the intervening rate λ_k^* and the dropping rate q_k^* and, further, the state transition probability matrix $\bar{T}_{k-1,k}$. With this matrix $\bar{T}_{k-1,k}$ and the initial state probability vector $\bar{S}_0$, we can derive the state probability vector $\bar{S}_k$ at any step, from Eq.(3.4).

If we want to define the probability density function of the intervening number (we denote this by $f_{int}^{(k)}(i)$ at the k-th step), this function can be obtained easily by the vector $\bar{S}_k$ as follows.

$$f_{int}^{(k)}(i) = P_{ik} \tag{3.17}$$

where P_{ik} is the (i+1)-th elements of $\bar{S}_k$. We note that this function is utilized for deriving the reassemble delay in the next chapter.

Now, we discuss the limit distribution of this vector $\bar{S}_k$. When the network under consideration is uniform, the input traffic matrix and the channel capacity at each step are equal,[†] i.e. the following condition

$$\lambda_k^* = \lambda_0 \ , \ q_k = q_0 \ , \ C_k = C_0 \qquad k \geq 0 \tag{3.18}$$

is established. Then, the state transition probability matrix is equal at all steps.

$$\bar{T}_{0,1} = \bar{T}_{1,2} = \cdots\cdots = \bar{T}_{k-1,k} = \cdots\cdots = \bar{T}_0 \tag{3.19}$$

Hence, from Eqs. (3.19) and (3.4), the state probability vector $\bar{S}_k$ can be simplified as

$$\bar{S}_k = \bar{S}_0 \cdot (\bar{T}_0)^k \tag{3.20}$$

[†] This situation may not be realistic, but in a few special case (for example, in a loop computer network), it is interesting to consider this situation.

When λ_k^* and q_k satisfy

$$\lambda_k^* \cdot q_k \cdot (1-q_k) \neq 0 \tag{3.21}$$

all the elements of $\bar{T}'_{k-1,k}$ become positive.

Therefore, the state sequence (X_k) of intervening number forms a strongly ergodic Markov chain. Therefore, the following are established.

$$\lim_{n\to\infty} (\bar{T}_0)^n = \bar{T}_\infty \tag{3.22}$$

$$\bar{S}_\infty = \bar{S}_0 \cdot \bar{T}_\infty \tag{3.23}$$

The limit of the state probability vector is easily obtained as the solution of the following.

$$\bar{S}_\infty = \bar{S}_\infty \cdot \bar{T}_0 \tag{3.24}$$

3.6 Numerical Results and Consideration

To examine the validity of two approximations, we compare analytical results and simulations on the probability density function $f_{int}(i)$. In simulations, two approximations are not introduced. We consider a three-step network with four nodes in series as shown in Fig.3.1.

The simulation is made by use of SOL(Simulation Oriented Language) of FACOM 230/60 Computer in Kyoto University. Fig.3.3 shows the calculated and the simulation results for L_1=0.5, L_2=0.1, L_3=0.0,† λ_0^*=0.3 and q_0=0.6. A good agreement is seen between the results of theory and simulation. We can insist that two approximations are very suitable.

As numerical examples, some results are presented. We select

† L_2 and L_3 are defined in Chapter 2.

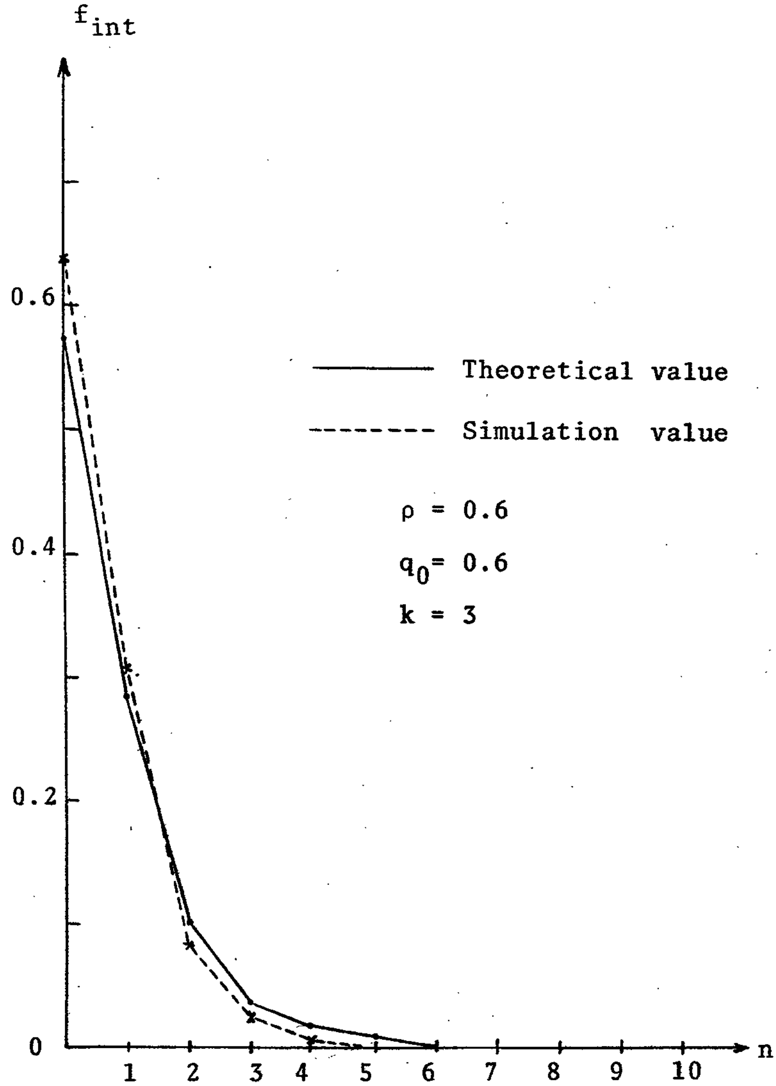


Fig. 3.3 A comparison between theoretical value and simulation value on Single packeting method

three packeting methods, i.e. Single, Sequential (N=10), and Complex (M=3,N=5). The parameter L_1, L_2, and L_3 were determined such that the transmission efficiency η_T is almost equal for the three systems. The dropping rate is $q_0=0.6$ and the intervening rate λ_0^* is varied with channel utilization factor.

Fig.3.4 shows the average number of intervening packets versus the channel utilization factor ρ. Fig.3.5 shows the ratio of the average intervening bit length to the average message length versus ρ. The Complex packeting has the minimum of the average intervening number and the intervening bit length rate. The Single packeting has, however, the maximum. These figures show that in the Complex packeting, packet-sequence of the same message is concentrately transmitted in the network and in the Single packeting, the packets are spread in broad time band.

Therefore, in the Complex packeting, the packet-arrival intervals of the same message are small, which is very advantageous in the design of the reassemble buffer. Fig.3.6 shows the probability distribution of the intervening number for $\rho=0.6$. Fig.3.7 shows the average intervening number versus the number of steps, k for $\rho=0.6$. The Sequential and the Complex packetings exhibit convergence of the average intervening number with an increase of the number of steps. The Single packeting has a convergence value of 10.03.

For $\lambda_0^*<0.5$ and $q_0\geq 0.6$, the state probability for above S_{80} is extremely small and can be neglected. Hence, numerical calculation is made with the maximum number of states as 79.

3.7 Conclusion

In this chapter, the probability density function of packet intervening numbers in the computer-communication networks has been analyzed on the basis of single Markov chain. The effectiveness of

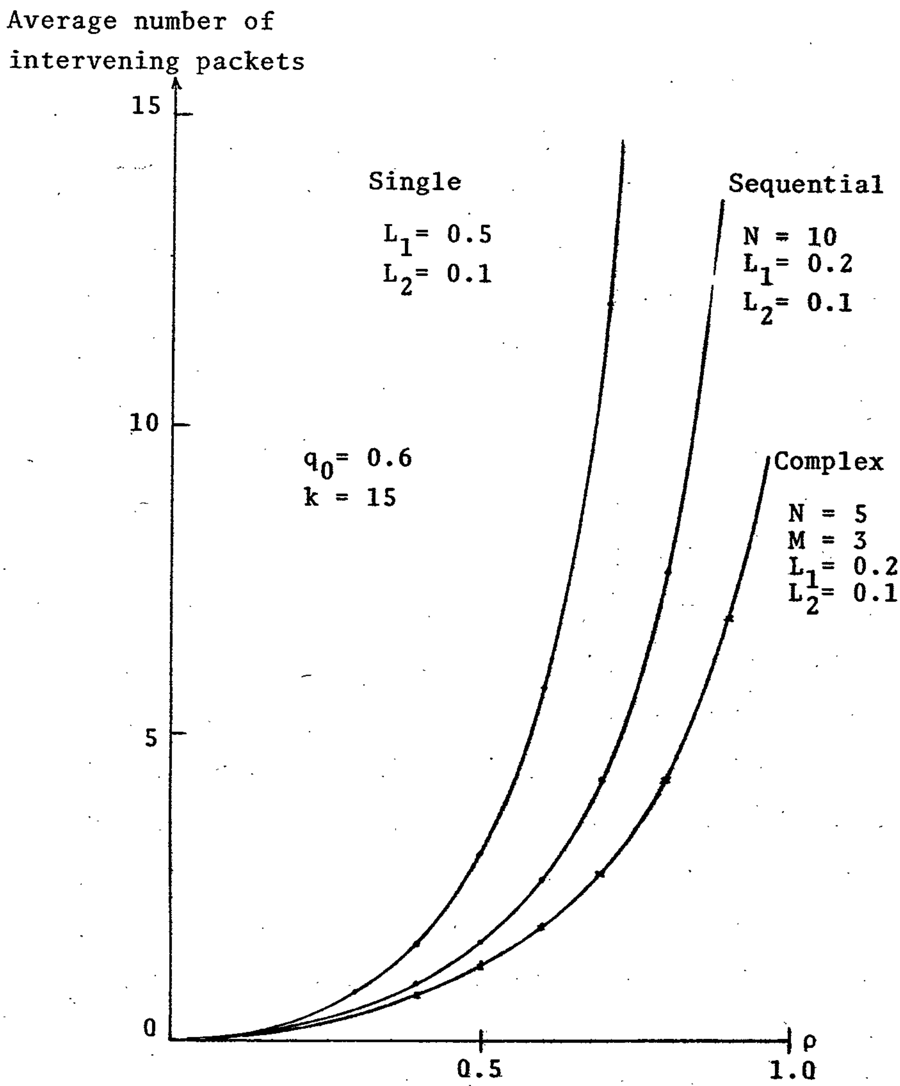


Fig. 3.4 The average number of intervening packet versus channel utilization factor (L_3 = 0.0)

the analysis has been confirmed by the simulation.

Furthermore, on the expectation of the intervening number, various packeting methods have been compared and it was found that the Complex packeting method is the best.

These analyzed results are employed to analyze the reassemble delay in the next chapter.

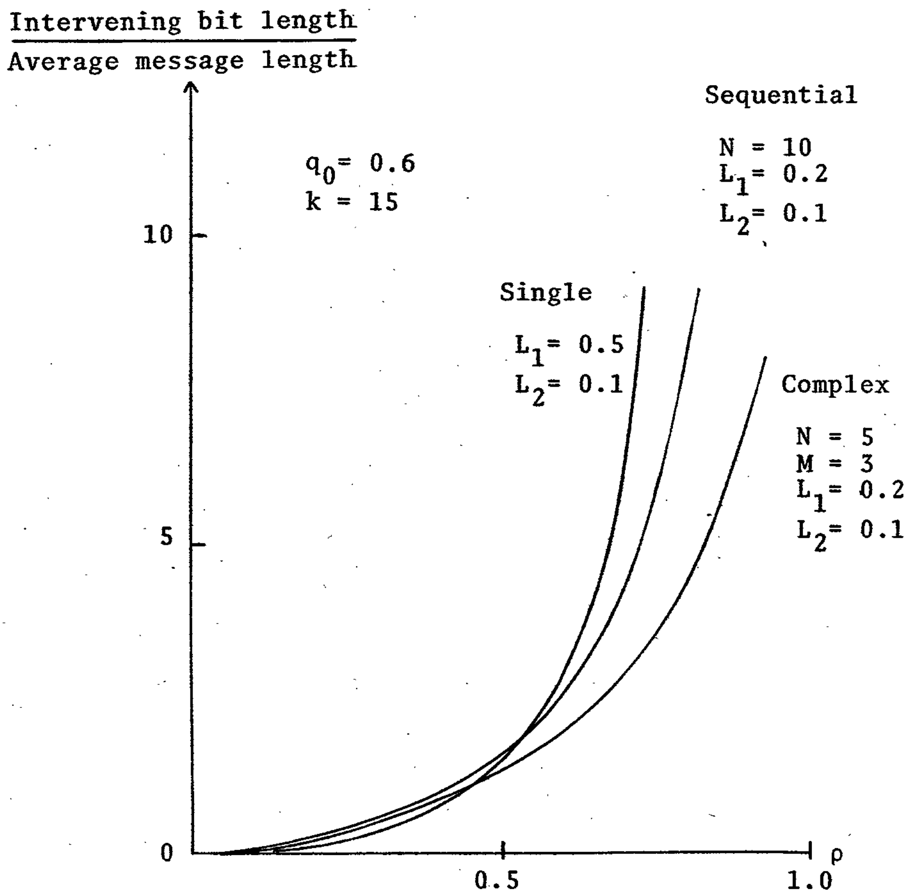


Fig. 3.5 Intervening bit rate versus channel utilization factor ($L_3 = 0.0$)

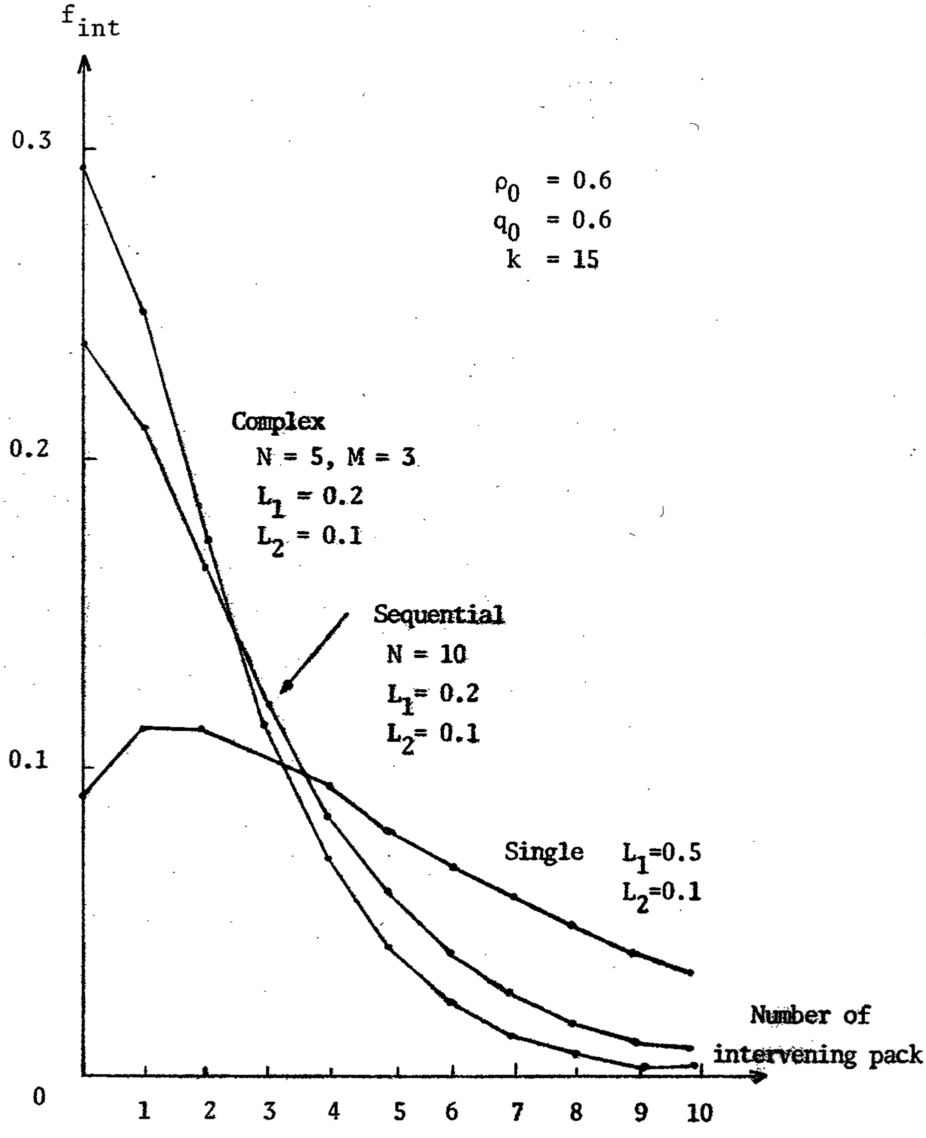


Fig. 3.6 The intervening probability distribution
($L_3 = 0.0$)

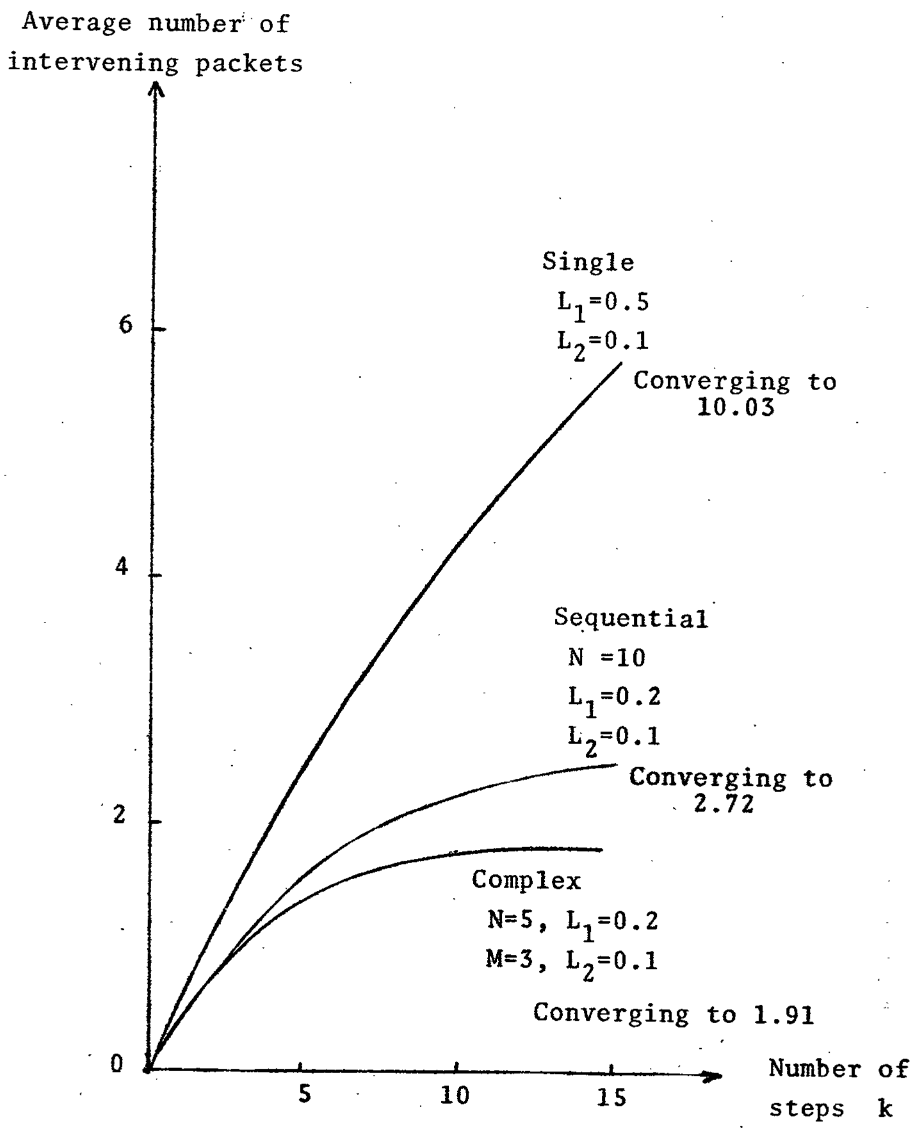


Fig. 3.7 The average intervening number versus tne number of switching steps (L_3 = 0.0)

CHAPTER 4

ANALYSIS OF MESSAGE TRAVERSE TIME

4.1 Message Traverse Model

In evaluating or designing the computer-communication networks, message traverse time is one of the most important measures. This problem was previously studied by some researchers and representative work have been done, especially by the analysis group of ARPA-NET. In this chapter, we will discuss on this problem from the aspects of two unique points. Those are:

1) Packeting methods (i.e. packet composition) are generally represented by the mathematical model shown in Chapter 2.
2) Reassemble delay is derived from the intervening number of packets between packet-sequence of the same message.

As in Chapter 3, we adopt a fixed routing procedure or directory routing procedure. Then, the network may be rewritten as the network of series node in Fig.4.1. The message traverse time t_M can be described as follows.

$$t_M = t_{C_sN_s} + \sum_{i\varepsilon Path\{N_s,N_d\}} \{t_{cpu}^{(i)} + t_T^{(i)}\} + t_{cpu}^{(d)} + t_R + t_{N_dC_d} \tag{4.1}$$

where:

$t_{C_sN_s}$: Transfer time from computer C_s to node N_s

$t_{cpu}^{(i)}$: Waiting and service time of the CPU at node N_i

$t_T^{(i)}$: Waiting and transmission time of the channel from node N_i to node N_{i+1}

t_R : Reassemble time at node N_d

$t_{N_dC_d}$: Transfer time from node N_d to computer C_d

We call sum of $t_{cpu}^{(i)}$ and $t_T^{(i)}$ as nodal delay at node N_i. The transfer time (i.e. $t_{C_sN_s}$ and $t_{N_dC_d}$) between the switching node and the computer is much smaller than the transmission time on the channel. So we neglect it. Hence, to obtain this message traverse time, we must analyze the nodal delay $(t_{cpu}^{(i)}+t_T^{(i)})$, CPU service time $t_{cpu}^{(d)}$, and reassemble time t_R at N_d. Investigating these, network model is divided into two processes.

1) Transmission node process: This process handles nodal delay encountered in passing through each node on the fixed path between N_s and N_d.
2) Reassemble Process: In this process, the reassemble delay, that is defined as the time difference between arrivals of the first packet and the final packet in the same message, is analyzed.

Assumptions required in this analysis of both processes are summarized as follows.

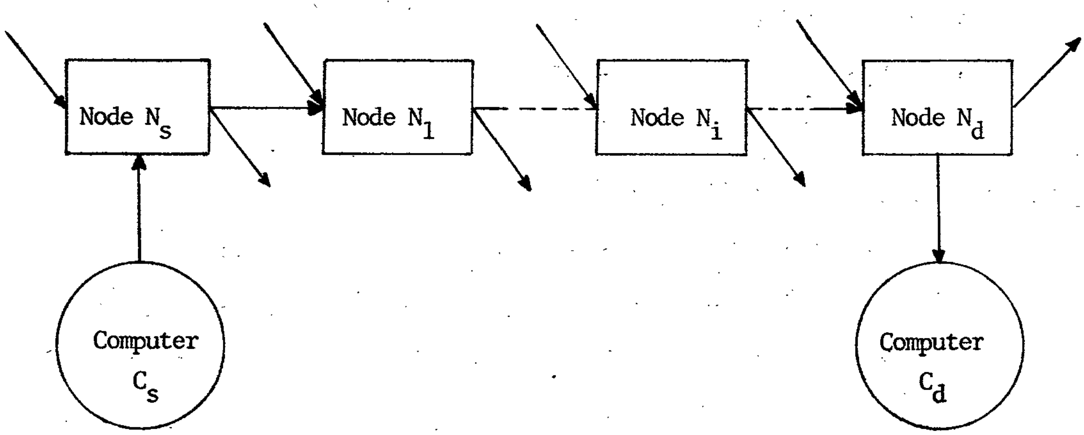


Fig. 4.1 Network model

4.1) For one input channel, arrival of message has the Poisson distribution which has λ as mean per unit time. By packeting this message, arrival of packet has the Poisson distribution, too, that has $\bar{M}\cdot\lambda$ as mean, where $\bar{M}$ is the average number of packet in one message.

4.2) The service time of the CPU at each node has the exponential distribution of $1/\mu_{cpu}$ as mean.

4.3) Whenever a packet is received at a node, a certain number of segments is newly chosen for this packet with probability $P_{sp}(i)$.†

4.4) The capacity of the channel between node N_i and N_{i+1} is given by C_i. For simplicity, we assume that C_i at any node is equal to C.

4.5) At the destination, the number of intervening packets between two adjacent packets in the same message is given by the probability density function of the packet intervening number $f_{int}(i)$, that is derived in Chapter 3.

4.6) Node memory which is utilized for store-and-forward buffer and reassemble buffer is assumed to be infinite.

4.7) Transmission channel is slotted by one-segment transmitting time.

4.2 Analysis of the Nodal Delay

As shown in Fig.4.2, packet which is received at each node is processed by the CPU (i.e. error checking and routing control etc.) and the next node is decided. When this node is the destination of the packet, it transfers the packet into the reassemble buffer, and otherwise comes into a queue of the channel for the next node.

Each node consists of a tandem of two types of queue. Those are

---

† $P_{sp}(i)$ is discussed in Chapter 2.

1) Queue with switching service of the CPU

2) Output queues with channel transmission

The second queues are composed at every output channel. From Assumption 4.6), buffer size of a node is infinite. So two types of these queues can be assumed to be independent of each other. We divide model of nodes into two queueing models and analyze them.

4.2.1 CPU Model

In this model, we can obtain the waiting time and the service time at CPU in the node. By Assumption 4.1), packets arrive from each input channel, having the Poisson distribution with mean $\bar{M}\cdot\lambda$.

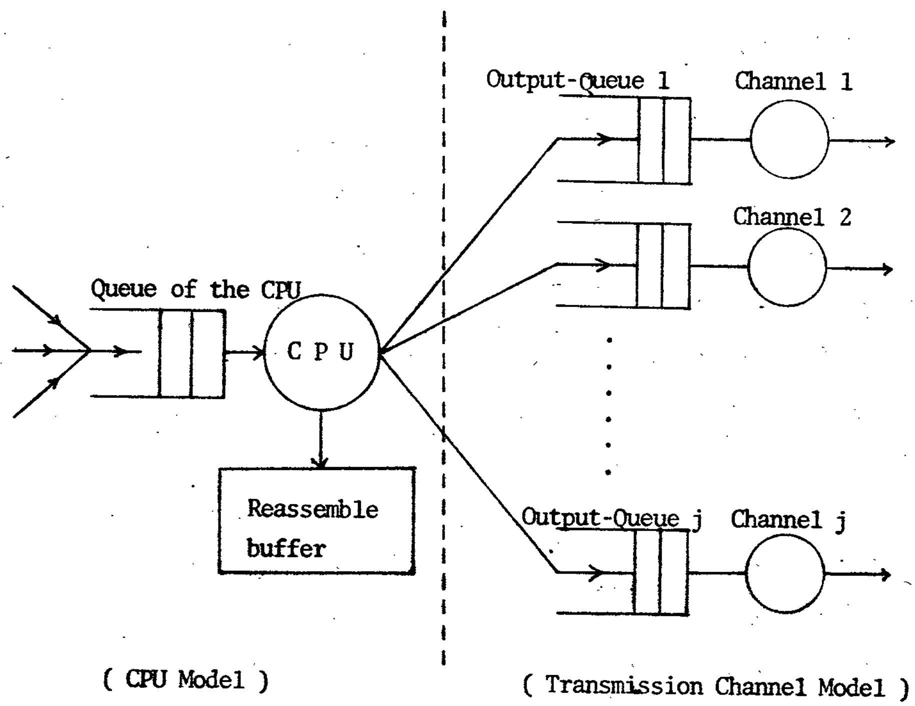


Fig. 4.2 Nodal model

Let the number of input channels be N_I, arrivals to the CPU have the same distribution with mean $N_I \cdot \bar{M} \cdot \lambda$. The service of the CPU is the exponential distribution with mean $1/\mu_{cpu}$. Hence, this CPU model is M/M/1 model in the queueing theory. The queueing time of the CPU at node N_i, is given by

$$T_{cpu} = \frac{1}{(1-\rho_p)} \tag{4.2}$$

where ρ_p is the utilization factor of the CPU and is defined as follows.

$$\rho_p = \frac{N_I \cdot \bar{M} \cdot \lambda}{\mu} \tag{4.3}$$

4.2.2 Channel Transmission Model

After CPU-processing, packet comes into the channel queue to be transmitted to the next node. Since CPU service has the exponential distribution, packet arrival at the channel queue has the Poisson distribution with mean $\bar{M} \cdot \lambda$. Each packet is composed of one or a certain number of segments. So, arrival of segment has a bulk-Poisson arrival with mean bulk-size $\bar{P}$. The $\bar{P}$ is the average segment number in one packet.

The packet transmission time has the special distribution that depends upon the probability $P_{sp}(i)$. But, when we consider segment composing packet, the transmission time of the segment is constant. So taking segment as a unit, this channel transmission model represents queueing model of the bulk-Poisson arrival and constant service.

We have defined a transmission unit time u as the time which is required to transmit one segment length.

$$u = \frac{m \cdot L_1}{C} \tag{4.4}$$

where m is the average message length and L_1 is the ratio of a segment length to m. The average number of packet arrival per u to one channel is denoted by λ_u and given by

$$\lambda_u = \bar{M} \cdot \lambda \cdot u \tag{4.5}$$

Now, we obtain the probability distribution of segment arrival. When the number of arrival packets is k, the probability that the total number of segments is i, is given as follows.

$$P(i/k) = \sum_{s_1+s_2+\cdots+s_k=i} \{ \sum_{j=1}^{k} P_{sp}(s_j) \} \qquad i \geq k > 0$$
$$= 0 \qquad i < k \tag{4.6}$$

Hence, the probability $P_{in}(i)$ that the number of input segments per u is i, is given by

$$P_{in}(i) = \sum_{k=1}^{i} \frac{(\lambda_u)^j \cdot \exp(-\lambda_u)}{k!} \cdot P(i/k) \qquad i > 0$$
$$= \exp(-\lambda_u) \qquad i = 0 \tag{4.7}$$

From Assumption 4.7), we consider that the transmission channel is slotted by one-segment transmission time u. Now, let us observe the number of segments in the queue of the channel at the end of the slotted time. At a time instant t_n, this number only depends upon the number at t_{n-1} and the number of segments in new arrival packets from t_{n-1} till t_n. Therefore, the number of segments in the queue at this time composes the imbedded Markov chain. We denote by $P_k^{(n)}$, the state probability that the number of segments is k at time t_n in the queue. Utilizing Eq.(4.7) and the state probability at t_{n-1}, the state equations at t_n are given by

$$P_0^{(n)} = P_{in}(0)\cdot(P_1^{(n-1)} + P_0^{(n-1)})$$

$$P_k^{(n)} = P_{in}(0)\cdot P_{k+1}^{(n-1)} + \sum_{i=1}^{k} P_{in}(k-i+1)\cdot P_i^{(n-1)} + P_{in}(k)\cdot P_0^{(n-1)}$$

$$\sum_{i=0}^{\infty} P_i^{(n)} = 1 \qquad (4.8)$$

Under the condition that the effective channel utilization factor denoted by ρ_c is smaller than 1, i.e.

$$\rho_c = \lambda_u\cdot\bar{P} = \bar{M}\cdot\bar{P}\cdot\lambda\cdot u < 1 \qquad (4.9)$$

the state equations shown in Eq.(4.8) are converged to the steady state equation as time increases to infinite. We denote by P_k, the state probability that the number of segments in the queue is k. Thus, the steady state equations are given by

$$P_k = P_{in}(0)\cdot P_{k+1} + \sum_{i=1}^{k} P_{in}(k-i+1)\cdot P_i + P_{in}(k)\cdot P_0$$

$$\sum_{i=0}^{\infty} P_i = 1 \qquad (4.10)$$

Hence, the average number of segments in the queue, L_s is given by

$$L_s = \sum_{i=0}^{\infty} i\cdot P_i \qquad (4.11)$$

From Little's formula, the average queueing time of segments, W_s becomes

$$W_s = \frac{L_s\cdot u}{\lambda_u\cdot\bar{P}} \qquad (4.12)$$

Here, W_s is the queueing time of one-segment. Thus, the average queueing time of i-segment-packet, W_i is given by

$$W_i = W_s + u(i-1) \tag{4.13}$$

Therefore, the expected value of the queueing delay of packet, T_T is given by

$$T_T = \sum_{i\varepsilon P} P_{sp}(i)\cdot W_i$$

$$= (\frac{L_s}{\lambda_u \cdot \bar{P}} + \bar{P} - 1)\cdot u \tag{4.14}$$

Thus, we can obtain the transmission delay on any node. Therefore the nodal delay can be given by the summation of the CPU service time and the transmission delay.

4.3 Analysis of Reassemble Delay

When a message is divided into several packets at the source node, all the packets of that message must be reassembled to the original message at the destination node. Then, till the final packet arrives, the preceding packets of the same message must wait for it at the destination node. We define this memory area as the reassemble buffer. We have already defined the time difference between arrival of the first packet and the final packet in the same message, as the reassemble delay.

As shown in Fig.4.1, packets of the same message take the same path. Here, we mark one message. This message is divided into L packets, and these packets are denoted by A_1, A_2,...., and A_L. The time when packet A_i arrives at the destination is denoted by t_{A_i}. Further, the time gap between the arrival time of A_i and A_{i+1} is denoted by τ_{i-1}. Therefore, the reassemble delay of the marked message t_R is defined as follows.

$$t_R = t_{A_L} - t_{A_1}$$

$$= \tau_{L-1} + \tau_{L-2} + \cdots\cdots + \tau_1 \qquad (4.15)$$

To obtain t_R, we must derive τ_i. The time gap τ_i is dependent upon the number of the intervening packets between A_{i+1} and A_i, and the number of segments in each intervening packet. From the Assumption 3.4), there is no null-gap-time between adjacent packets in the same message. Hence, under the condition that the intervening number between A_{i+1} and A_i is n at the node N_{d-1}, that is the preceding node to the destination node N_d, the probability that the time gap τ is k·u, is given by

$$g\{\tau=k\cdot u \ / \ n\} \begin{cases} = \sum_{j_1+j_2+\cdots+j_{n+1}=k} \{\prod_{l=1}^{n+1} P_{sp}(j_l)\} & k \geq n+1 \\ = 0 & k<n \end{cases} \qquad (4.16)$$

Utilizing the probability density function $f_{int}^{(d-1)}(i)$ that has been derived in Chapter 3, the probability that the time gap τ is k·u, is given by

$$g\{\tau=k\cdot u\} = \sum_{j=0}^{k-1} f_{int}^{(d-1)}(j)\cdot g\{\tau=k\cdot u \ / \ j\} \qquad (4.17)$$

With Eq.(4.17), we obtain the probability density function of the time gap between adjacent packets of the same message.

Now, we are ready to derive the probability function of the reassemble delay. This function is the L-fold compounded function of $g\{\tau=k\cdot u\}$. Further, the number of compounded L depends upon the probability function $P_{pm}(i)$, which is derived in Chapter 2.

When message is divided into n packets, the probability that the reassemble delay of this message t_R is i·u, becomes

$$P\{t_R=i\cdot u\ /\ n\} = \sum_{j_1+j_2+\cdots+j_{n-1}=i} \prod_{k=1}^{n-1} g\{\tau=j_k\cdot u\} \qquad i\geq n-1$$

$$= 0 \qquad i<n-1 \qquad (4.18)$$

Since the probability that message is divided into n packets is given by $P_{sp}(n)$, the probability that reassemble time t_R is $i\cdot u$, is

$$P\{t_R=i\cdot u\} = \sum_{k=2}^{i-1} P_{pm}(k)\cdot P\{d=i\cdot u\ /\ k\} \qquad i>0$$

$$= P_{pm}(1) \qquad i=0 \qquad (4.19)$$

Therefore, the average reassemble time T_R is given by

$$T_R = \sum_{j=0}^{\infty} j\cdot P\{t_R=j\cdot u\}\cdot u \qquad (4.20)$$

4.4 Message Traverse Time and Numerical Results

From Eq.(4.1), the message traverse time T_M for given path becomes

$$T_M = \sum_{i\in Path\{N_s,N_d\}} \{T_{cpu}^{(i)} + T_T^{(i)}\} + T_{cpu}^{(d)} + T_R \qquad (4.21)$$

$T_{cpu}^{(i)}$ and $T_T^{(i)}$ are shown in Eq.(4.2) and Eq.(4.14) of Sec.4.2. T_R is derived in Eq.(4.20) of Sec.4.2, when the path is given and the number of steps is known.

Now, for a numerical example, we evaluate the message traverse time on three packeting methods (i.e. Single, Sequential, and Complex). Message length has the exponential distribution with mean 1000 bits. The parameters of the three packeting methods are chosen as follows.

Single : $L_1 = 0.5$ $L_2 = 0.1$ $L_3 = 0.01$

Sequential : $L_1 = 0.2$ $L_2 = 0.1$ $L_3 = 0.01$

Complex : $L_1 = 0.2$ $L_2 = 0.1$ $L_3 = 0.01$

Further, for simplicity, let the channel capacity in all steps C be 10k bits and the average service time of the CPU per one packet $1/\mu_{cpu}$ be 0.01 sec.

Fig.4.3 shows the message traverse time versus channel utilization factor ρ in the case that the step number of the series network is 3 and the channel number of input and/or output to a node is 4. Here, ρ means the channel utilization factor per message defined by the following.

$$\rho = \frac{m \cdot \lambda}{C} \tag{4.22}$$

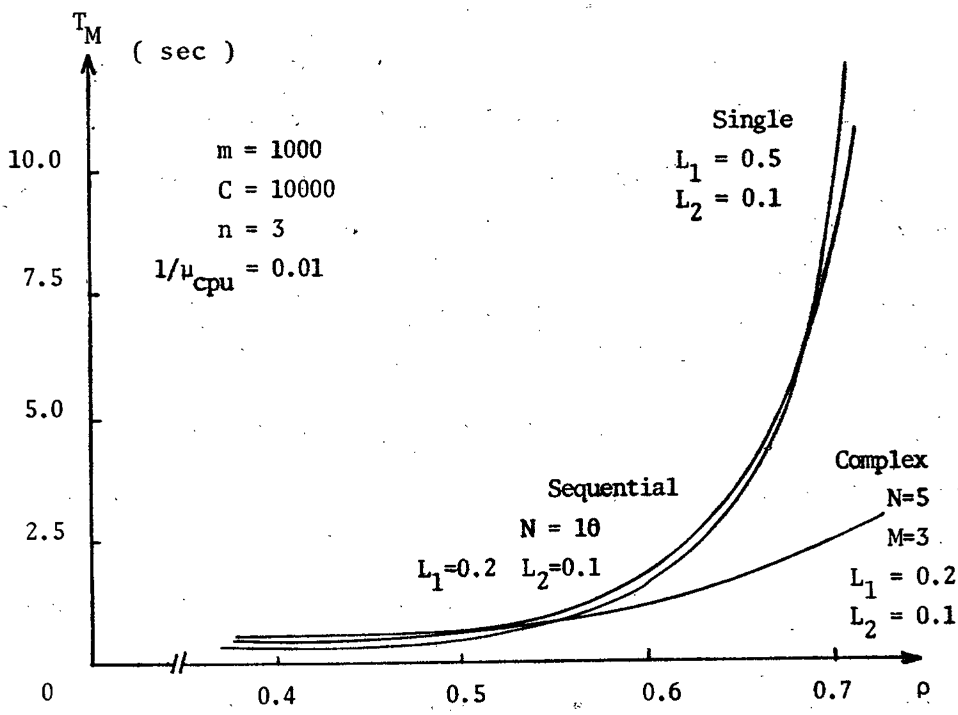


Fig. 4.3 The average message traverse time versus channel utilization (L_3 = 0.01)

From Eq.(4.9) and above Eq., the effective channel utilization factor ρ_c is given by

$$\rho_c = \bar{M}\cdot\bar{P}\cdot\frac{\rho\cdot C}{m}\cdot u \qquad (4.23)$$

Since the three packeting methods have different transmission efficiencies, effective channel utilization factor ρ_c is different in each case for the same channel utilization factor ρ. When ρ is increased to 1, ρ_c tends to 1 at the value of ρ less than 1. In the Single packeting, this value is ρ=0.681. The Sequential packeting has ρ=0.718 and the Complex packeting has ρ=0.777. Fig.4.3 shows that in each method, T_M diverges as ρ tends to this value. In the case that ρ is less than 0.6, the three methods have almost similar characteristics.

Fig.4.4 shows T_M versus the step number in the case that ρ is 0.5 and the number of input/output channels is 4. When n is larger than 4, T_M is linearly increasing as n increases. For the large value of the step number n, the Single packeting is surperior to the others. This is the reason why the Single packeting has a large reassemble delay, but smaller nodal delay. When n tends to a large value, the reassemble delay converges but nodal delay is increasing linearly to n. Therefore, advantage of the Single packeting is obvious.

Fig.4.5 (a), (b), and (c) show the service time of CPU, the channel transmission delay T_T, and the reassemble time T_R versus ρ. We choose the average service time of the CPU as 10 msec.. This is much smaller than the ratio of the message length to channel capacity, 0.1. So, T_{cpu} gives little effect to T_M. Furthermore, channel capacity C=10k bits/sec is relatively small value in the computer-communication networks. That is the reason why T_M has the value of order of second. If we choose C as 50k bits/sec, T_M will be less than the one by a factor of 5 in this case.

4.5 Conclusion

In this chapter, we have shown the analysis of the message traverse time. This consists of three factors. Those are, 1) CPU service time, 2) channel transmission delay, and 3) reassemble delay.

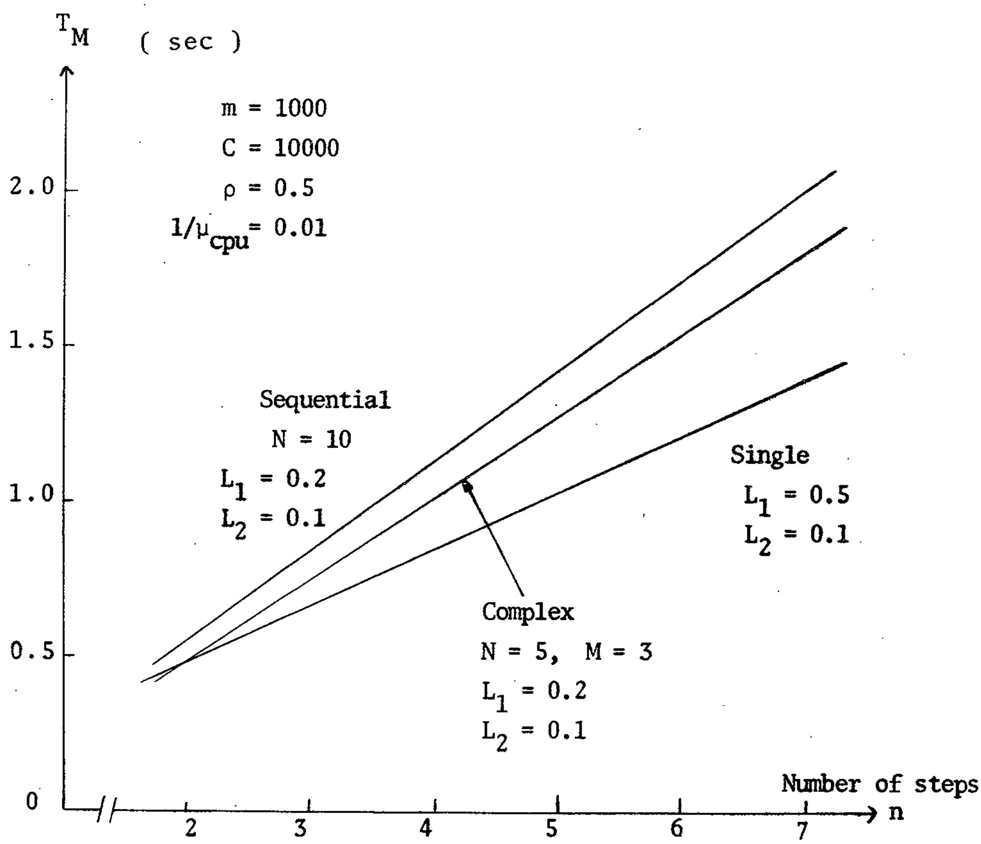


Fig. 4.4 The average message traverse time versus the number of switching steps (L_3 = 0.01)

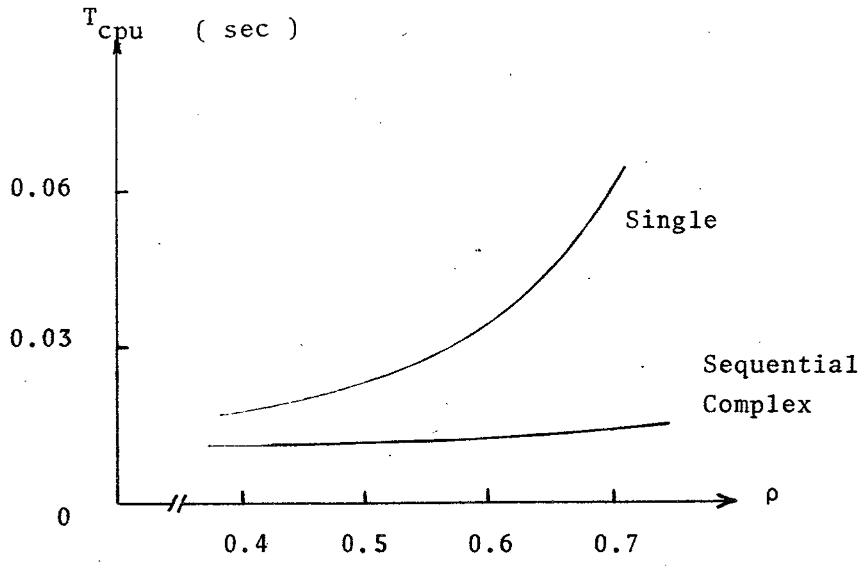


Fig. 4.5(a)   The average CPU time

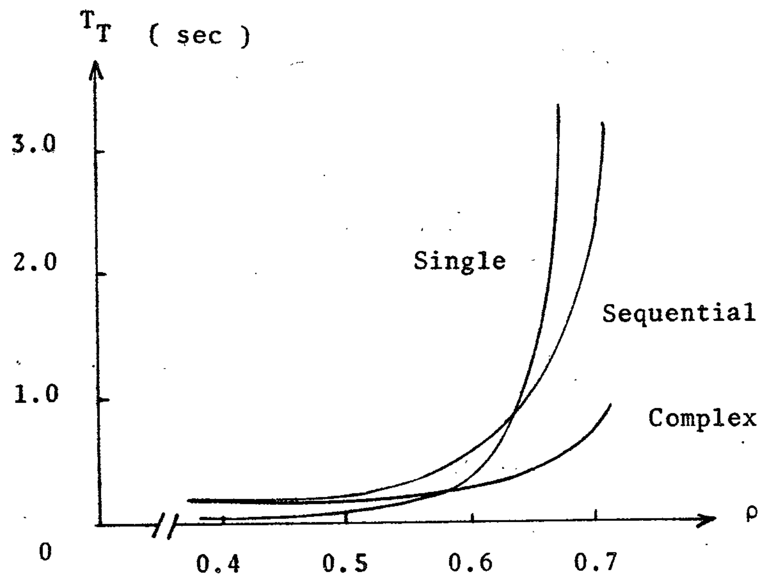


Fig. 4.5(b)   The average channel transmission time

The first is obtained by M/M/1 queueing model. To analyze the second, we consider the channel transmission model as the imbedded Markov chain taken a segment as a unit. We sum up these two into a nodal delay. The third can be easily obtained by the compound method of probability function, utilizing $f_{int}(i)$ and $P_{sp}(i)$. The message traverse delay can be easily derived from these three results.

From this analysis, we can obtain the message traverse time under the condition that channel capacity, network topology, and traffic matrix are given. On the other hand, choosing the fixed value of the message traverse time as an evaluation measure, the optimal designs of capacity assignment and topological structure of the network would become possible.

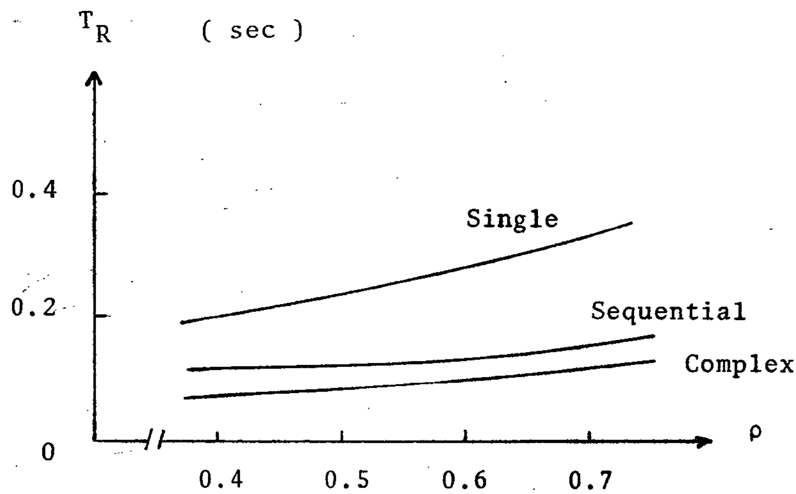


Fig. 4.5(c) The average reassemble time

CHAPTER 5

ACK METHODS FOR PACKET RETRANSMISSION

5.1 Introduction

5.1.1 Selective Repeat ARQ

In packet switching, selective repeat ARQ method is generally used to enhence the reliability of packet channel-transmission. The process of this method is shown in Fig.5.1 and described as follows.

We assume that a node N_A sends a packet to a node N_B. A copy of this packet is kept in the buffer of N_A (defined as copy-buffer). The node N_B checks this packet and sends an ACK signal to N_A to inform the receipt-state. There are two cases. First, if packet is erroneously received, N_B asks N_A to retransmit the same packet. Secondly, if no error is found in the receiving packet, N_B instruct N_A to reject the copy packet. The node N_A continues to send other packets and to keep copies too. If N_A knows the receipt-state of one packet by an ACK signal, N_A decides to retransmit or to reject the copy of this packet, according to the ACK signal. In the former case, the same process is repeated by both nodes and in the latter case, the packet transmission is completed.

5.1.2 Classification of Retransmitted Packet and ACK Signal

Now, we classify packet-retransmission into four cases.

1) Detecting channel errors
2) Rejecting received packets due to the overflow of Store-and-Forward buffer at the receiving node
3) Rejecting the specific packets in order to recover the network from the locking-up due to over-congestion
4) Missing several packets due to error of an ACK signal, channel failure, and system-down of the node, etc.

For the possible case of retransmission as shown above, three types of ACK signals can be considered.

a) Packet-ACK: This is exchanged with the adjacent nodes, whenever one packet is transmitted, and is adapted to the cases 1) and 2). (Sometimes this is defined as link-by-link ACK.)

b) Message-ACK: This is transfered to the message-source node from the message-destination node. This is utilized in the case 4) (Defined as end-to-end ACK).

c) Control-ACK: This is exchanged suitably between the unfixed nodes that irregularly need to be exchanged, and adapted to the case 3).

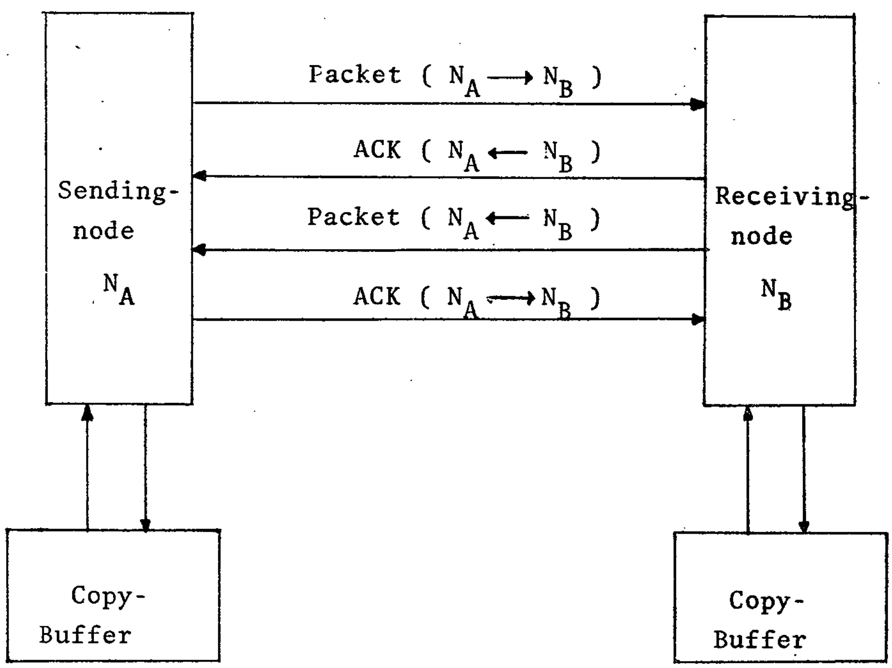


Fig. 5.1 The process of the selective repeat ARQ method

5.1.3 The Objective and the Model

In this chapter, we intend to investigate the control of retransmission between the adjacent nodes in the network. Hence, we deal with the packet-ACK among the three ACKs. The packet-ACK is adapted to the case where channel errors or rejections due to the S/F buffer-overflow occur. A channel error and a packet-rejection are independent of each other, so it is easy to discuss on both cases. But, for simplicity, we only deal with a channel error, since this has higher frequency of occurence than packet rejection.

We discuss on the various types of the ACK methods about packet-ACK for retransmission due to a channel error. In analyzing and evaluating the ACK methods, we adopt two factors as the evaluation measures.

1) Transmission efficiency η_T defined in Chapter 2

2) Packet response time T_{res}

The latter T_{res} is defined as follows.

Definition 5.1 T_{res} is the time separation, from the time when the sending node begins to transmit packet till the time when the sending node decides by a returned ACK signal, whether the same packet must be retransmitted or a copy of that packet must be discarded. During this time T_{res}, copy packet is being kept in the sending node. So, T_{res} is the time required to complete link-by-link transmission of packet.

Furthermore, assumptions required in this analysis are summarized as follows.

5.1) Channel errors on transmission are independent of each other.

5.2) Error rate of i-segment-packet is given by ε_i

5.3) All channel errors are able to be detected.

5.4) ACK signal cannot be damaged by a channel error.

5.5) Buffer-memory size in a node is infinite. So there is no packet rejection.

5.6) Packets which are exchanged between adjacent nodes are independent of each other.

5.7) The service time of the CPU has the exponential distribution.

5.2 Classification of ACK Methods

We defined the means how the receiving node informs the receipt-state of packet to the sending node employing an ACK signal, as ACK methods. These ACK methods are classified into three independent classes, concerning to the way that an ACK signal is transmitted.

The first class is the problem to decide whether an ACK signal (Hereafter we call the ACK signal simply as ACK) is really transmitted. In this class, there are three ways. Those are:

1.a) Positive-ACK (P-ACK): ACK is returned to the sending node of packet only when packet is correctly received. So, if the receiving node detects a channel error, it does not turn an ACK back to the sending node. If the sending node receives no ACK by the specific time T_N, it judges that packet is erroneously received and retransmits the same packet.

1.b) Negative-ACK (N-ACK): This is the reverse to the P-ACK. Thus, ACK is returned to the sending node, only when packet is erroneously received.

1.c) All-ACK (A-ACK): ACK is returned to the sending node, whenever packet is received. The sending node judges the receipt-state by the information in the ACK.

The second class concerns the problem of deciding in which fashion ACK will be transmitted. In this class three types are considered.

2.a) Private packet for ACK: Whole packet is utilized to transmit

ACK

2.b) Piggy-back fashion: ACK is piggy-backed in the header of normal packet which is transmitted to the same node.

2.c) Sub-communication-channel: The network has another sub-channel for transmission of control information besides main channel. In this chapter we omit 2.c).

The third class concerns the number of ACK signals which an ACK packet can carry at a time. There are two cases.

3.a) Single ACK Control: One ACK packet can carry only one ACK.

3.b) Multi-ACK Control: An ACK packet can carry several numbers of ACKs. In multi-ACK control, two control methods can be considered. Those are i) time-control and ii) waiting-number-control. The details are described in Sec.5.5 and Sec.5.6.

Now, these three classes are independent of each other. Combining these three classes, the various ACK methods are obtained. We exhibit some of them, for example.

Ex.1 Single control, private fashion, P-ACK: In this case, P-ACK is utilized. So only when a packet is correctly received, ACK is returned to the sending node with an ACK packet.

Ex.2 Multi-ACK with time-control, piggy-back fashion, A-ACK: In this case, ACK signal is returned whenever packet is received. ACK signal waits for the packet to be transmitted for the same node. When the earliest ACK signal to the specific node waits for piggy-back packet during some constant time, then, all the ACK signals for that node are transmitted with private-packet for ACK.

5.3 Evaluation Measures for ACK Methods

5.3.1 Packet Response Time

As defined in Sec.5.1, packet response time T_{res} is the time separation between the beginning time of packet transmission and the time of deciding which packet is retransmitted or rejected. Hence, the packet response time of i-segment-packet, that is denoted by $t_{res}(i)$, can be generally described as follows.

$$t_{res}(i) = t_p(i) + t_{cpu}^{(R)} + t_{ACK}(i) + t_{cpu}^{(T)} \tag{5.1}$$

where:

$t_p(i)$: Channel transmission time of i-segment-packet from the sending node to the receiving node

$t_{cpu}^{(R)}$: CPU service time at the receiving node

$t_{ACK}(i)$: Time required for the receiving node to inform the receipt-state to the sending node by a certain way in the case of i-segment-packet

$t_{cpu}^{(T)}$: CPU service time required for the sending node to decide whether the receiving node asks to retransmit or not

As defined in Chapter 3, we denote by u, the time required to transmit one segment. So, $t_p(i)$ is given by

$$t_p(i) = i \cdot u \tag{5.2}$$

The service time at the node, $t_{cpu}^{(R)}$ and $t_{cpu}^{(T)}$ can be described as the M/M/1 queueing model as well as the service time described in Chapter 4. Let the average number of arrival packet at the node per unit time be λ_p and the average sevice time of the CPU be $1/\mu_{cpu}$. Then, the utilization factor of the CPU ρ_{cpu} is given by

$$\rho_{cpu} = \frac{\lambda_p}{\mu_{cpu}} \tag{5.3}$$

The average values of $T_{cpu}^{(R)}$ and $T_{cpu}^{(T)}$ become

$$T_{cpu}^{(r)} = T_{cpu}^{(t)} = \frac{1}{\mu_{cpu}(1-\rho_{cpu})} \tag{5.4}$$

The t_{ACK} differs in each ACK method, but can be represented by the general form as follows.

$$t_{ACK}(i) \begin{cases} = t_W^{(R)} + t_p^{(ACK)} & \text{when ACK is returned} \\ = T_c & \text{when ACK is not returned} \end{cases} \quad (5.5)$$

where:

$t_W^{(R)}$: Time that ACK signal waits to transfer to the sending node at the receiving node

$t_p^{(ACK)}$: The channel transmission time of packet with ACK

T_c : The time-out duration when ACK is not returned

Here, the sending node waits for ACK-return after transfering packet. When ACK is not received in the duration T_N, the sending node determines that ACK is not returned and processes the concerning packet in copy buffer. The duration T_N is given by

$$T_N = T_p + T_{cpu}^{(R)} + T_{cpu}^{(T)} + T_c \quad (5.6)$$

where T_p is the average value of packet transmission time and becomes

$$T_p = \sum_{i \in P} P_{sp}(i) \cdot t_p(i) = \bar{S} \cdot u \quad (5.7)$$

In deciding T_N, T_N must have the larger value than the time duration when ACK is most slowly returned. So, T_N is expressed as follows.

$$T_N = \max\{t_p(i)\} + 2\ s\ T_{cpu} + \max\ T_W^{(R)} + t_p^{(ACK)} \quad (5.8)$$

where s is the safty-factor.

Therefore T_c is given by

$$T_c = \max\{t_p(i)\} + \max\{t_W^{(R)} + t_p^{(ACK)}\} + 2 \cdot (s-1) \cdot T_{cpu} - \bar{S} \cdot u \quad (5.9)$$

Analyzing these $t_W^{(R)}$ and $t_p^{(ACK)}$, we can obtain the t_{ACK}, T_c, and further t_{res}.

5.3.2 Transmission Efficiency

Transmission efficiency η_T is discussed in Chapter 2. This is represented as follows.

$$\eta_T = \frac{1}{L_1 \cdot [L_4^* + \sum_{i \varepsilon P} E_{sm}(i) \cdot \{i \cdot \gamma_i + \beta \cdot L_5^*\}]} \tag{5.10}$$

Here, only the value of β and L_5^* depend on ACK methods. The factor β is the number of packet for the packet-ACK to one transmitting packet. L_5^* is the ratio of a packet length for ACK to a segment length. So, for the various ACK methods, by deriving β and L_5^*, we can analyze η_T. The η_T is the expected value of the message information which is included in one transmitted bit. So, $1/\eta_T$ is the expected value of transmitted bits that are required to transmit one message bit.

Now we consider the relation between η_T and effective channel utilization factor ρ^*. Let channel capacity be C and input message traffic be λ, then channel utilization factor per input message, ρ is given by

$$\rho = m \cdot \lambda / C \tag{5.11}$$

The ρ^*, which takes into account of packeting loss, packet-transmission, and ACK transmission, is represented as follows.

$$\rho^* = \frac{(m \cdot \frac{1}{\eta_T}) \cdot \lambda}{C} = \frac{\rho}{\eta_T} \tag{5.12}$$

5.4 Single ACK Control with ACK-private Packet

In this section, we analyze t_{ACK} in Eq.(5.1) for the P-ACK, N-ACK, and A-ACK, when one ACK signal is carried by one ACK-private packet. We defined ACK private-packet as ACK packet. ACK packet is generally composed of one segment. So, in Eq.(5.5), the channel transmission time of ACK packet $t_p^{(ACK)}$ becomes

$$t_p^{(ACK)} = u \tag{5.13}$$

So, in Eq.(5.10), $L_5^*=L_4^*=1$ is determined. Now we only analyze the $t_W^{(R)}$. In this analysis we introduce two approximations as follows.

Approximation 5.1 At the receiving node, the phenomenum that an ACK packet waits for the transmission of an ACK packet seldom occurs.

Approximation 5.2 ACK packet independently occurs to the normal packet. So the occurence of ACK is uniformly distributed to the time duration of packet transmission.

Approximation 5.2 means that an occurence of ACK packet depends upon the probability function of the uniform distribution to the time axes of the normal packet which is transmitted to the same node. This is shown in Fig.5.2.

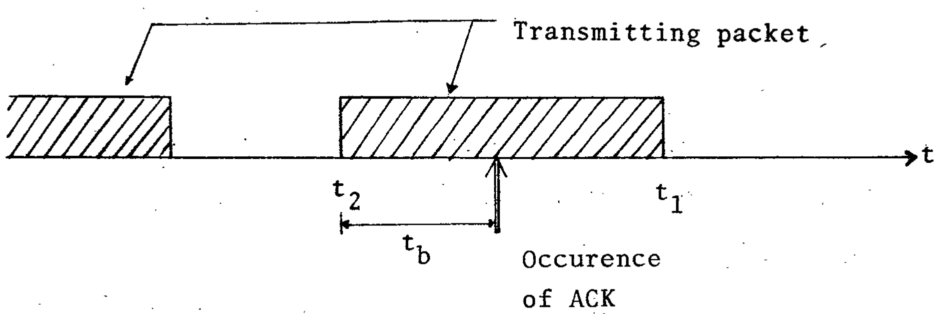


Fig. 5.2 The occurence of ACK signal

From this,

$$t_W^{(R)} \begin{cases} = t_b & \text{to event A} \\ = 0 & \text{to event B} \end{cases} \qquad (5.14)$$

where event A represents the case that when ACK packet is going to transfer, another normal packet is being transmitted on the same channel, and event B is the case that when ACK packet is going to transfer, channel is empty.

In event A, t_b is the time that requires for transmitting packet to complete the transmission. So, t_b has the uniform distribution for earlier packet length. The transmitting packet is the normal packet and has the number of segments depended upon the probability $P_{sp}(i)$. So, the average time of t_b is given by

$$T_b = \sum_{i \varepsilon P} P_{sp}(i) \cdot \frac{i \cdot u}{2} \qquad (5.15)$$

Furthermore, an occurence of event A depends on the probability that the channel is used. This probability is the effective channel utilization factor ρ^* given in Eq.(5.12). Hence, the average value of $t_W^{(R)}$ is given by

$$T_W^{(R)} = \rho^* \cdot T_b \qquad (5.16)$$

The number of ACK-return to one sending packet, and this ρ^* differ in each ACK method. Now, we analyze each case.

5.4.1 P-ACK Method

In this method, only when packet is correctly received, ACK is returned to the sending node. If an ACK is returned during time T_N, the sending node discards the copy of packet, otherwise it retransmits the same packet. The probability of retransmission of i-segment-packet is equal to the packet error rate ε_i. So the rate that ACK is returned is $(1-\varepsilon_i)$ and the one that ACK is not returned is ε_i. Hence, from Eq.(5.5), the t_{ACK} of i-segment-packet is given by

$$t_{ACK}(i) = (1-\varepsilon_i)\cdot(\frac{\rho^*\cdot\bar{S}\cdot u}{2} + u) + \varepsilon_i\cdot T_c \tag{5.17}$$

The packet response time of i-segment packet, T_{res}, becomes

$$T_{res}(i) = i\cdot u + \frac{2}{\mu_{cpu}\cdot(1-\mu_{cpu})} + t_{ACK}(i) \tag{5.18}$$

Therefore, the average value of t_{res}, the transmission efficiency η_T, and the constant-time-duration T_c are given by

$$T_{res} = \sum_{i\varepsilon P} P_{sp}(i)\cdot t_{res}(i)$$

$$= \bar{S}\cdot u + \frac{2}{\mu_{cpu}\cdot(1-\rho_{cpu})} + (1-\varepsilon_i)\ (\frac{\rho^*\cdot\bar{S}\cdot u}{2}+u) + \varepsilon_i\cdot T_c \tag{5.19}$$

$$\eta_T = \frac{1}{L_1\cdot[1 + \sum_{i\varepsilon P} E_{sm}(i)\cdot\{i\cdot\gamma_i + (1-\varepsilon_i)\}} \tag{5.20}$$

$$T_c = 2\cdot u\cdot \max\{P\} + 2\cdot(s-1)\cdot T_{cpu} + u - \bar{S}\cdot u \tag{5.21}$$

where ε is the packet error rate and given by Eq.(2.20).

5.4.2 N-ACK Method

This method is considered as the reverse to P-ACK, and only when packet is erroneously received, ACK is returned. So, the rate that ACK is returned is ε_i and the one that ACK is not returned is $(1-\varepsilon_i)$. The average value of packet response time and the transmission efficiency η_T are given by

$$T_{res} = \bar{S}\cdot u + \frac{2}{\mu_{cpu}\cdot(1-\rho_{cpu})} + \varepsilon_i\cdot(\frac{\rho^*\cdot\bar{S}\cdot u}{2}+u) + (1-\varepsilon_i)\cdot T_c \tag{5.22}$$

$$\eta_T = \frac{1}{L_1 \cdot [1 + \sum_{i \varepsilon P} E_{sm}(i) \cdot \{i \cdot \gamma_i + \varepsilon_i\}]} \tag{5.23}$$

In this case, T_c is the same in Eq.(5.21).

5.4.3 A-ACK Method

In this method, the receiving node returns ACK, whenever packet is received. So the rate that ACK is returned is 1. The times t_{ACK}, t_{res}, T_{res}, and η_T are given by

$$t_{ACK}(i) = \frac{\rho^* \cdot \bar{S} \cdot u}{2} + u \tag{5.24}$$

$$t_{res}(i) = i \cdot u + \frac{2}{\mu_{cpu} \cdot (1-\rho_{cpu})} + t_{ACK}(i) \tag{5.25}$$

$$T_{res} = (\bar{S} + \frac{\rho^* \bar{S}}{2} + 1) \cdot u + \frac{2}{\mu_{cpu}(1-\rho_{cpu})} \tag{5.26}$$

$$\eta_T = \frac{1}{L_1 \cdot [1 + \sum_{i \varepsilon P} E_{sm}(i) \cdot (i \cdot \gamma_i + 1)} \tag{5.27}$$

5.4.4 Results of Simulation and Calculation

To examine the validity of approximations, we compare analytical results and simulation on the packet response time. The simulation is made using GPSS of IBM 370/165 Computer in Touyo Information System Ltd.

Fig.5.3 shows the packet response times of the calculation and the simulation versus channel utilization factor ρ, for L_1=0.5, L_2=0.05, L_3=0.01, and $q=10^{-6}$ with the Single packeting. The correspondence between simulation and theory can be said good. Thus, the assumptions that we have introduced are proved suitable.

Fig.5.4 shows the transmission efficiency versus bit error rate q. Fig.5.5 shows the packet response time versus bit error rate q. In both figures, we can observe that, when bit error rate q is less than 10^{-4}, packet retransmission cannot affect the η_T and T_{res}. This is the same result as shown in Fig.2.7 in Chapter 2. In Chapter 2 we compare the various packeting methods, and in this chapter we compare the various ACK methods.

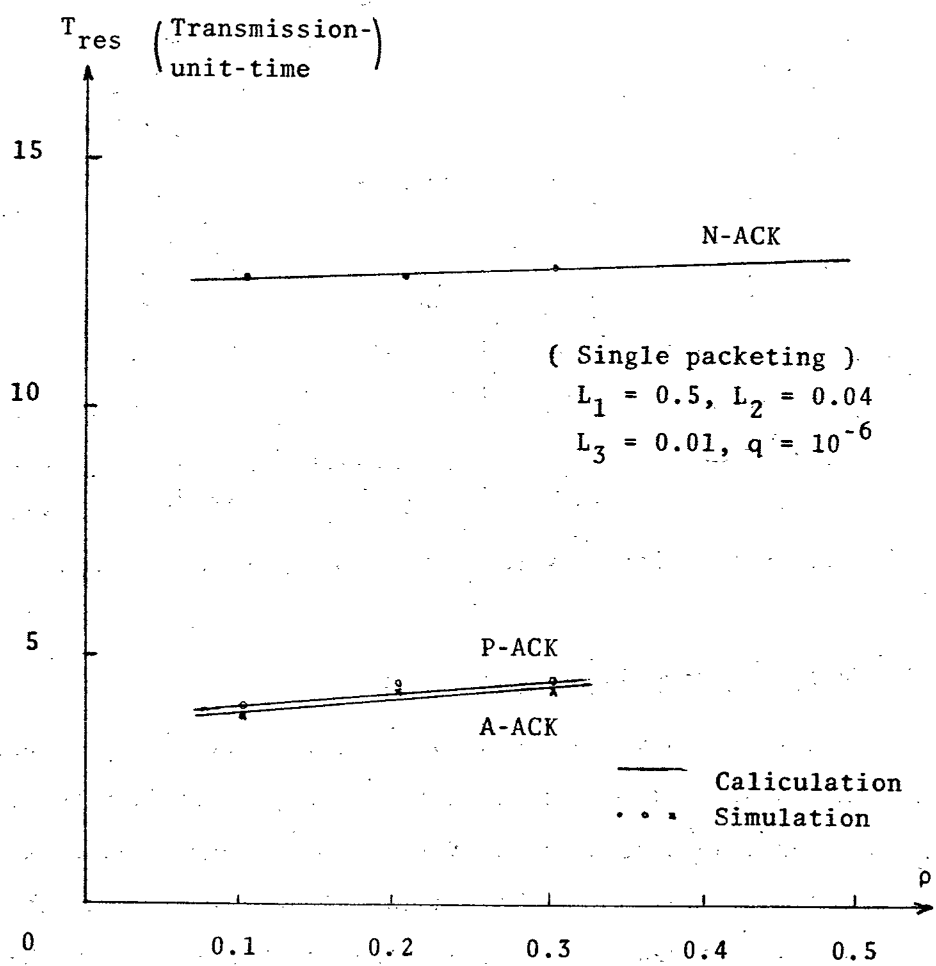


Fig. 5.3 The average packet response time versus bit error rate

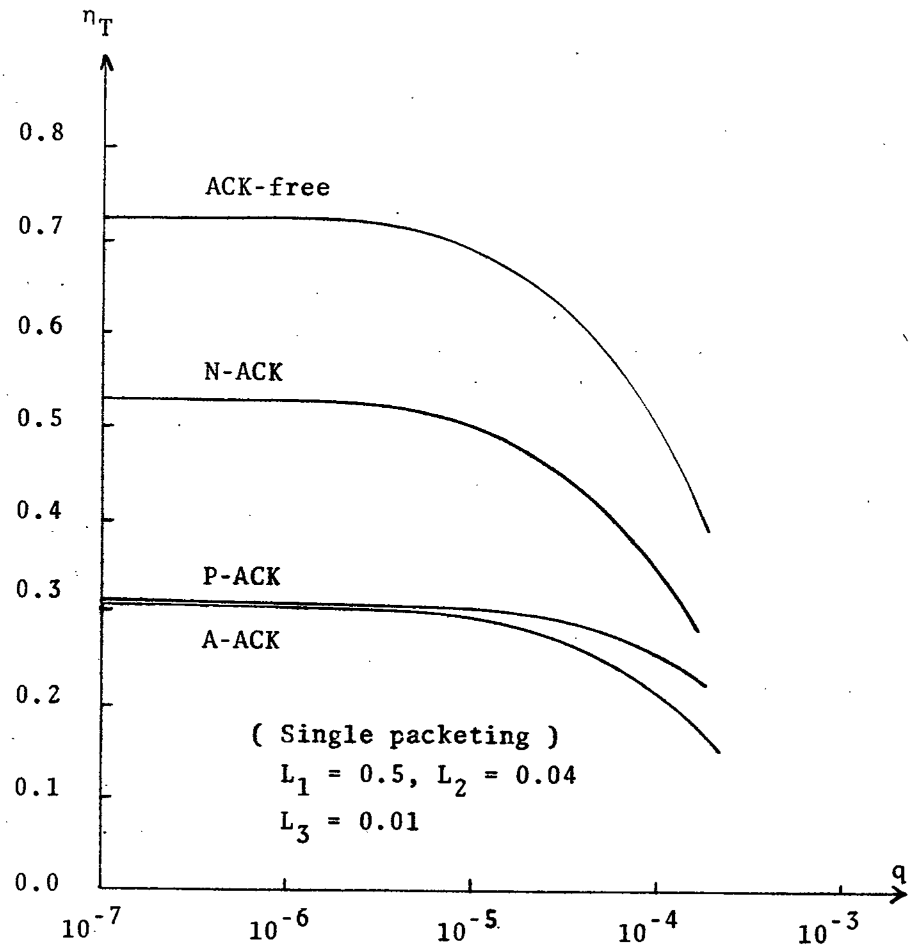


Fig. 5.4 Transmission efficiency versus bit error rate

5.5 Multi-ACK Methods with Time-Control

In this and next sections, we deal with multi-ACK methods. In this case, both ACK-private packet and piggy-back fashion are analyzed. Here, we note that the former section dose not deal on the piggy-back fashion. When the single-ACK method is used, this may be possible but not realistic. The validity of the piggy-back fashion is derived only by the multi-ACK methods.

In multi-ACK methods, $t_W^{(R)}$ of Eq.(5.5) is generally represented as follows.

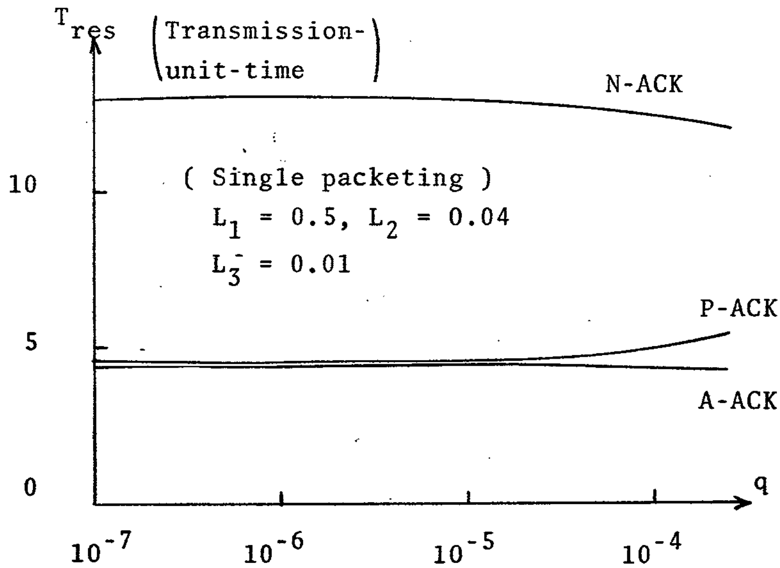


Fig. 5.5 The average packet response time versus bit error rate

$$t_W^{(R)} = t_W' + t_b \qquad \text{to event A}$$
$$= t_W' \qquad \text{to event B} \qquad (5.28)$$

where the events A and B are similar to those in Eq.(5.14), and t_W' is the time separation between the time when ACK occurs and the time when ACK-return is possible after the constraint is satisfied. The time t_b is the same in Sec.5.4.

In almost all cases ACK must wait at the receiving node for this T_W'. The constraint described above differs from each multi-ACK method. This section deals with time-control multi-ACK.

5.5.1 <u>ACK private-packet fashion</u>

Between any adjacent nodes, ACK packet is exchanged, whenever constant time duration D is passed. All ACK signals that occur during this D return to the sending node by one ACK packet which has one-segment length. So the constraint in this ACK method is that ACK packet is transmitted only at the fixed time. The number of ACK signals that occur during D has the Poisson distribution, since the CPU serves packets and generates ACK signals depending on the Poisson distribution. We denote this by $P_{ACK}(i)$.

$$P_{ACK}(i) = \frac{\exp(-\lambda_A \cdot D) \cdot (\lambda_A \cdot D)^i}{i!} \tag{5.29}$$

where λ_A is the average number of ACK signal that the CPU generates per nuit time and is represented as follows for each case.

$$\lambda_A \begin{cases} = \bar{M} \cdot \lambda \cdot (1-\varepsilon) & \text{where P-ACK} \\ = \bar{M} \cdot \lambda \cdot \varepsilon & \text{where N-ACK} \\ = \bar{M} \cdot \lambda & \text{where A-ACK} \end{cases} \tag{5.30}$$

Fig.5.6 shows the occurences of k ACK signals. We denote by t_i, the time when the i-th ACK signal occurs, and x_i is defined as follows.

$$x_i = t_i - t_{i-1} \tag{5.31}$$

Thus, x_i has the exponential distribution f(x) with mean $1/\lambda_A$.

$$f(x) = \lambda_A \cdot \exp(-\lambda_A x) \tag{5.32}$$

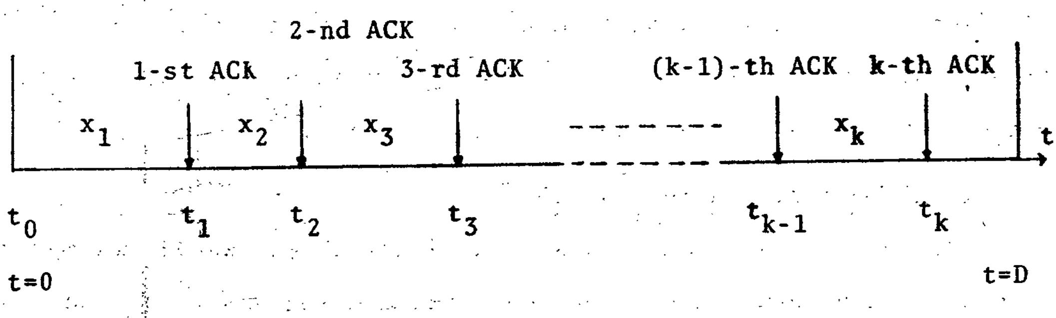


Fig. 5.6 The occurence of ACK signals in time duration D

The average value of x_i, that is denoted by X_i, is given by

$$X_1 = \int_0^D x_1 \cdot f(x_1) dx_1$$

$$X_2 = \int_0^D f(x_1) dx_1 \int_0^{D-x_1} x_2 \cdot f(x_2) dx_2$$

$$\vdots$$

$$X_i = \int_0^D f(x_1) dx_1 \int_0^{D-x_1} f(x_2) dx_2 \cdot\cdots\cdot \int_0^{D-(x_1+x_2+ \quad +x_{i-1})} x_i \cdot f(x_i) dx_i \tag{5.33}$$

We execute integration and obtain the following.

$$X_i = \frac{1}{\lambda_A} - \exp(-\lambda_A \cdot D) \cdot \{\frac{1}{\lambda_A} + \sum_{j=1}^{i} \frac{D^j \cdot \lambda_A^{j-1}}{j!}\} \tag{5.34}$$

The average time of $t_W^{'}$, under the condition that k ACK occur during D, that is denoted by $T_W^{'}(k)$, is given by

$$T_W^{'}(k) = D - \frac{\sum_{i=1}^{k} (k-i+1) \cdot X_i}{k} \tag{5.35}$$

Hence, the average time of $t_W^{'}$ is given by

$$T_W^{'} = \sum_{k=1}^{\infty} P_{ACK}(k) \cdot T_W^{'}(k) \tag{5.36}$$

Therefore, T_{res}, η_T, and T_c are

$$T_{res} = \bar{S} \cdot u + 2 \cdot T_{cpu} + (1-P) \cdot T_c + P \cdot \{\sum_{k=1}^{\infty} P_{ACK}(k) \cdot T_W^{'}(k) + \rho^* \cdot (\frac{\bar{S} \cdot u}{2} + u)\} \tag{5.37}$$

$$\eta_T = \frac{1}{L_1 \cdot \{1 + \sum_{i \varepsilon P} E_{sm}(i) \cdot (i \cdot \gamma_i + \frac{\beta_i}{\lambda_A \cdot D})\}} \tag{5.38}$$

$$T_c = \{2 \cdot \max\{P\} + 1 - \bar{S}\} \cdot u + D + 2 \cdot (s-1) \cdot T_{cpu} \tag{5.39}$$

where

$$P \begin{cases} = 1-\varepsilon_i & \text{where P-ACK} \\ = \varepsilon_i & \text{where N-ACK} \\ = 1 & \text{where A-ACK} \end{cases}$$

$$\beta_i \begin{cases} = 1 & \text{where P-ACK} \\ = \gamma_i - 1 & \text{where N-ACK} \\ = \gamma_i & \text{where A-ACK} \end{cases}$$

5.5.2 Piggy-back Fashion

The ACK signals piggy-back to the normal packet which is transmetted to the identical node N_T. But, after the node N_R sends the normal packet, if there occurs no normal packet to N_T during the fixed time D, an ACK private-packet is dispatched to transmit all the ACKs. The probability P_{AP} that an ACK packet is dispatched is the probability that there occurs no normal packet to node N_T during D. Thus, P_{AP} is given by

$$\begin{aligned} P_{AP} &= 1 - \int_0^D \lambda_p \exp(-\lambda_p \cdot x)\,dx \\ &= \exp(-\lambda_p \cdot D) \end{aligned} \tag{5.40}$$

where λ_p is the average number of normal packet that is transmitted to the node N_T per unit time, and becomes

$$\lambda_p = \bar{M} \cdot \lambda$$

In the case where an ACK piggy-backs to the normal packet, the waiting time of the ACK $T_{W1}^{(R)}$ at the node N_R is given by

$$T_{W1}^{(R)} = P_{other}\cdot(T_{other} + T_{emp}) + (1-P_{other})\cdot T_{emp}$$
$$= P_{other}\cdot T_{other} + T_{emp} \qquad (5.41)$$

where P_{other} is the probability that when ACK is being transmitted, there exists transmitting packet on channel, T_{other} is the average time that required the packet to be transmitted completely, and T_{emp} is the time separation till the next normal packet occurs. In Sec.5.4, P_{other} is ρ^* and T_{other} is $\bar{S}\cdot u/2$. If the time separation becomes D, the ACK private packet is dispatched. Then, T_{emp} is

$$T_{emp} = \frac{\int_0^D t\cdot\lambda_p\cdot\exp(-\lambda_p\cdot t)dt}{\int_0^D \lambda_p\cdot\exp(-\lambda_p\cdot t)dt} \qquad (5.42)$$

Hence, $T_{W1}^{(R)}$ is

$$T_{W1}^{(R)} = \frac{\bar{S}\cdot\rho^*\cdot u}{2} + \frac{1}{1-\exp(-\lambda_p\cdot D)}\cdot\{\frac{1}{\lambda_p} + \exp(-\lambda_p\cdot D)\cdot(D + \frac{1}{\lambda p})\} - \bar{S}\cdot u \qquad (5.43)$$

When an ACK is returned by an ACK private-packet, the waiting time $T_{W2}^{(R)}$ is similar to Eq.(5.34). In this case, we need not consider the situation that the ACK packet waits for transmitting packet on channel. Thus, $T_{W2}^{(R)}$ is given by

$$T_{W2}^{(R)} = \sum_{k=1} P_{ACK}(k)\cdot T_W'(k) \qquad (5.44)$$

Hence, $T_W^{(R)}$ is given by

$$T_W^{(R)} = (1-P_{AP})\cdot T_{W1}^{(R)} + P_{AP}\cdot T_{W2}^{(R)} \qquad (5.45)$$

Therefore, T_{res}, η_T, and T_c become as follows.

$$T_{res} = \bar{S}\cdot u + 2\cdot T_{cpu} + (1-P)\cdot T_c + P\cdot\{T_W^{(R)} + u\} \tag{5.46}$$

$$\eta_T = \frac{1}{L_1\cdot\{1 + \sum_{i\varepsilon P} E_{sm}(i)\cdot(i\cdot\gamma_i + P_{AP}\cdot\beta_i/\lambda_A\cdot D)\}} \tag{5.47}$$

$$T_c = \{3\cdot\max\{P\} - \bar{S}\}\cdot u + D + 2\cdot(s-1)\cdot T_{cpu} \tag{5.48}$$

5.6 Multi-ACK Methods with Waiting-Number-Control

In this section, the constraint on the ACK methods is that the waiting number of ACK signals reaches to the fixed number N.

5.6.1 ACK-private-Packet Fashion

When the waiting number of ACK signal is N, an ACK packet is dispatched to the sending node. The time $t_W^{(R)}$ is similar to Eq.(5.28). Then we derive t_W'. The occurence of an ACK signal is shown in Fig.5.7. We denote by t_i, the time when the i-th ACK is arrived after an ACK packet has returned. Further, x_i is defined as Eq.(5.31), $x_i=t_i-t_{i-1}$. The value of x_i has the exponential distribution with mean $1/\lambda_A$. Now we denote by τ_i the time separation between t_i and t_n. Then τ_i is the following.

$$\tau_i = t_n - t_i = x_{i+1} + x_{i+2} + \cdots + x_n \tag{5.49}$$

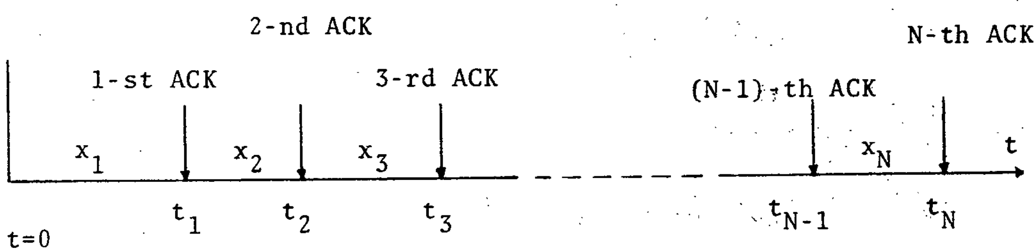


Fig. 5.7 The occurence of N ACK signals

This τ_i is the waiting time of the i-th packet and has the (n-i)-Erlang distribution with mean $(n-i)/\lambda_A$. We defined Y_i as the expected value of τ_i. Then, Y_i is given by

$$Y_i = \int_0^\infty \tau_i \cdot f_{erl}^{(i)}(\tau_i) d\tau_i$$

$$= (n-i)/\lambda_A \tag{5.50}$$

where $f_{erl}^{(i)}(x)$ is the i-Erlang distribution.

$$f_{erl}^{(i)}(x_i) = \frac{(\lambda_A)^i}{(i-1)!} \cdot x^{i-1} \cdot \exp(-\lambda_A \cdot x) \tag{5.51}$$

Hence, T_W becomes

$$T_W' = \frac{\sum Y_i}{n} = \frac{n-1}{2 \cdot \lambda_A} \tag{5.52}$$

The T_{res}, η_T, and T_c are given by

$$T_{res} = \bar{S} \cdot u + 2 \cdot T_{cpu} + (1-P) \cdot T_c + P\{\frac{n-1}{2 \cdot \lambda_A} + \rho^* \cdot (\frac{\bar{S} \cdot u}{2} + u)\} \tag{5.53}$$

$$\eta_T = \frac{1}{L_1 \cdot [1 + \sum_{i \epsilon P} E_{sm}(i) \cdot (i \cdot \gamma_i + \beta_i/n)]} \tag{5.54}$$

$$T_c = \{2 \cdot \max\{P\} + 1 - \bar{S}\} \cdot u + 2 \cdot (s-1) \cdot T_{cpu} + s \cdot \frac{n-1}{\lambda_A} \tag{5.55}$$

where β_i is similar in Sec.5.5 and s is the safety factor.

5.6.2 Piggy-back Fashion

The ACK signals are carried to the sending node N_T by the normal packet which is transmitted to the identical node N_T. When the waiting number of ACKs becomes N before the normal packet occurs, then an ACK packet which contains all ACK signals is dispatched. First, we derive the probability P_{AP} that an ACK packet is dispatched This is equal to the probability that no normal packet occurs before the n-th ACK signal occurs. The time separation between the arrivals of the first packet and the n-th packet, which is denoted by τ, has the (n-1)-Erlang distribution $f_{erl}^{(n-1)}(x)$ with mean $(n-1)/\lambda_A$. Hence, P_{An} is given by

$$P_{An} = \int_0^\infty f_{erl}^{(n-1)}(x)\cdot(1-\int_0^x \lambda_p\cdot\exp(-\lambda_p\cdot t)dt)dx$$
$$= \frac{\lambda_A^{n-1}}{(\lambda_A+\lambda_p)^{n-1}}\int_0^\infty \frac{(\lambda_A+\lambda_p)^{n-1}}{(n-2)!}\cdot x^{n-2}\cdot\exp\{-(\lambda_p+\lambda_A)x\}dx$$
$$= \left(\frac{\lambda_A}{\lambda_A+\lambda_p}\right)^{n-1} \tag{5.56}$$

In the case where an ACK piggy-backs to the normal packet, the waiting time of the ACK, $T_{W1}^{(R)}$ at the node N_R is given by Eq.(5.41). In this case, the time T_{emp} is

$$T_{emp} = \int_0^\infty \lambda_p\cdot t\cdot\exp(-\lambda_p\cdot t)dt-\bar{S}\cdot u = 1/\lambda_p-\bar{S}\cdot u \tag{5.57}$$

Hence $T_{W1}^{(R)}$ is

$$T_{W1}^{(R)} = \frac{\bar{S}\cdot\rho^*\cdot u}{2} + \frac{1}{\lambda p} - \bar{S}\cdot u \tag{5.58}$$

When an ACK is returned by ACK packet, the waiting time $T_{W2}^{(R)}$ is the same as the case of the ACK-private packet fashion. We denote by Y_i the average waiting time of the i-th ACK signal. Y_i is given by

$$Y_i = \int_0^\infty \lambda_p\cdot\exp(-\lambda_p\cdot t)dt\int_0^t x\cdot f_{erl}^{(n-i)}(x)dx$$
$$= \frac{n-i}{\lambda_A}\left(1-\lambda_p\sum_{j=1}^{n-i+1}\frac{\lambda_A^{j-1}}{(\lambda_A+\lambda_p)^j}\right) \tag{5.59}$$

In this case, there is no transmitting packet on channel. Thus, $T_{W2}^{(R)}$ is given by

$$T_{W2}^{(R)} = \frac{\sum_{i=1}^{n} Y_i}{n} + \frac{\bar{S}\cdot\rho^{*}\cdot u}{2} \tag{5.60}$$

Therefore, T_{res}, η_T, and T_c are given by

$$T_{res} = \bar{S}\cdot u + 2T_{cpu} + (1-P)\cdot T_c + P\{(1-P_{An})\cdot T_{W1}^{(R)} + P_{An}\cdot T_{W2}^{(R)}\} \tag{5.61}$$

$$\eta_T = \frac{1}{L_1\{1+\sum_{i\varepsilon P} E_{sm}(i)\cdot(i\cdot\gamma_i+\beta_i\cdot P_{An}/n)\}} \tag{5.62}$$

$$T_c = 3\max\{P\}+(s-1)\cdot\bar{S}\cdot u+2(s-1)T_{cpu} \tag{5.63}$$

5.7 Calculated Results of Multi-ACK Methods

In this section, we obtain the calculation results on the packet response time T_{res} and the transmission efficiency η_T and compare various ACK methods with them. In numerical results, the A-ACK method and the Single packeting method are utilized. Furthermore, let average message length be 1000 bits, channel capacity C 50 kbits/sec, L_1 0.5, L_2 0.04, L_3 0.01, and bit error rate q 10^{-6}.

Fig.5.8 and Fig.5.9 show the transmission efficiency η_T versus channel utilization factor ρ in the cases of the ACK-private packet fashion and the piggy-back fashion. Here, the case where N is equal to 1 means the Single ACK control method. Concerning to η_T, the multi-ACK methods are superior to the Single ACK method, without the condition that ρ is less than 0.2 in time-control ACK method. The time-control ACK method has bad characteristics in small ρ. This is the reason why in this method, an ACK packet is returned to the adjacent node whether any ACK signals exist or do not exist. In this point, the time-control ACK method must be improved.

So, we consider the constraint that if there exists no ACK signal, the ACK packet is not returned. In Figs.5.8 and 5.9, the dotted lines show this case. We can observe the good improvement. Furthermore, it can be seen that the piggy-back fashion is generally better than the ACK-private fashion.

Fig.5.10 and Fig.5.11 show tha packet response time versus channel utilization factor ρ in the cases of the ACK-private packet fashion and the piggy-back fashion. In the case of small value of ρ, the time-control method is considerably better than the waiting-number-control method.

Improved time-control-method is quite superior to the waiting-number-control, regarding to the packet transmission time. Furthermore, as well as the case of η_T, in the case of T_{res}, the piggy-back fashion is superior to ACK-private-packet fashion.

Therefore, we can conclude that, as long as η_T and T_{res} are concerned, the multi-ACK method with time-control and the piggy-back fashion is the most suitable to the packet-switching computer-communication network.

5.8 Conclusion

In this chapter, we discuss on the ACK method of the selective repeat ARQ to be suitable to the computer-communication network. We classify the various ACK methods and can exploit the time-control multi-ACK method as a result of classification.

Furthermore, as the evaluation measure, we introduce the transmession efficiency η_T described in Chapter 2 and define a new measure, the packet response time T_{res}. Concerning to η_T and T_{res}, we analyze the various ACK methods theoretically. Thus, calculating

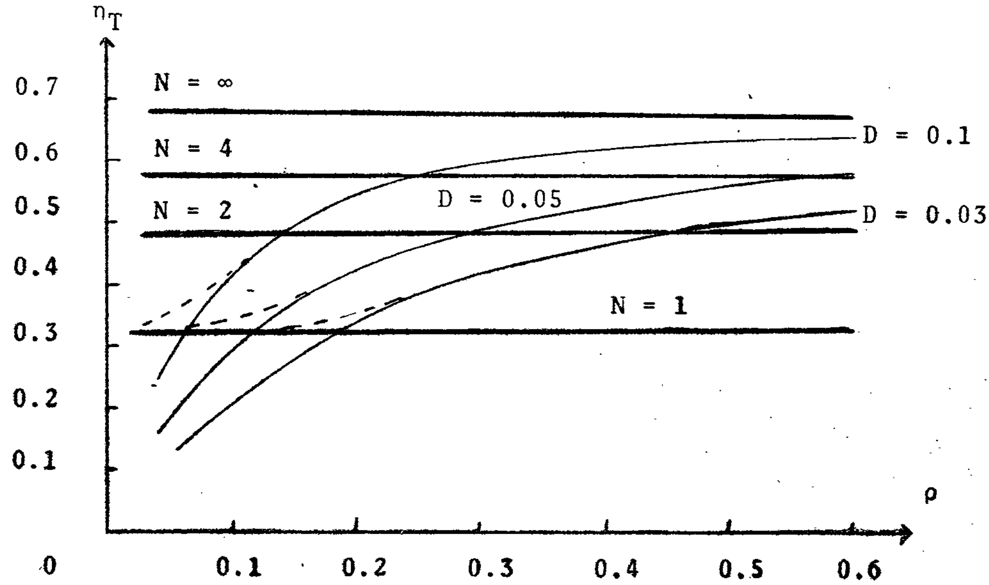


Fig. 5.8 Transmission efficiency versus channel utilization factor in the case of ACK private-packet fashion

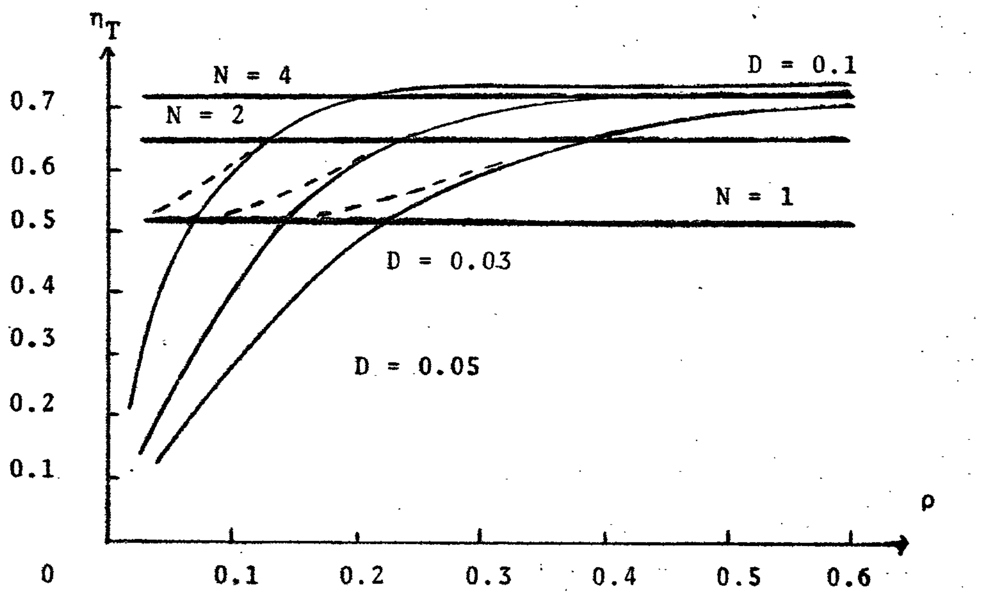


Fig. 5.9 Transmission efficiency versus channel utilization factor in the case of piggy-back fashion

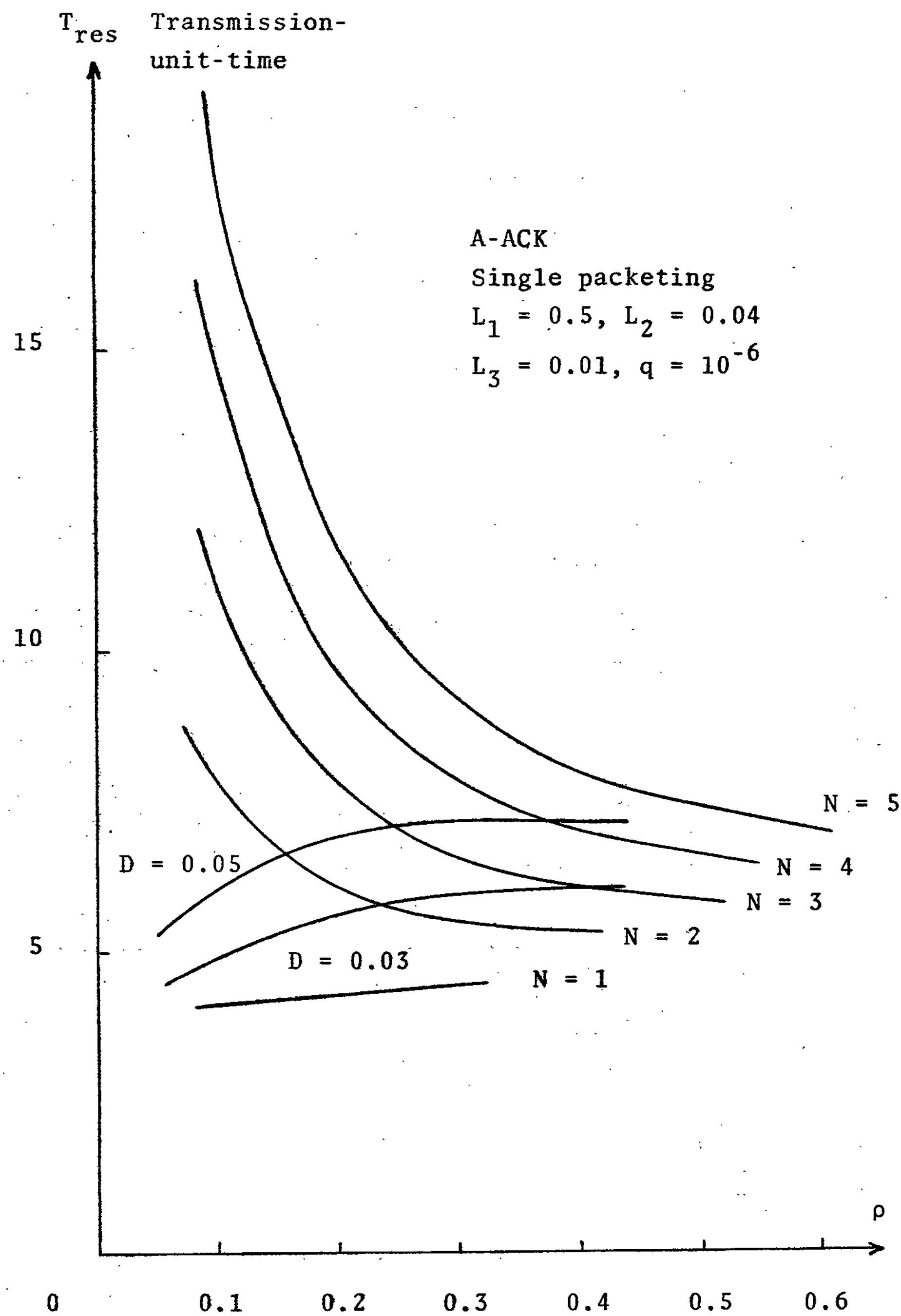


Fig. 5.10 The average packet response time versus channel utilization factor in the case of ACK private-packet fashion

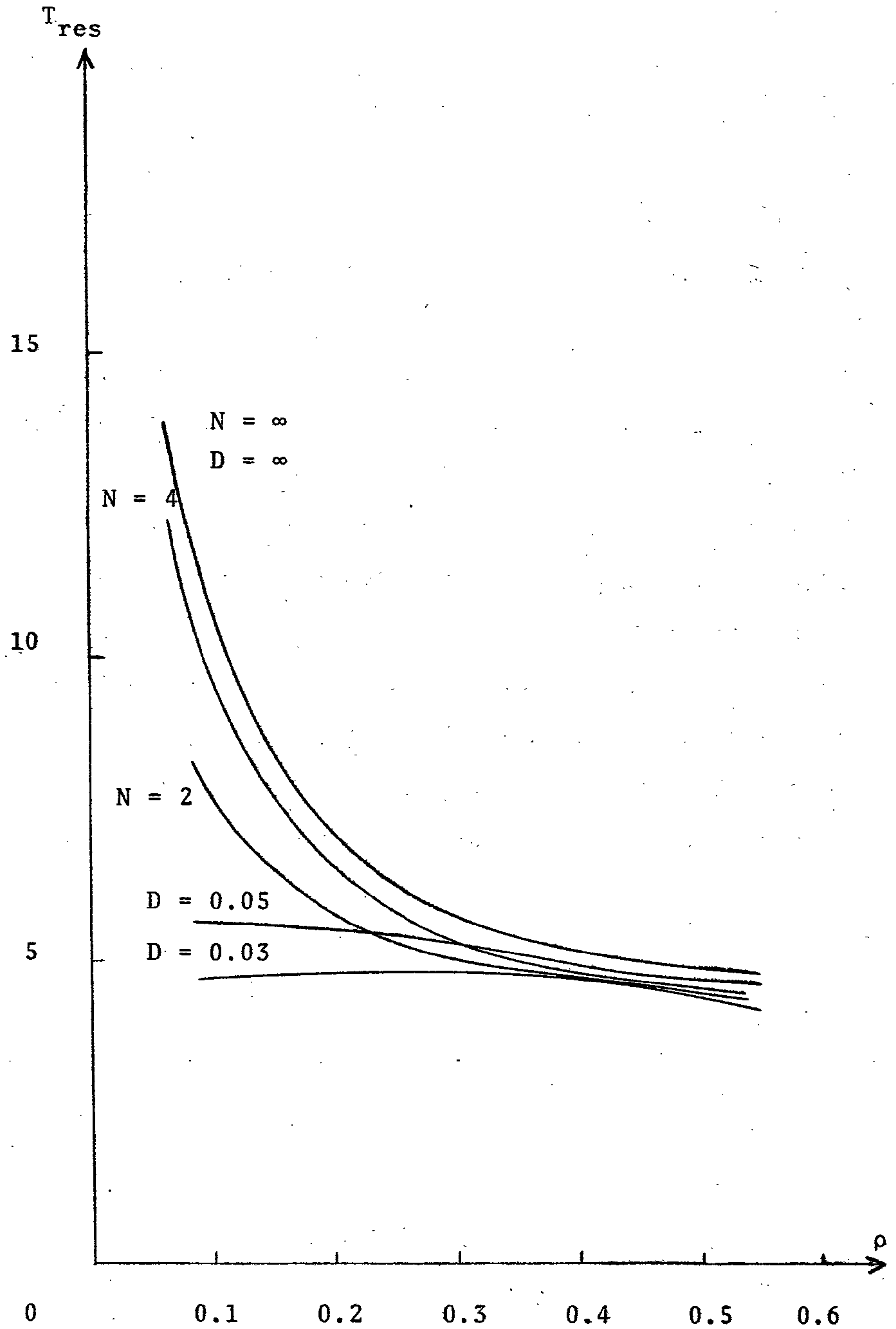


Fig. 5.11 The average packet response time **versus channel** utilization factor in the case of **piggy-back** fashion

them in the some practical case of channel utilization factor ρ, the numerical results are obtained.

Comparing the various ACK methods in regard to η_T and T_{res}, we can obtain some important results. We summarize them as follows.

1) The multi-ACK methods are generally superior to the single ACK methods.

2) In the multi-ACK, the piggy-back fashion is considered to have the effectiveness and to be better than the ACK-private packet fashion.

3) In the multi-ACK methods, the time-control is superior to the waiting-number control in the almost all the value of channel utilization factor ρ.

4) Therefore, we insist that the multi-ACK method with time-control and piggy-back fashion is the best ACK method in the computer-communicaion network as long as the transmission efficiency and the packet response time are concerned.

CHAPTER 6

ISARITHMIC FLOW CONTROL METHOD

6.1 Introduction

Congestion control problem is one of the most difficult and the most important problems in general communication network. In computer-communication network, since a message must traverse more rapidly and more reliablly, the requirement of congestion control method is urgent. Basically, this problem is classified into two classes. Those are:

1) Routing procedure
2) Flow control

The first deals how messages or packets are smoothly and rapidly transfered through the network. Thus, by finding a good path from the source to the destination (of messages or packets), the communication systems are prevented from over-congestion. The algorithm that finds or selects a good route for each message or packet is the routing procedure.

The second deals with the methods that the networks are prevented from the over-congestion by limiting the amount of the input-traffic. In this problem, the difficulties are how to find the messages that will cause to over-congestion and how to halt these messages out of the computer network without serious delay-losses.

In this chapter, we discuss on the latter method, i.e. flow control. There are three principal methods on flow control. Those are:

1) End-to-end link control
2) Local control
3) Centralized control

In the first method, the number of links between any pair of nodes in the network is restricted. But, the data rate of link is different each other. Thus, when the data rate is quite fluctuated according to each message, this method is considered to be disadvantageous. In the second method, neighboring nodes inform the state of congestion of its own node for each other. But, since the informations of remote-located nodes are much delayed, rapid effect for congestion control cannot be obtained. In the third method, the network has the information-center which can instruct the situation of the congestion. This center makes each node halt, permit and reject the input-messages. But the control information of this center must be updated with the advance of time. This updating information may newly cause the another over-congestion.

In 1972, D.W.Davies presented the Isarithmic flow control methods. In this method, amount of packets less than constant value can be simultaneously admitted into the network, introducing the concept of permits. On this Isarithmic networks, W.L.Price has investigated many problems by computer simulation. But, there is no theoretical analysis on this method.

In this chapter, we try to study theoretically on the behaviors of one arbitrary node in the network. Thus, we represent a node model as duplicated queueing system and analyze it. Furthermore, concerning to this method, some improvements and their advantages are presented.

6.2 Concept of Isarithmic Network

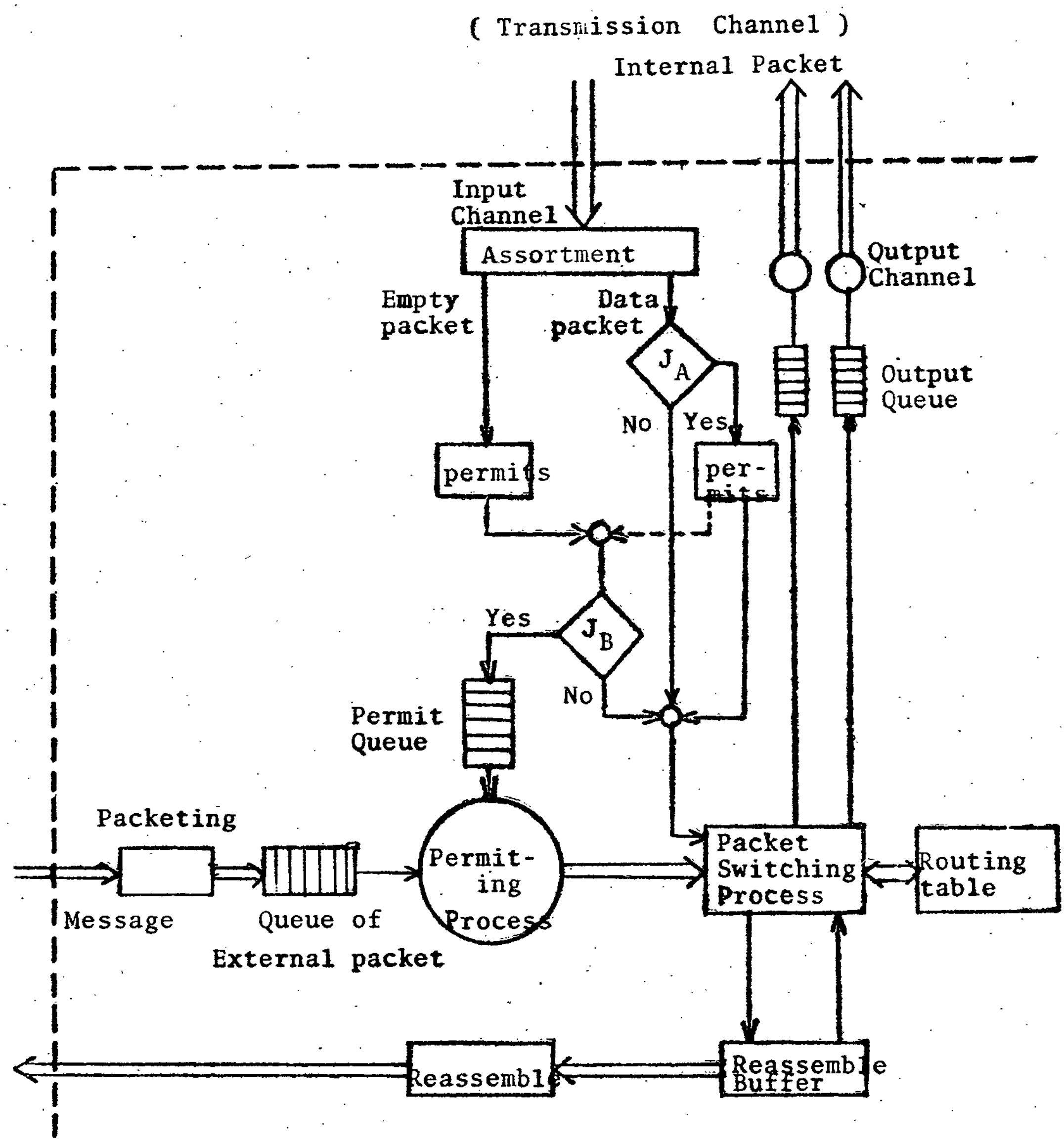
In this method, a message is divided into some packets of constant length at the source node. Thus, the Single packeting method, which is described in Chapter 2, is utilized. Further, the total number of packets in the network is kept as the value less than constant value. Since the traffic in the network is strictly restrained under the total capacity of network, the occurence of fatal-congestion cannot be existed, although slight congestions may locally occur.

A constant number of permits are issued in the network. When a message comes in the network, it is divided into some packets at the source node. Hereafter, we define these packets as external packets. Then, among these external packets, the packets which obtain a permit can be admitted in the network. The packets that have permits and are traversing in the network are defined as internal packets. The remaining packets which cannot obtain permits must wait at this node till they get permits. On the other hand, permits are cruising in the network for waiting external packets as well as taxi for fares. When permit finds a waiting external packet, it makes that packet admit in the network. Then, this packet can traverse in the network and reach the destination node. At this point, a permit is released from this packet, and it is cruising again.

There are two types of the internal packets in the Isarithmic network. Those are

1) Data packet
2) Empty packet

The first indicates the normal packet, including ACK-packet, Control-packet, and so on. This packet always has one permit and releases it at the destination node. The second has no information (i.e. message data and control information), and only carries several permits. Each node has the permit buffer that some permits can be stored for external packets. The size of this buffer is finite and is usually chosen as small value. If this permit-buffer is not full, arrival permits are stored in it. But, when excess permits arrive, they cannot be stored and continue to cruise in the network. In this case, the empty packet is utilized to carry these permits.



Where: J_A means whether this node is the destination of this packet.

J_B means whether permit buffer is full.

Fig. 6.1 The behavior of switching node in the Isarithmic network

Thus, we represent the behavior of each node in the Isarithmic network in Fig.6.1. As shown in Fig.6.1, we define the process that an external packet obtains a permit, as the permitting process. In each node, there are five types of queues. Those are:

1) Output queues on each transmission channel
2) Queue for the CPU service
3) Reassemble queue
4) Queue of external packet
5) Permit queue

The 1), 2), and 3) are discussed in Chapter 4. The queue of external packet is composed of the external packets that cannot obtain permit at arrival time and wait for it. On the other hand, the permit queue is composed of the permits that can be stored in the permit-buffer and wait for the external packet.

The permitting process is corresponding to these two queues. Furthermore, we pursue the analysis of permitting process and the behaviors of these two queues.

6.3 Analysis of Permitting Process

6.3.1 Duplicated Queueing Model with Permits and External Packets

In the Isarithmic packet-switching network, the message traverse time T_M is given by

$$T_M = T_p^{(s)} + \sum_{i \in Path\{N_S, N_D\}} (T_{cpu}^{(i)} + T_T^{(i)}) + T_{cpu}^{(d)} + T_R^{(d)} \quad (6.1)$$

We have already discussed on $T_{cpu}^{(i)}$, $T_T^{(i)}$, and $T_R^{(d)}$ in Chapter 4 and those can be adopted in this chapter. In Eq.(6.1), only the first term $T_p^{(s)}$ is unique for us. This means the time required for each external packet to obtain a permit after it arrived at the source node N_S. We try to obtain this $T_p^{(s)}$ theoretically, in this section.

The time T_p is the average time in the queue of the external packet of the duplicated queueing model as shown in Fig.6.2. Furthermore, the following assumptions are introduced.

6.1) The arrival of the external packet has the Poisson distribution with mean λ_o per unit time.

6.2) The arrival of the internal packet at a node from the neighboring nodes has the Poisson distribution with mean λ_{in} per unit time.

6.3) The buffer for the external packets has infinite size.

6.4) The node can store M permits. If the number of permits in the node exceeds M, the excess permits are transfered to the adjacent nodes with the empty packet.

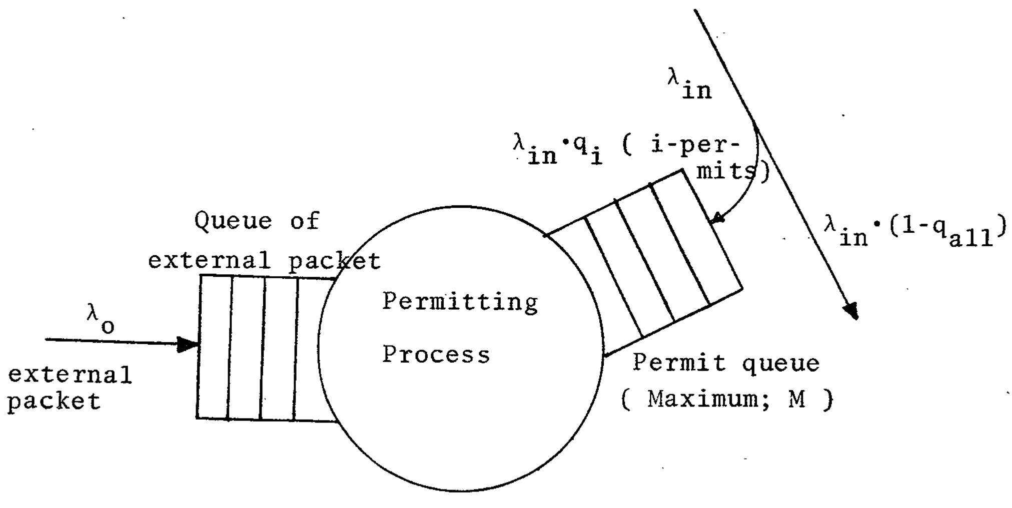


Fig. 6.2 Model of permitting process

6.5) The internal packet has k-permits with the probability q_k (k= 0, 1, 2, ... , S). Thus the maximum value of permits that one internal packet can carry is S.

6.6) The service time of the CPU is much less than the waiting time of an external packet in the permitting process. Thus, it can be neglected.

6.3.2 Analysis of Duplicated Queueing Model

From the assumptions, both the external packet and the internal packet have the Poisson arrival, the probability that one external packet and internal packet arrive at the node during quite minute time Δt are

$$\begin{aligned} &\lambda_o \cdot \Delta t + O(\Delta t) && \text{for external packet} \\ &\lambda_{in} \cdot \Delta t + O(\Delta t) && \text{for internal packet} \end{aligned} \qquad (6.2)$$

where $O(\Delta t)/\Delta t$ tends to 0 as Δt tends to 0.
The probability that more than two packets arrive during Δt is $\lambda_{in} \cdot \lambda_o \cdot (\Delta t)^2 + O(\Delta t)$ and is equal to $O(\Delta t)$. Thus, during Δt, more than two arrivals cannot occur.

Then we define the state of this permitting process. The node can store the maximum of M permits, simultaneously. We represent this state as E_0. When the node stores i permits ($1 \leq i \leq M$), this is represented as E_{M-i} In these states (i.e. E_0, E_1, ... , E_{M-1}), an external packet can obtain a permit, as soon as it arrives at the node. Hence, there is no waiting of the external packets. The state E_M means that there is no permit and no external packet in this permitting process. While the node does not store any permit, the arriving external packets wait for the permits. When i external packets wait for permits, we represent this state as E_{M+i}. Therefore, we can define the state E_i ($i=0,1,2,\ldots,M,\ldots$). On condition that the state of the permitting process is E_n at time $t+\Delta t$, we can assume that one of the following three events occured during Δt. Those are:

1) Event A : The state is E_n at time t and there is no arrival of a packet during Δt.

2) Event B : The state is E_{n-1} at time t and one external packet arrives during Δt.

3) Event C : The state is E_{n+k} at time t and one internal packet with k permits arrives during Δt $(k=1, 2, \ldots, S)$.

In Fig.6.3, we represent the state transition diagram. Here, concerning to the event C, we must add the following special case. That is:

When n is equal to 0, there is no lower state than E_0. Thus, in this case, we must modify the event C as follows. The state is E_i at time t and one internal packet with k permits arrives during Δt, where $i=1, 2, \ldots, S$ and $k=i, i+1, \ldots, S$.

From the state transition diagram in Fig.6.3, the state equations are given by

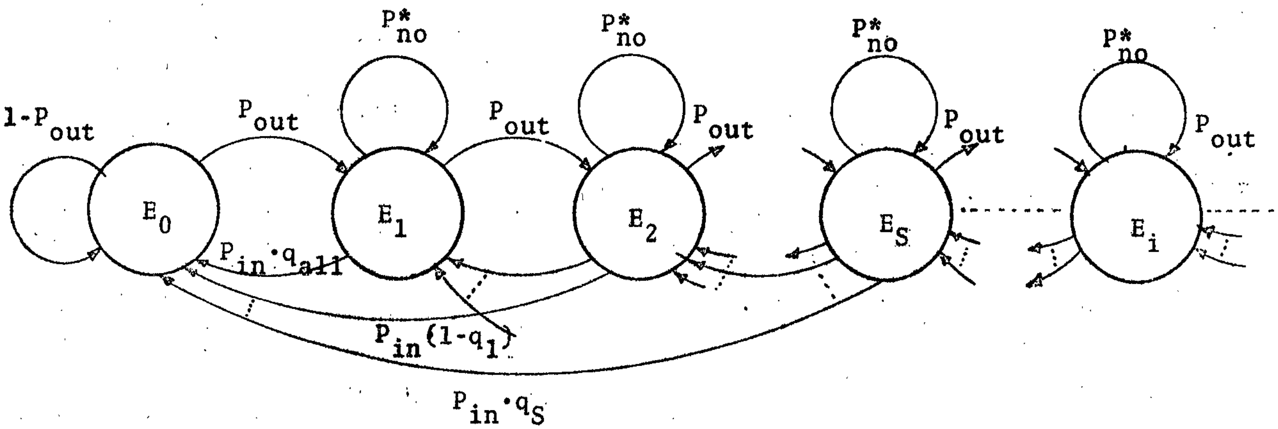
$$P_0(t+\Delta t) = P_0(t)(1-P_{out})+\sum_{i=1}^{S} P_i(t)\cdot(1-\sum_{k=0}^{i-1} q_k)\cdot P_{in}$$

$$P_n(t+\Delta t) = P_n(t)\cdot(1-P_{out}-q_{all}\cdot P_{in})+P_{n-1}(t)\cdot P_{out} + \sum_{k=1}^{S} P_{n+k}(t)\cdot P_{in}\cdot q_k \qquad (6.3)$$

where $P_i(t)$ is the probability that the state is E_i at time t,

$P_{in} = \lambda_{in}\cdot\Delta t + 0(\Delta t)$,

$P_{out} = \lambda_o\cdot\Delta t + 0(\Delta t)$,



Where : $P^*_{no} = 1 - P_{out} - q_{all} \cdot P_{in}$

Fig. 6.3 State transition diagram

and $$q_{all} = \sum_{i=1}^{S} q_i = 1 - q_0 .$$

When we make Δt tend to 0, the state equation is derived as follows.

$$P_0'(t) = \sum_{i=1}^{S} P_i(t) \cdot (1 - \sum_{k=0}^{i-1} q_k) \cdot \lambda_{in}$$

$$P_n'(t) = P_{n-1}(t) \cdot \lambda_o + \sum_{k=1}^{S} P_{n+k}(t) \cdot \lambda_{in} \cdot q_k - P_n(t) \cdot (\lambda_o + q_{all} \cdot \lambda_{in}) \qquad (6.4)$$

Under the condition that the traffic-processing-rate for the external packet defined as ρ_E, is less than 1, we can obtain the steady-state equation, where ρ_E is

$$\rho_E = \frac{\lambda_o}{\lambda_{in} \sum_{i=1}^{S} i \cdot q_i} \qquad (6.5)$$

The steady-state equation is

$$\sum_{i=1}^{S} P_i (1 - \sum_{k=0}^{i-1} q_k) \cdot \lambda_{in} - P_0 \cdot \lambda o = 0$$

$$P_{n-1} \cdot \lambda_o + \sum_{k=1}^{S} P_{n+k} \cdot \lambda_{in} \cdot q_k - P_n (\lambda_o - q_{all} \cdot \lambda_{in}) = 0 \qquad (6.6)$$

where P_i is the probability that the state is E_i. Here, we introduce Z_n defined as follows.

$$Z_n = \sum_{i=1}^{S} P_{i+n} (1 - \sum_{k=0}^{i-1} q_k) \cdot \lambda_{in} - P_n \cdot \lambda_o \qquad (6.7)$$

Then,

$$Z_0 = 0 \qquad (6.8)$$

$$Z_{n+1}-Z_n = \lambda_o\cdot P_n - P_{n+1}(q_{all}\cdot\lambda_{in}+\lambda_o) + i(\sum_{k=0}^{S} P_{n+k+1}\cdot q_k) = 0 \qquad (6\cdot9)$$

Thus,

$$Z_n = 0 \qquad (6.10)$$

In the case where S is equal to 1, the state probability P_i is easily obtained. Eq.(6.10) is simplified as follows.

$$P_{n+1}\cdot\lambda_{in}(1-q_1) = 0 \qquad (6.11)$$

Hence, the state probability P_n is

$$P_n = \frac{\lambda_o}{\lambda_{in}\cdot q_{all}}\cdot P_{n-1} = \left(\frac{\lambda_o}{\lambda_{in}\cdot q_{all}}\right)^2 P_{n-2} = \cdots\cdots\cdots\cdots = \rho_E^n\cdot P_0 \qquad (6.12)$$

For these state probabilities, the following constraint is obtained.

$$\sum_{i=0}^{\infty} P_i = 1 \qquad (6.13)$$

Therefore,

$$P_0 = 1-\rho_E$$

$$P_n = (1-\rho_E)\cdot\rho_E^n \qquad (6.14)$$

6.3.3 Calculated Results of the Expected Waiting Time

In this section, we obtain the expected waiting time of the external packets and the permits. The former may imply the effectiveness of congestion control. For, if these external packet is admitted without any buffering-effect, the network may suffer the over-congestion. The latter shows the excess-control of this method. The average total time of stored permits in the whole network represents the overhead of the network throughput capacity. At first, we will obtain the expected waiting time of the external packets in the queue of permitting process. The average number of the external packets in the queue L_{out} is given by

$$L_{out} = \sum_{k=M+1}^{\infty} P_k \cdot (k-M)$$

$$= \frac{\rho_E^{M+1}}{1-\rho_E} \qquad (6.15)$$

From Little's Formula, W_{out} becomes

$$W_{out} = \frac{L_{out}}{\lambda_o} = \frac{\rho_E^M}{\lambda_{in} \cdot q_{all} \cdot (1-\rho_E)} \qquad (6.16)$$

Then, we obtain the expected waiting time of permit W_{per}. The average number of the permits in the permitting process L_{per} is

$$L_{per} = \sum_{k=0}^{M-1} (M-k) \cdot P_k$$

$$= m - \frac{\rho_E (1-\rho_E^m)}{1-\rho_E} \qquad (6.17)$$

Thus, W_{per} is

$$W_{per} = \frac{L_{per}}{\lambda_{in} \cdot q_{all}} \qquad (6.18)$$

Fig.6.4 shows the characteristics of the average number of permitting process L_{out} and L_{per} versus M, the maximum value of permits that one node can store. Fig.6.5 (a) and (b) show the characteristics of the average waiting time of permitting process W_{out} and W_{per} versus M. For small value of M, L_{per} and W_{per} are quite small, but L_{out} and W_{out} are very large. The more M increases, the larger L_{per} and W_{per} become and the smaller L_{out} and W_{out} become. Both characteristics are crossing at the middle value range of M.

Fig.6.6 shows the average waiting time W_{out} and W_{per} versus the traffic-processing-rate ρ_E. The time W_{out} is quite small for $\rho_E<0.5$ and tends to infinity as ρ_E tends to 1. On the other hand, W_{per} is 0 at both $\rho_E=0.0$ and $\rho_E=1.0$, and has the maximum value at $\rho_E=0.5$ for each M. The reasons are the following:

At $\rho_E=1$, the arrival rate of the external packet is equal to one of the permits. The buffer of the external packet is infinite, but the permits can be stored to the value of M at most. Hence, the waiting time of permit is 0. At $\rho_E=0$, the arrival of the external packets is zero or the one of permits is infinite. Thus, the buffer

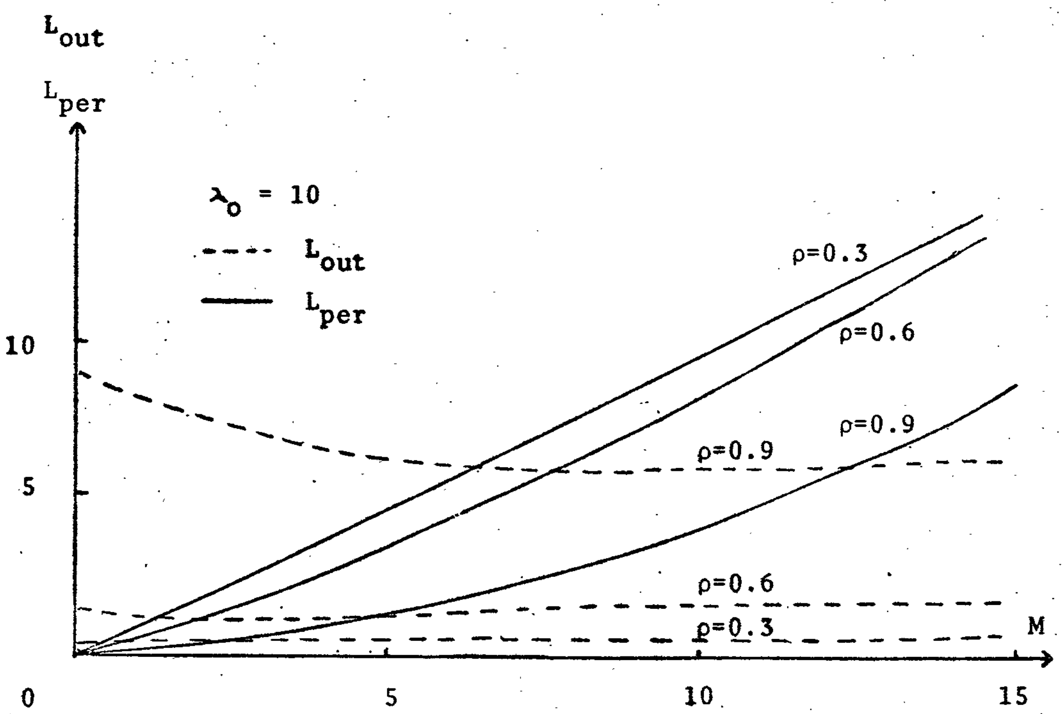


Fig. 6.4 The average waiting number of permitting process for permits and external packets

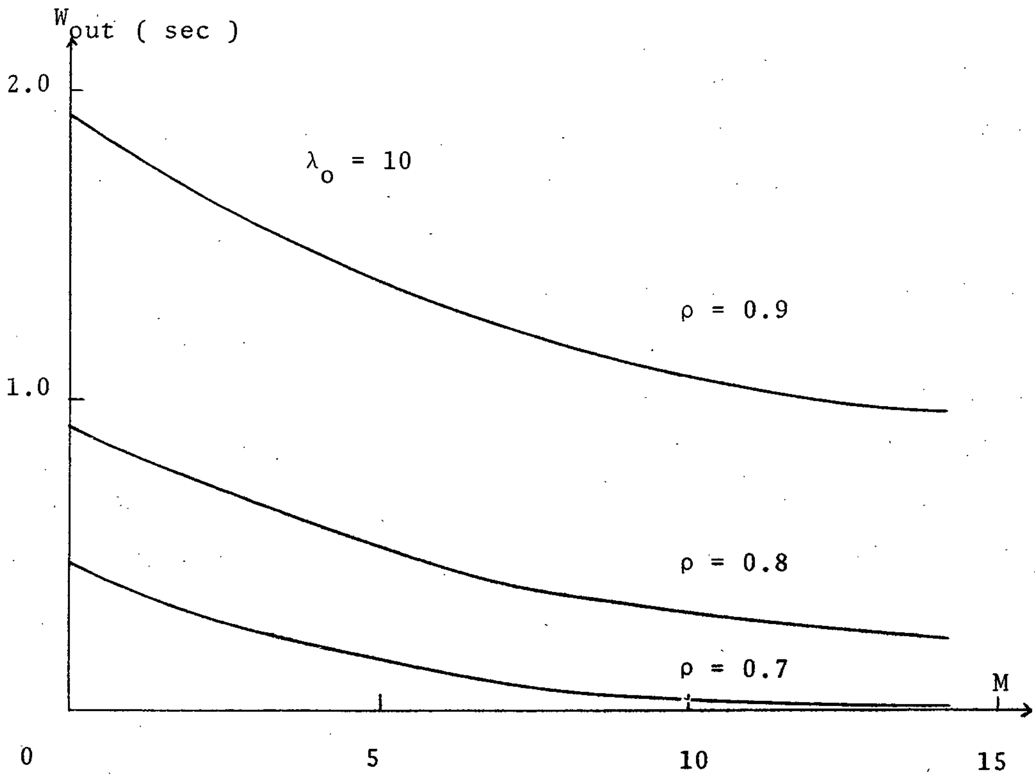


Fig. 6.5(a) The average waiting time of external packet

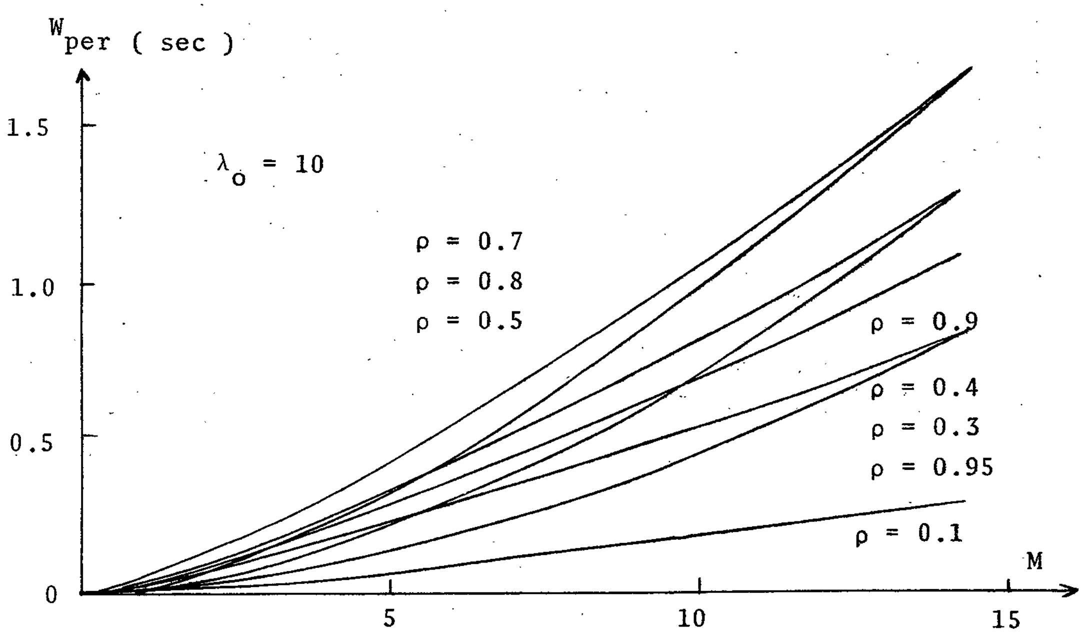


Fig. 6.5(b) The average waiting time of permit

of permits is full at once and almost all permits except only M mermits are transmitted to the adjacent node with no waiting time for the external packet. Therefore, the average waiting time of permits is nearly equal to 0.

Fig.6.7 shows the optimal value of M versus ρ_E. We define optimal M as the value of M that makes the summation $W=\alpha \cdot W_{per} + \beta \cdot W_{out}$ be the smallest. Here, we choose $\alpha=\beta=1$. Thus, given ρ_E, we can design the number of permits that can be stored in the node.

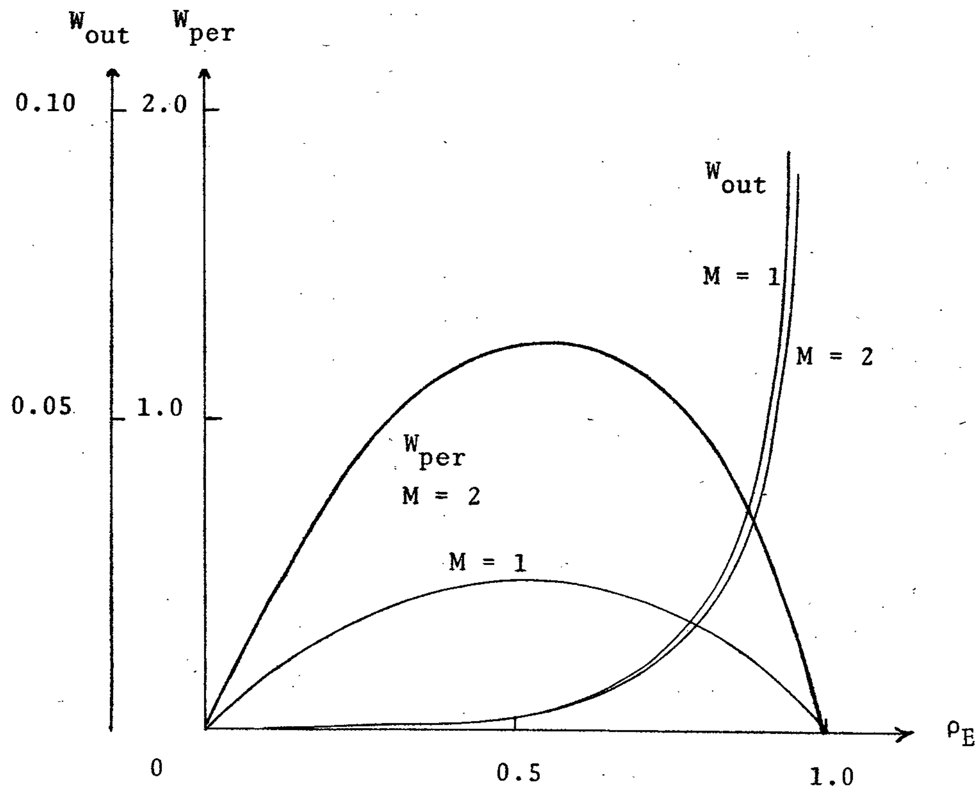


Fig. 6.6 The average waiting time of external packet and permit versus channel utilization factor

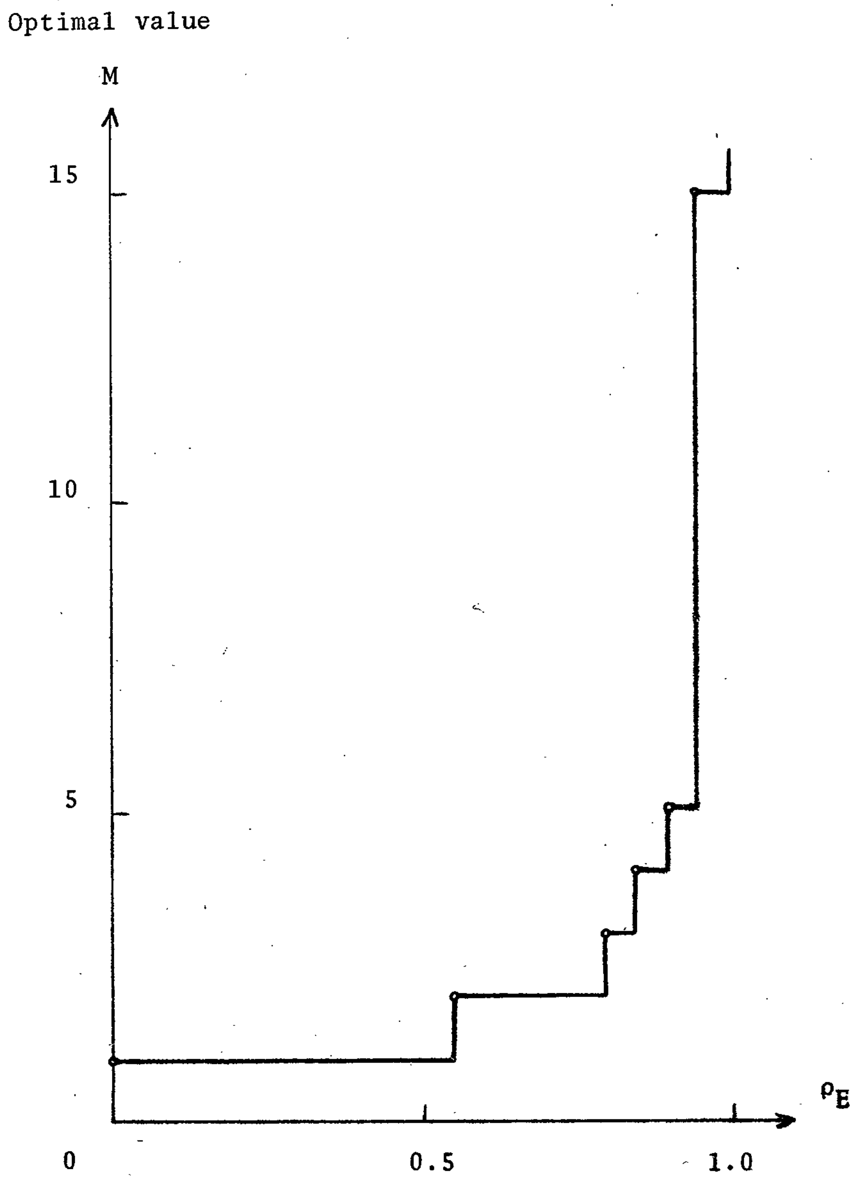


Fig. 6.7 The optimal value M for summation of W_{out} and W_{per} versus channel utilization factor

6.4 Improvement of Isarithmic Method

In the Isarithmic network, the external packet cannot be admitted in the network, before it obtains a permit. Thus, an external packet with no permit cannot be transmitted to the next node, even if the transmission channel is empty. This state makes the channel utilization be low value. Among these external packets which do not have a permit, the packets that destine the adjacent node may stay in the network for a short time. Thus, these packets cannot cause the network over-congestion, if they are admitted in the network without a permit, only when the channel is empty.

Therefore, we introduce the following improvement. That is:

When a transmission channel to the adjacent node is empty, an external packet which destines to the same node can be admitted without a permit. In this section, we analyze the effectiveness of this improvement. Thus, we represent this process as in Fig.6.8. We consider one output channel of the switching node. One internal packet has one permit with probability q_{all}. For simplicity, the node cannot store the permit. So, when there is no waiting external packet, the permit is transfered to the next node at once. Among the external packets, there are some packets which destine to the adjacent node with probability P. When the transmission channel is empty. We assume that this channel is slotted by one-packet-length and each packet is transmitted in each time slot.

We represent by $P_k^{(n)}$, the probability that at time n, the number of the waiting packets and the transmitting packets in the output queue of this transmission channel is k. Both the external packet and the internal packet have the Poisson arrival. Hence, the probability $P_0^{(n+1)}$ is given by

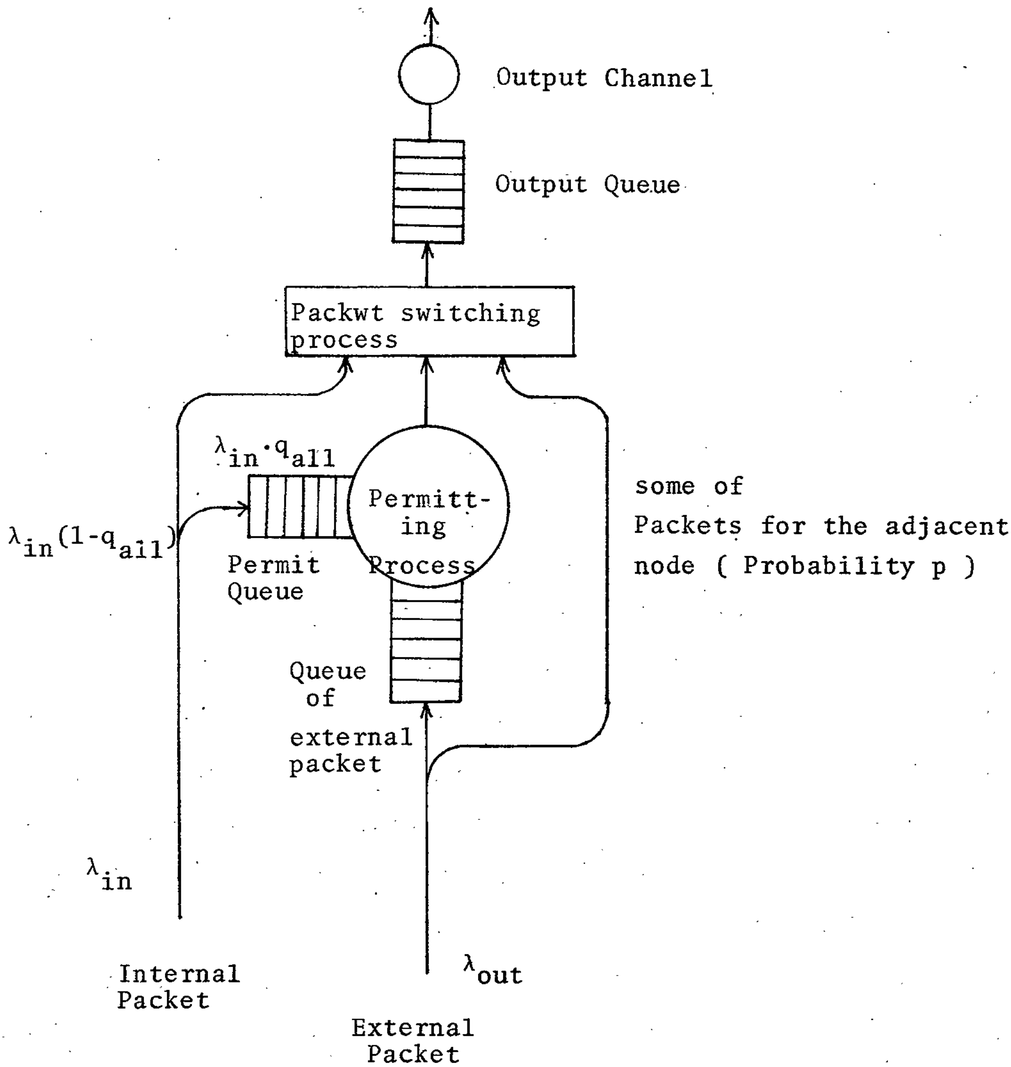


Fig. 6.8 Model of permitting process in the improved Isarithmic network

$$P_0^{(n+1)} = \{P_0^{(n)}+P_1^{(n)}\}\cdot Pois(0 / \lambda_o)\cdot Pois(0 / \lambda_o\cdot p) \qquad (6.19)$$

where Pois(A/B) is the probability that the number of arrival is A, under the condition that the mean arrival rate is B per unit time. Thus, in general, the probability $P_k^{(n+1)}$ is given by

$$P_k^{(n+1)} = \{P_o^{(n)}+P_1^{(n)}\} Pois(k / \lambda_{in}) Pois(0 / \lambda_o\cdot p)$$
$$+Pois(k-1 / \lambda_{in})\cdot\{1-Pois(0 / \lambda_o\cdot p)\}$$
$$+\sum_{i=2}^{k+1} P_i^{(n)}\cdot Pois(k-i+1 / \lambda_{in}) \qquad (6.20)$$

If the utilization factor of this channel is less than 1, we can obtain the steady-state-probability. We define this as P_i.

$$\lim_{n\to\infty} P_k^{(n)} = P_k \qquad (6.21)$$

$$\sum_{k=1}^{\infty} P_k = 1 \qquad (6.22)$$

Therefore, we can obtain the numerical value of the probability P_0 and the number of packet which can go through the permitting process per unit time, defined as λ_o'.

$$\lambda_o' = (P_0+P_1)\cdot p\cdot\lambda_o \qquad (6.23)$$

Then, the arrival rate of the external packet to the queue of the external packet changes and new arrival rate λ_o^* is

$$\lambda_o^* = \lambda_o-\lambda_o' = \lambda_o\{1-(P_0+P_1)\cdot p\} \qquad (6.24)$$

From Eq.(6.15), the average number of the external packets in the queue of the permitting process L^*_{out} is

$$L^*_{out} = \frac{\rho^*}{1+\rho^*} \tag{6.25}$$

where

$$\rho^* = \lambda^*_o/\lambda_{in} \cdot q_{all}$$

The improvement factor τ, which is defined as the ratio of the difference between L_{out} and L^*_{out} to L_{out} is

$$\tau = 1 - \frac{L^*_{out}}{L_{out}} = 1 - \frac{\rho^*(1-\rho)}{\rho(1-\rho^*)} \tag{6.26}$$

Fig.6.9 shows the improvement factor τ versus traffic-processing-

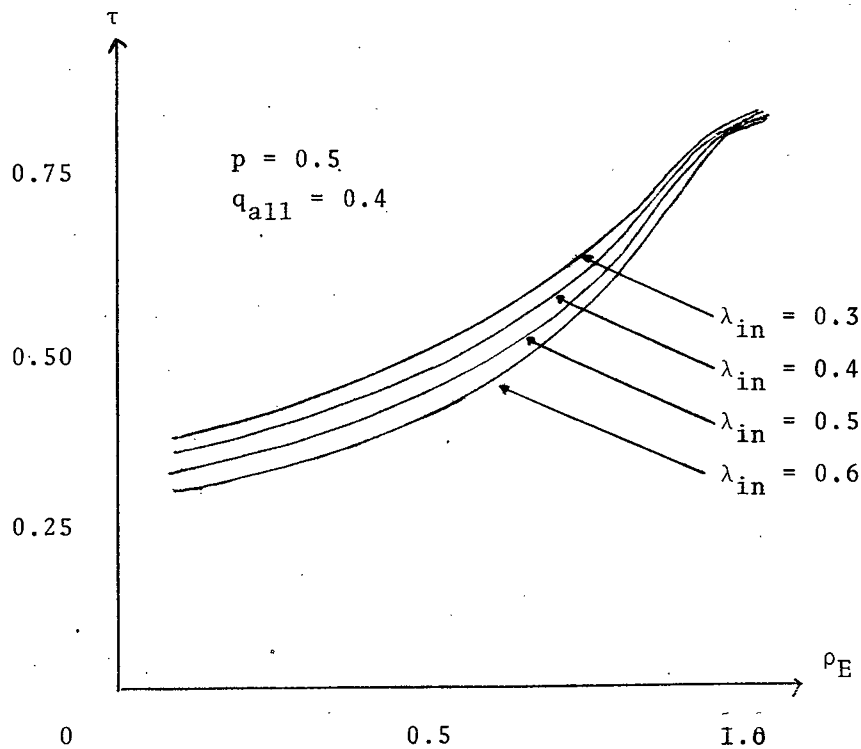


Fig. 6.9 The degree of improvement versus traffic-processing-rate

rate ρ_E. Introducing this means, L_{out} is improved more than 40% at ρ_E=0.5, without any serious congestions.

6.5 Conclusion

The Isarithmic flow control method is one of the most interesting methods in regard to keeping the packet switching network from over-congestion. In this chapter, we represent the switching node in the Isarithmic network as the duplicated queueing model and analyze this model theoretically. As a result of this analysis, we can derive the average waiting time of the external packet and the permit. We assume that the former implies the effectiveness of the congestion-control and the latter, the excess control. We show the numerical results of those characteristics. Further, utilizing the waiting time of the external packet, we can obtain the message traverse time in the Isarithmic network.

This Isarithmic method has the obvious disadvantage, i.e. low throughput. This is the reason why the external packet with no permit cannot be transmitted, even if the transmission channel is empty. Thus, we present the one method of improvement so that the external packet which destines to the adjacent node can be transmitted without a permit, only when the transmission channel is free. Further, we can analyze this improved Isarithmic network with Imbedded Markov chain and obtain the extents of this improvement. From this analysis, we insist that this improvement makes the throughput of the network quite large, without any serious congestions.

CHAPTER 7

BLOCK SWITCHING METHOD

7.1 Introduction

So far we have been discussing on the packet switching for computer-communication networks. As discussed in the previous chapters, the packet switching is most advantageous for a traffic in the computer-communication network, where messages have short length as an average and frequently occur. But, when the messages are very long such as data-file or still-picture transmission, the direct switching or the time-division-multiplexing line switching is superior to the packet switching. Further, the tranffic characteristics of the computer-communication network are widely fluctuating according to the time of day, the day of week, the week of month, season, and geographical area etc. Thus, it seems unfavorable that only one switching method whether it is packet switching or line switching, handles the whole of network traffic. Therefore, we may insist that in the computer-communication network, the hybrid switching system which has both the packet-switching function and the line-switching function is more advantageous

The hybrid switching networks are generally classified into three types. Those are:

1) Independent parallel type
2) Dependent parallel type
3) Compound type

In the first type, the network is duplicately composed of two independent subnetworks of the packet-switching and the line-switching. Two subnetworks are connected with each other through the interface at every node. So, network user can choose one between two switching systems, according to his message length. In realization of this system, there is not any problem except the interface of the subnetworks. But, since it must be composed of two independent networks, it is quite costly and has low system efficiency.

In the second type, the network is constructed with both functions and is operated in one of two functions. The operation of the network can exchange from one function to another. Thus, the network chooses the switching function between two options according to the characteristics of the input traffic at each time. In this case, when the whole network is operated in only one switching function, this hybrid method can not be so efficient. But, when the network is composed of several independent channels, which can be operate in any function, and when the network operator can vary the rate of the number of line-switching channels to the number of packet-switching channels according to the characteristics of input-traffic, this network may have superiority to the first type. Some researchers have investigated this type of the hybrid switching systems and confirmed the effectiveness of this system. But few have described the realization of such methods. Even if these systems can be realized, they will be quite complicated.

In the third type, the network has both functions and can be operated in both switching modes, simultaneously. Thus, network users utilize one function of two optional switching functions according to his message length, at any time.

This chapter deals with the third type, i.e. the compound hybrid-switching system. We present the realizable method of compound-type hybrid-switching and define this as *block switching*. This method is based upon the slotted-packet-switching, where the transmission channel is slotted into one-packet-length and a packet is sent on a time slot. Further, this method has a new ability of reservation of time slots in addition to the functions of the slotted packet switching. We describe the general concept and the composition of the block switching system, and discuss on the superiority of this system to other switching systems.

7.2 Basic Concept of Block Switching System

In the block switching system, the Single packeting is utilized. So, all packets in the network have the same length (i.e. one-segment-length). Further, all transmission channel in the network is slotted by this one-packet-length, and a packet is transmitted on this time slot by timing-controller.

There are simultaneously two transmission modes in this system. Those are:

a) Store-and-Forward mode (S/F mode)

b) Reserved block mode (R/B mode)

The S/F mode is similar to the normal packet switching. Thus, a message is divided into several packets in the source node and each packet independently traverses through the network on Store-and-Forward way, without setting the end-to-end link before the message is transmitted.

On the other hand, the R/B mode corresponds to the line switching. That is the reason why, in this mode, a message sender must connect the end-to-end link with a message receiver before he transfers this message, as well as in the line switching. The link connection means the reservation of adequate number of time-slots (we call time slot as block, hereafter) at each channel on the linked path. When link is completely connected, the message sender transfers the message to the source node. At the source node, this message is transformed into several packets and these packets are transmitted on Store-and-Forward way in the network.

Corresponding to two transmission modes, there are four types of packets in the network as shown in Fig.7.1. Those are:

1) S/F message-packet

2) R/B link-packet

3) R/B message-packet

4) ACK packet

The S/F message-packet is utilized to transmit message-bits on S/F mode. On the other hand, when a network user wants to transmit a message on R/B mode, he must connect the end-to-end link to his receiver by the R/B link-packet, at first. After connecting link, he can transmit this message, utilizing the R/B message-packet. The ACK-packet is utilized to exchange acknowledgments and control informations between nodes. The first and the fourth packets are transmitted on S/F mode, while the second and the third on R/B mode.

Now, we describe the actual operation of the block switching network. As shown in Fig.7.2, a network user can choose one between two optional modes, according to his message length. If his message length is short, he will choose the S/F mode transmission. He types in the message from his terminal console, indicating his receiver and a transmission mode, where his receiver may be a large computer or a terminal console in remote-location. Further, sometimes the sender may be the large computer. The message is handled by the network as in the case of packet switching. When the message is correctly delivered to the receiving node, a message-ACK is returned to the source node by ACK-packet and the node transfers it to the source terminal. By this ACK, he knows that his message is completely sent to his destination.

On the other hand, if his message is quite long, the S/F mode transmission is considered to be disadvantageous way. Thus, he must

a) S/F message-packet

ID	AD	MN	CI	PN	TEXT (message bits)	CH

b) R/B link-packet

ID	AD	MN	CI	LN	Additional Information

c) R/B message-packet

ID	AD	MN	TEXT (message bits)	SN

d) ACK packet

ID	AD	MN	CI	ACK	Additional Information	CH

ID : Identification of packet

MN : Message number

CI : Control information

PN : Packet number

AD : Address information

LN : Link number

CH : Check bits for channel error detection

SN : Sequence number of packets of R/B mode

ACK : Acknowledgement information

Fig. 7.1 Four types of packets in the block switching method utilize the R/B mode transmission. At first, he types in a R/B link-packet with the name of his destination and required transmission data-rate. This link-packet reserves the time slots of each channel on the path from the source node to the destination node. Reaching to the destination node, link-packet completely connects the end-to-end link. Then, the destination node returns a link-ACK to the source node by the ACK-packet. When the source node recieves this, the source node transfers it to the source terminal. At this time, between the source and the destination terminals, the logical path is connected. Then, the user can transmit his message into the network, according to the data rate of this logical path. Then this message is packetted at the source node and is transmitted on R/B mode transmession. The final packet of this message makes this link open, when it is transmitted.

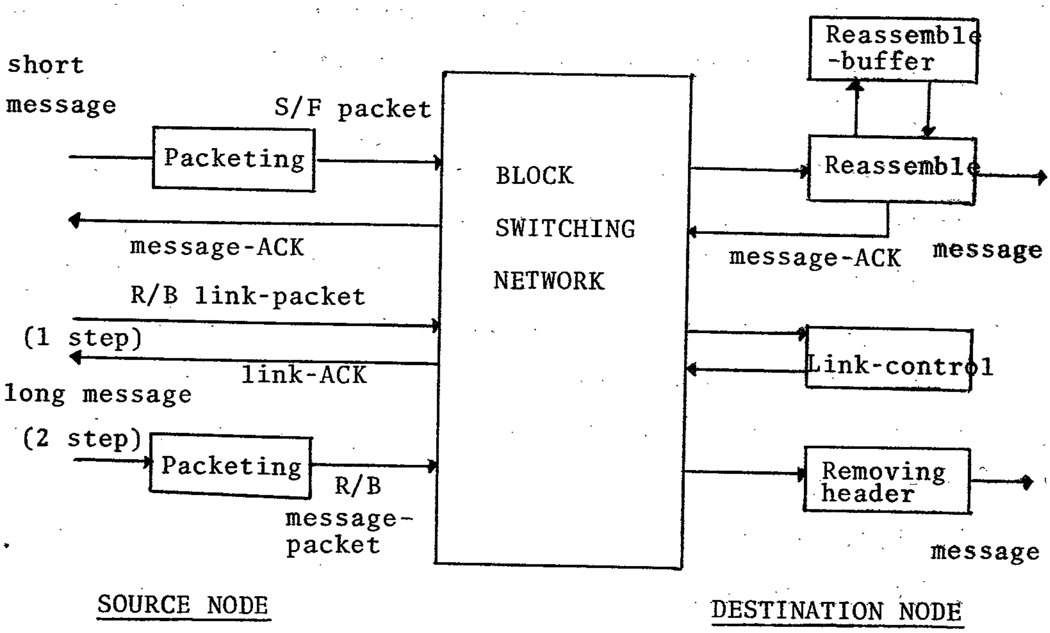


Fig. 7.2 The Behavior of the block switching network

7.3 Operating Behaviors of Switching Node

The operation that the switching node processes the arrival packets is shown in Fig.7.3. The arrival packets are recognized to be one of four types by its identification bits. Each type of packet is served with different nodal processing.

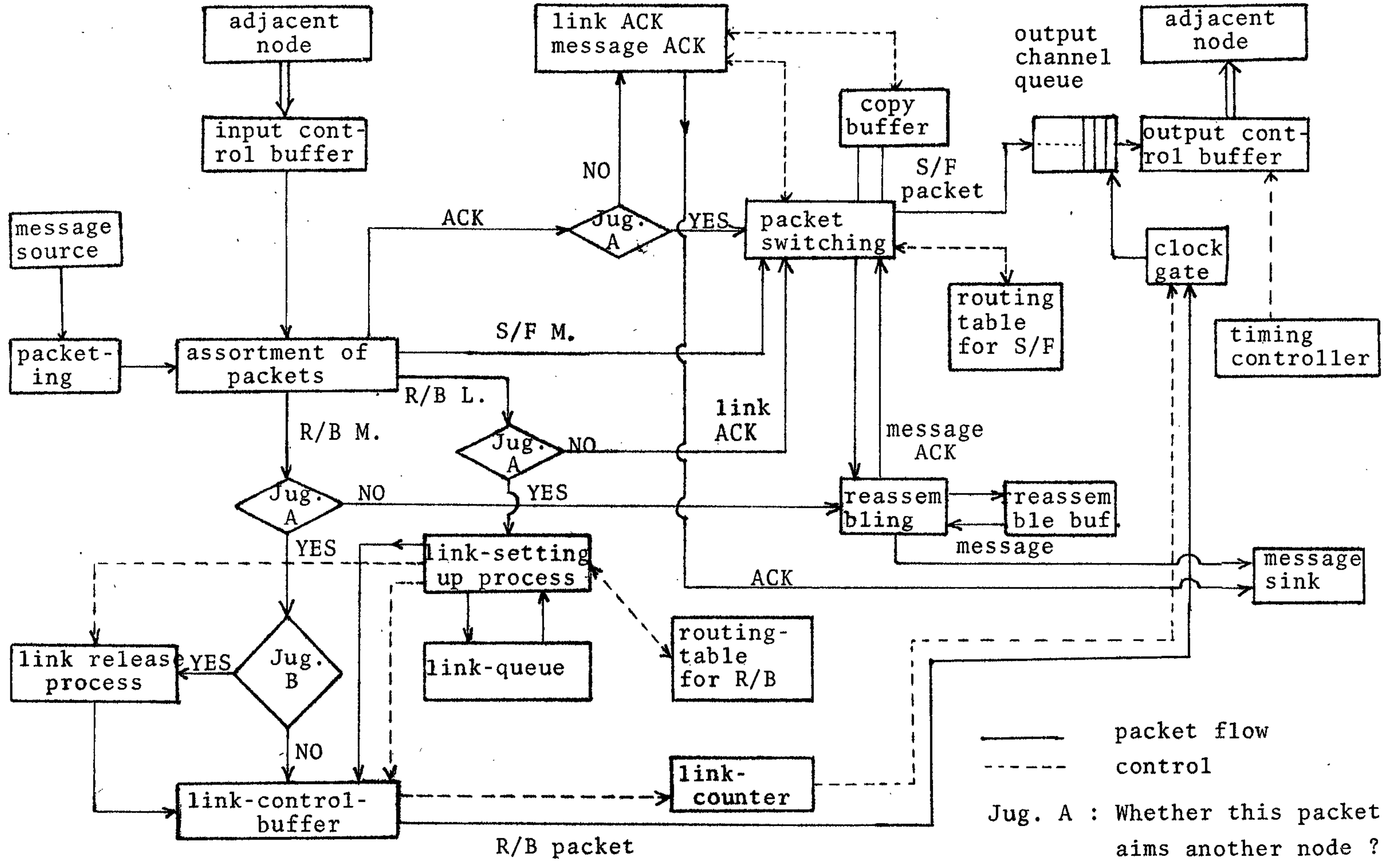
The first, an S/F message-packet is served with the packet switching processing. When an incoming packet is an S/F message-packet and destines this node, it is stored in the reassemble buffer till all packets of the same message arrive. If this packet destines another node, the next node to be transmitted is determined, utilizing the routing-control-table for the S/F packet.† Then, the packet is put into the output queue to wait for being transmitted to the node decided. If there is no other waiting packet in this queue, or if this packet has the highest priority in the queue at that time, it is transmitted to the next node on a given block. In this figure (i.e. Fig.7.3), we omit the process of a packet-ACK as described in Chapter 5.

The second, an R/B link-packet is served with the link-connection process. At first, an incoming link-packet is inquired whether it aims at this node, or not. In the former case, since the link-packet reaches to its destination node, the link is completely connected from the source node to the destination node and a logical path is established. Thus, this node returns a link-ACK to the source node, utilizing an ACK-packet.

Further, in the latter case of the R/B link-packet, the next node is chosen from all the neighboring nodes by the routing-table

---

† We mean by the S/F packet, both the S/F message-packet and the ACK-packet.



Jug. A : Whether this packet aims another node ?

Jug. B : Whether this packet is the final packet ?

Fig. 7.3 The behavior of switching node

for the R/B packet.† Then this switching node sets up a link to the next node. To set up a link is to secure one link-control-buffer in this node, and then to reserve the blocks at intervals of constant number of blocks. This interval of block can vary for each link. Thus, choosing adequate number of interval, we can obtain the requir data rate of a link. In one transmission channel, the number of lin control-buffers (we sometimes call this as a link-buffer) is limited to a fixed value. If all link-buffers are already utilized, link - packet is put into the link-queue and waits till a link-buffer is able to be used for this link-packet. The details of the link-control behavior and a link-control-buffer at each node are described i the next section. After setting up a link, the R/B link-packet is sent to the next node on one of the receiving blocks.

The third, an R/B message-packet is served with the process analogous to time-division-multiplexing line-switching. At first, the R/B message packet is examined whether this node is the destination of it, or not. In the former case, the header of packet is taken of and the contents of packet are transfered to the destination termina or computer. In the latter case, the packet is removed into the lin control-buffer and waits for the reserving block of that link-buffer The reserving block of each link-buffer is indicated by the link-counter in each link-buffer. When the link-counter directs that the following block is reserved, the clock gate is open and the R/B message-packet intervenes in the top of the output queue of the channel for the specific next node. Then, at the following block, the R/B message-packet can be transmitted. When, the final packet of this

---

† Similarly, we mean by the R/B packet, both the R/B message-packet and the R/B link-packet.

link is transmitted, it makes the link-buffer open at each node through the path. After this, this link-buffer can be utilized by any other link.

An ACK-packet is served with the nodal process as well as an S/F message packet.

In operating the block switching network, we must transfer several kinds of control—information to update the routing table, to monitor the failure of the system and so on. We assume that an ACK-packet can transfer those control-informations.

7.4 Link Control Behaviors and Link-Control-Buffer

In this section, a link-control-buffer and its behaviors are described. Each switching node has a certain number of link-control-buffers, which is shown in Fig.7.4. The link-control-buffer (or link-buffer) has three basic functions. Those are:

1) To store safely the R/B packet of this link
2) To detect the reserving block of this link
3) To transfer the R/B packet on the reserving block

Accomplishing these functions, the link buffer is composed of the followings as shown in Fig.7.4.

a) Packet-tag-number

This corresponds to the function 1) and is used to identify the packet from other packets in this node. When the node completely receives R/B packet, the node assignes a tag-number to that packet. This tag-number is put into the link-buffer. When an R/B packet is transfered to the next node, the packet-tag number in the link-buffer becomes zero and keeps zero till the following R/B packet is received.

b) Link-counter

This is applied to accomplish the functions 2) and 3). This link-counter indicates the reserving block and controls the clock gate to make packet-tag-number intervene into the top of output-queue.

In each link, the reservation of block is controlled not by the position of a block in the time axes, but by the interval of the block number, which is denoted by R. Thus, each link has the occupying block at every (R+1) blocks. This is managed by this link-counter.

The link-counter is keeping step with each block (i.e. time slot) and is substracted one at every block. On setting-up-link, the R/B link–packet secures the link-buffer and starts to reserve blocks. At the first reserving block, this link-packet is transfer-

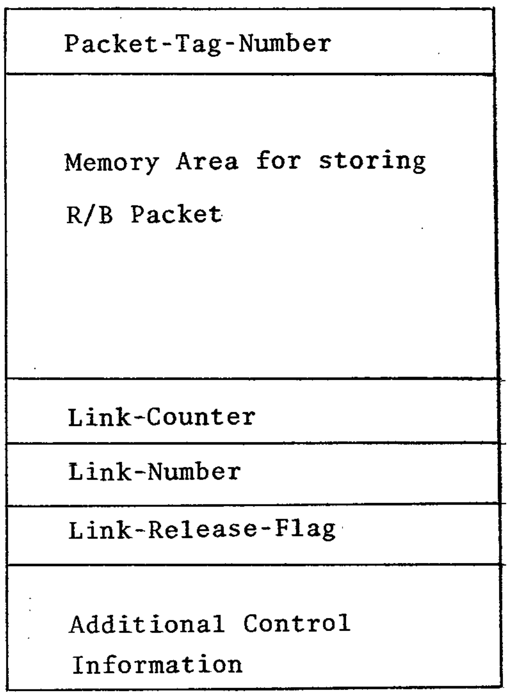


Fig. 7.4 Link-control-buffer

ed to the next node, and then, link-counter is set at the value of R. The counter is taken away one at each block. When the value of the counter becomes zero, the counter indicates that the next block is reserved by this link-buffer. Then, the counter opens the clock-gate and makes packet-tag-number intervene to the top of the output queue. Hence, the R/B packet in this link-buffer can utilize the next block. If there is no R/B packet at this time, (that is, packet-tag-number indicates zero) no intervening occurs, and another S/F packet can utilize this block. At the next block, the counter is reset at the value of R, again.

The interval of blocks, R is variable at each link. Thus, choosing an adequate value for R, we can obtain the required data rate of the link. But, since the value of R is not fixed, and several link-buffers are simultaneously utilized to one channel, some link counters may indicate the reservation block at the same time. Therefore, we must give priority to the link-control-buffer. When reservations are duplicated, the order of transmission is decided by the grade of the priority of the links. This additional delay of the R/B packet may affect the behaviors of the link-control at the next node. Thus, on setting-up-link, reservation of the switching node is set up so that the R/B packet will have to stay at node during several blocks.

c) Link-release-flag

This is utilized to release a link-control-buffer from a link. This flag has 1 or 0 as a value.

When a link-packet secures a link-buffer, the value of this flag is set to be 0. This is kept 0 till the final R/B-message-packet in this link arrives. This final packet resets the flag to be 1.

If the value of this flag is 0, no link-release operation is done. But when this is 1, this makes the contents of this link-buffer clean and the link-counter stop, as soon as the packet-tag-number intervenes into the output queue. After this, this link-buffer can be used by another link.

7.5 A Comparison of Three Switching Methods for Throughput Characteristics

In this section, we compare three switching methods i.e. packet switching, line switching, and block switching on the throughput characteristics that one transmission channel can handle. Let channel capacity be C, one packet length that is equal each other, P_L, and one link capacity in line switching and R/B mode of block switching, C_0.

For simplicity, we adopt the following analysis models to each switching methods.

1) Packet switching is represented by M/D/1 queueing model with input traffic of λ_p and transmission time of P_L/C.
2) Line switching is represneted by M/M/s queueing model. In this model, let average input traffic be λ_c, average message length, $\alpha \cdot P_L$, and density of channel-transmission, β. Where $\alpha \geq 1$, $0 < \beta \leq 1$, and the holding time of channel becomes

$$h = \frac{\alpha \cdot P_L}{\beta \cdot C_0} \qquad (7.1)$$

3) Block switching is represented by the compound model of the first and the second models with mixture rate r, where r is a rate of channel capacity of the R/B mode to the total channel capacity C.

A) Packet switching

This model has the Poisson arrival with mean λ_p per unit time and the constant service of P_L/C. Then, the channel utilization factor ρ_p becomes

$$\rho_p = \frac{\lambda_p \cdot P_L}{C} \tag{7.2}$$

For the value of ρ_p and the service time required for a packet to be completely transmitted on channel, the summation of the waiting time and the service time in this queue, W is given by

$$W = \frac{2-\rho_p}{2C\ (1-\rho_p)} \tag{7.3}$$

The throughput of packet switching T_p is given by

$$T_p = C \cdot \rho_p \cdot \eta_T \tag{7.4}$$

where η_T is the transmission efficiency discribed in Chapter 2.

B) <u>Line switching</u>

This model is assumed to have the Poisson arrival with mean λ_c, the exponential service time with mean h, and S independent channels. Since the total channel capacity is C and C_0 for each channel, the channel number S, the channel utilization factor of one channel a_c, and the total channel utilization factor ρ_c are given by

$$S = C/C_0 \tag{7.5}$$

$$a_c = \lambda_c \cdot h \tag{7.6}$$

$$\rho_c = a_c/S = \lambda_c \cdot h \cdot C_0/C \tag{7.7}$$

In this case, the call reject rate ε_r that message is rejected, due to the fact that all channels are already utilized, is given by

$$\varepsilon_r = \frac{a_c^S}{S!} \quad \sum_{n=0}^{S} \frac{a_c^n}{n!} \tag{7.8}$$

Eq.(7.8) is called as Erlang's loss formula. The throughput of line-switching is given by

$$T_c = \lambda_c \cdot \alpha \cdot P_L \tag{7.9}$$

C) Block switching

In this method, channel capacity C is divided into two modes, i.e. R/B mode and S/F mode according to the mixture rate r. Hence the R/B mode has capacity with $r \cdot C$ and S/F mode with $(1-r) \cdot C$.

The R/B mode resembles to the line switching. In this case, let all links have the same link capacity C_0, for simplicity. The allowable number of link in one channel S_{bc} becomes

$$S_{bc} = \frac{r \cdot C}{C_0} \tag{7.10}$$

The holding time is similar to Eq.(7.1) and call reject rate is given similarly by Eq.(7.8). The throughput of the R/B mode of the block switching T_{bc} is given by

$$T_{bc} = (1-r) \cdot \lambda_c \cdot \alpha \cdot P_L \tag{7.11}$$

Since the S/F mode is almost similar to the packet switching, the throughput T_{bs} is

$$T_{bs} = (1-r) \cdot C \cdot \rho_p \cdot \eta_T + \{r \cdot C - \lambda_c \cdot (\alpha+1) P_L\} \cdot \rho_p \cdot \eta_T \tag{7.12}$$

Therefore, the total throughput of the block switching T_b is

$$T_b = T_{bs} + T_{bc} \tag{7.13}$$

Fig.7.5 shows the comparison of the throughput rate of the block switching to the line switching and that of the block switching the to the packet switching versus mixture rate r. In this case, let C be 1 Mbits/sec, C_0 50 kbits/sec, and P_L 1000 bits. Furthermore, we decide the service grade of each switching method as follows. In the packet switching and the S/F mode of the block switching, the waiting and the transmission time $\bar{W}$ is chosen to be less than 0.002 second (This means that only the waiting time is less than 0.001). In the line switching and the R/B mode of the block switching, call reject rate is chosen to be less than 0.001. Further, we choose α=20, β=0.5. Thus, the average holding time of the line switching and the R/B mode becomes 0.8 sec. Since the Single packeting is utilized in the packet switching and the S/F mode, the transmission efficiency η_T is chosen to be 0.5. As shown in Fig.7.5, the block switching is superior to the other two switching methods on the throughput characteristics.

7.6 Considerations

We consider the advantages, the disadvantages, and the important problems on the block switching and we summarize them as follows.

A) The advantages of the block switching

A-1) In the block switching network, a network user can choose the transmission mode from two optional modes according to his message length or holding time.

A-2) If a user chooses R/B mode, he can easily select the data rate of link by setting adequate value R, according to his message length or data rate of his terminal.

A-3) In the line switching, the channel cannot be utilized by any other users and is empty, during establishing the link-connection. This is an over-head of the line switching that cannot be neglected. But, in the block switching, when link-connection is established, the reserved block which has been already set-up link can be utilized freely by other S/F packets. Thus, there exists no set-up loss in the block switching.

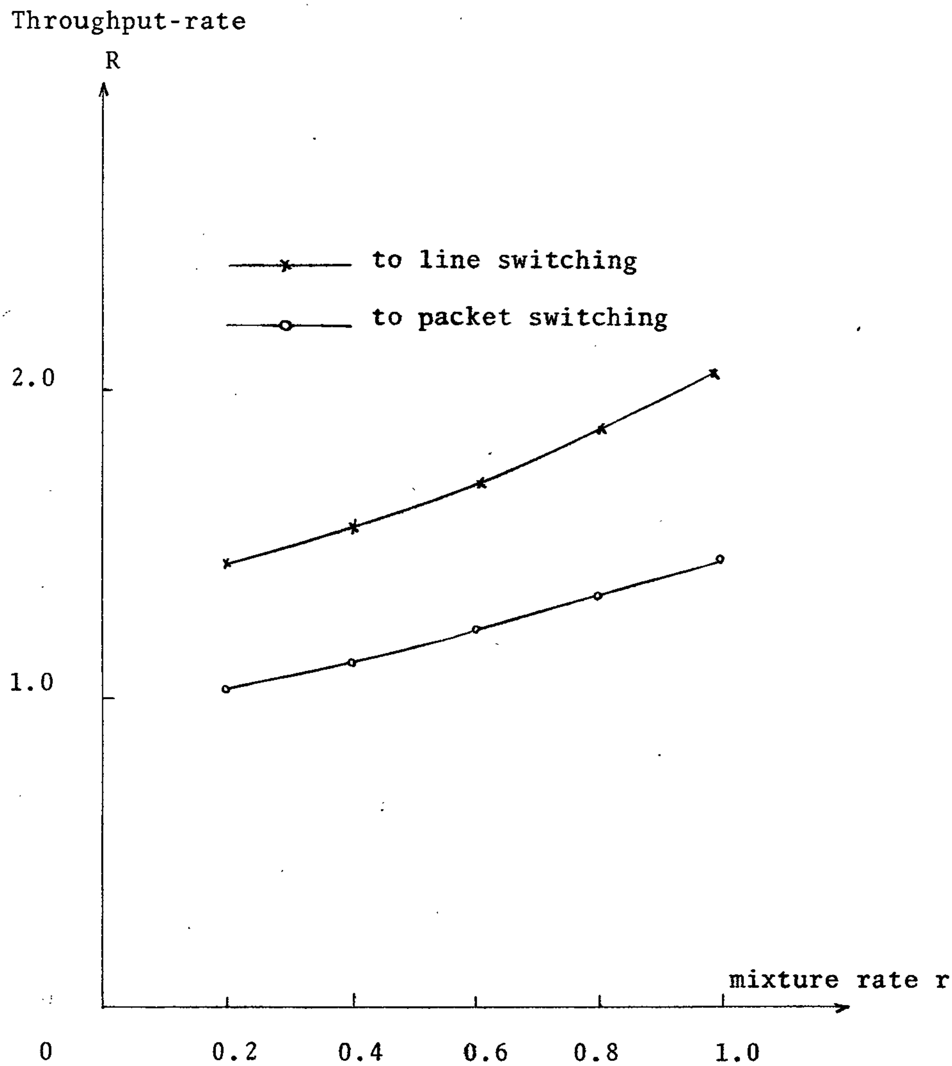


Fig. 7.5 Throughput rate versus mixture rate

A-4) When the whole network is over-congested and the all lock-up phenomenum occurs, the packet switching method cannot release the network from this lock-up without any serious losses. On the contrary, the block switching has the R/B mode capacity that cannot cause the lock-up. Thus, even if the area of S/F mode is fully locked-up, it can easily release the network from lock-up, utilizing several number of R/B channels as S/F channels.

A-5) As shown in Fig.7.5, the block switching is better than other switchings in throughput characteristics.

A-6) As described above, this block switching has more efficient and more flexible ability than the line switching and the packet switching.

B) <u>The disadvantages of the block switching</u>

B-1) In the block switching, the switching node requires more functions than in other switchings. Thus, the load of the CPU in the node is considered to increase greatly.

B-2) Each channel is slotted by the one-packet-length. Thus, in each channel, the timing function is required.

B-3) In each link of the R/B mode, the link-control-buffers are required at each node on the path of link. Hence, in the switching node, a great deal of buffer memory is needed.

B-4) In this method, the Single packeting method is utilized. This packeting method has worse efficiency than other packeting methods and the variable-length-type.

C) <u>The important problems in the block switching</u>

C-1) Routing procedures for the R/B mode packet and the S/F mode packet

C-2) Affection of the fluctuation of traffic of the R/B mode on the delay and congestion state of the S/F mode

C-3) Schedules of the reservations of the R/B mode in setting up link

7.6 Conclusion

In this chapter, we present the block switching method as hybrid switching. The general concepts of block switching are described. Then, we compare this method with the packet switching and the line switching in throughput and obtain the superiority of block switching to these switching methods. Furthermore, we discuss on the advantages, the disadvantages, and the problems on the block switching, and insist that there are many good points in this method.

In constructing the hybrid network, there are three types of the orientations. Those are:

1) Line-switching oriented hybrid network with the ability of packet switching
2) Packet-switching oriented hybrid network with the ability of line-switching
3) Neutral hybrid network with the ability of both switchings

The block switching can become any of the three hybrid network, varying the limit number of the link-control-buffer in the node. It is strongly insisted that the rate between two switching modes can be varied, even while network is operating. We insist that the block switching system can be actually realized, in addition to those advantages.

CHAPTER 8

CONCLUSIONS

This thesis has pursued to gain insight into the packet switching system applicable to computer-communication networks. The major objectives of this thesis are:

1) To develop the mathematical tools for evaluation and optimizing the system performance
2) To exploit useful techniques for packet switching

For these purposes, we have investigated several important problems described as follows.

1) Packeting method
2) Analysis of the delivering behavior of packet-sequence of the same message
3) Analysis of the message traverse time
4) Packet-ACK method
5) Isarithmic flow control method
6) Block switching method

In regard to each problem, the representative results in this thesis are summarized as follows.

<u>Packeting method</u>

1-a) Four types of packeting methods (i.e. Single, Sequential, Exponential, and Complex packetings) have been presented.

1-b) Utilizing three items (i.e. the set of packets P, the probability $P_{pm}(i)$ that one message is divided into i packets, and the probability $P_{sp}(i)$ that one packet contains i-segment), four packeting methods have been represented mathematically.

1-c) Transmission efficiency and packet error rate have been defined as the evaluaion measures, and have been derived mathematically in each packeting method. Utilizing these measures, we can design the optimum packet structure.

The delivering behavior of packet-sequence of the same message

2-a) We have been able to represent this process as the Markov chain model, introducing some approximations. The validity of these approximations has been secured by computer simulation.

2-b) Analyzing this Markov model, the probability density function of packet-intervening number has been obtained. Utilizing this probability function, the reassemble delay of a message can be easily derived.

The message traverse time

3-a) From the aspects of some unique points, the message traverse time of general packeting methods has been obtained.

3-b) Utilizing this as the evaluation measure, the designs of topological structure and capacity assignment can be accomplished.

Packet-ACK method

4-a) Two multi-ACK control methods (i.e. time-control and waiting-number-control) have been presented.

4-b) Various ACK methods have been classified into three classes.

4-c) The transmission efficiency and the packet response time have been defined as the performance measures, and have been mathematically analyzed on each packet-ACK method.

4-d) As long as the transmission efficiency and the packet response time are concerned, we can insist that time-control multi-ACK method with piggy-back fashion is the best in the various packet-ACK methods.

Isarithmic flow control method

5-a) We have represented one switching node in the network as the duplicated queueing model and have analyzed them mathematically.

5-b) As the results of this analysis, the effectiveness of the congestion control and the excess-control have been obtained.

5-c) The optimum storage size of permit can be easily derived.

5-d) Further, the message traverse time in the Isarithmic network can be obtained.

5-e) The improvement that the external packet which destines for the adjacent node can be transmitted without a permit, only when the transmission channel is free, have been introduced. It is noted that this improvement makes the throughput of the network quite large, without any serious congestions.

Block switching method

6-a) The block switching that have both abilities of packet switching and line switching have been presented.

6-b) This method has been shown to have many advantages regarding to throughput, flexibility, reliability, and high performance.

6-c) We insist that this block switching can be easily realized and vary its characteristics suitably according to the change of the input-traffic of the network.

In this dissertation, we have developed several mathematical tools to evaluate and design the packet-switching computer-communication networks. But we have not dealt with the optimization problems at all. Therefore, utilizing these results, the optimization of the packet-switching computer-communication network is left unsolved for future research.

REFERENCES

[1] G.C.Cole," Computer Network Measurements: Techniques and Experiments," Doctor Dissertation of UCLA , 1971.

[2] J.Martin," Telecommunication and the Computer," Prentice-Hall, 1969.

[3] G.J.Evans," Experience Gained from the American Airlines SABRE System Control Program," Proc. of the 22nd National Conference, ACM, 1967.

[4] M.Hamsher," Communication System Engineering Handbook," McGraw-Hill, 1968.

[5] S.F.Mendicino," OCTOPUS : The Lawrence Radiation Laboratory Network," Courant Computer Science Symposium 3, Dec. 1970.

[6] Mckay, et al," IBM Computer Network / 440," Courant Computer Science Symposium 3, Dec. 1970.

[7] D.W.Davirs, et al," A Digital Communication Network for Computers Giving Rapid Response at Remote Terminals," ACM Symposium on Operating System Principles, Oct. 1967.

[8] R.A.Scantlebury, et al," The Design of a Message Switching Center for a Digital Communication Network," IFIP Information Processing 68, Vol.2 1969.

[9] D.Davies," Packet Switching in a Public Data Network," IFIP Congress 1971.

[10] L.G.Roberts, et al," Computer Network Development to Achieve Resource Sharing," AFIPS SJCC, May. 1970.

[11] F.E.Heart, et al," The Interface Message Processor for the ARPA Computer Network," AFIPS Conference Proceedings, SJCC, May 1970.

[12] C.S.Carr, et al," HOST-HOST Communication Protocol in the ARPA Network," AFIPS Conference Proceedings, SJCC, May 1970.

[13] S.Ornstein, et al," Terminal IMP for ARPA Computer Network," AFIPS SJCC Vol.40, 1972.

[14] L.Kleinrock," Communication Nets : Stochastic Message Flow and Delay," McGraw-Hill, 1964.

[15] P.Baran," On Distributed Communication Networks," IEEE Trans. Vol.CS-12, March 1964.

[16] P.J.Burke," The Output of a Queueing System," Operations Research, Vol.4, 1956.

[17] J.R.Jackson," Network of Waiting Lines," Operations Research, Vol.5, 1957.

[18] H.Frank, et al," Topological Optimization of Computer Networks," Proc. of the IEEE Vol.60, No.11, Nov. 1972.

[19] W.W.Chu," Optimal File Allocation in a Multicomputer Information System," IFIP Information Processing 68, Vol.2, 1968.

[20] P.P.S.Chen," Optimal File Allocation in Multi-Level Storage Systems," AFIPS NCC Vol.42, June 1973.

[21] G.W.White," Message Format and Data Communication Link Control Principles," IEEE Trans. on COMMUNICATION Vol.COM-20, No.3, June 1972.

[22] W.W.Chu," Buffer Behavior for Poisson Arrivals and Multiple Synchronous Constant Outputs," IEEE Trans. on COMPUTERS, Vol. C-19, No.6, June 1970.

[23] W.W.Chu," Buffer Behavior for Batch Poisson Arrivals and Single Constant Output," IEEE Trans. on COMMUNICATION, Vol.COM-18, No.5, Oct. 1970.

[24] J.H.Chang," An Analysis of Buffering Techniques in Teleprocessing Systems," IEEE Trans. on COMMUNICATION, Vol.COM-20, No.3, June 1972.

[25] J.F.Zeigler, et al," Nodal Blocking in Large Networks,"

IEEE I.C.C. 1971.

[26] G.L.Fultz," Adaptive Routing Techniques for Message Switching Computer-Communication Networks," Doctor Dissertation of UCLA, 1972.

[27] F.B.Wood," Optimum Block Length for Data Transmission with Error Checking," Communication and Electronics, No.40, Jan. 1959

[28] W.W.Chu," Optimal Fixed Message Block Size for Computer Communications," IFIP Congress, 1971.

[29] R.L.Kirlin," Variable Block Length and Transmission Efficiency," IEEE Trans. on COMMUNICATION Tec. Vol.COM-17, No.3, June 1969.

[30] R.T.Prosser," Routing Procedures in Communications Network-Part 1 ; Random Procedures," IRE Trans. on Communication Systems CS-10, Dec. 1962.

[31] R.T.Prosser," Routing Procedures in Communications Network-Part 2 ; Directory Procedures," IRE Trans. on Communication Systems, CS-10, Dec. 1962.

[32] B.W.Boehm, et al," Adaptive Routing Techniques for Distributed Communications System," IEEE Trans. on COMMUNICATION, Vol.COM-17, No.3, June 1969.

[33] G.L.Fultz, et al," Adaptive Routing Techniques for Store-and-Forward Computer-Communication Network," IEEE I.C.C. 1971.

[34] R.E.Kahn, et al," Flow Control in a Resource-Sharing Network," IEEE Trans. on COMMUNICATION Vol.COM-20, No.3, June 1972.

[35] D.W.Davies," The Control of Congestion in Packet-Switching Network," IEEE Trans. on COMMUNICATION, Vol.COM-20, No.3, June 1972

[36] W.L.Price," Simulation Studies of an Isarithmically Controlled Store and Forward Data Communication Network," Information Processing 74, 1974.

[37] W.L.Price," Simulation of Packet-Switching Networks Controlled

on Isarithmic Principles," Proc. of 3rd ACM/IEEE Data Communications Symposium, Nov. 1973.

[38] R.J.Benice, et al," Comparisons of Error Control Techniques," IEEE Trans. on COMMUNICATION, Vol.COM-12, Dec. 1964.

[39] R.J.Benice, et al," An Analysis of Retransmission Systems," IEEE Trans. on COMMUNICATION, Vol.COM-12, Dec. 1964.

[40] L.Kleinrock, et al," Packet-Switching in a Slotted Satellite Channel," AFIPS NCC, Vol.42, June 1973.

[41] L.G.Roberts," Dynamic allocation of satellite capacity through packet reservation," AFIPS NCC Vol.42, June 1973.

[42] P.E.Jackson, et al," A Study of Multiaccess Computer Communication," AFIPS SJCC, Vol.34, 1969.

[43] B.Meister," Optimization of a New Model for Message-Switching Networks," IEEE I.C.C. 1971.

[44] R.G.Miller," A Contribution to Theory of Bulk Queues," J. Roy. Statist. Soc., Vol.21, 1959.

[45] J.Otterman," Grade of Service of Direct Traffic Mixed with Store-and-Forward Traffic," IRE Trans. on COMMUNICATIONS SYSTEMS March 1962.

[46] J.Otterman," Probability Distribution of Delays for Store-and-Forward Traffic Mixed with Direct Traffic," IEEE Trans. on COMMUNICATIONS SYSTEMS, March 1963.

[47] T.Hasegawa, et al," Digital Data Dynamic Transmission Systems," IEEE Trans. on COMMUNICATION TEC., Sep. 1964.

[48] F.Closs," Message Delays and Trunk Utilization in Line-Switched and Message-Switched Data Networks," First USA-JAPAN Computer Conference, 1972.

[49] J.D.C.Little," A Proof for the Queueing Formula : L=λW," Operational Research, Vol.9, 1952.

[50] A.G.Gatfield," ARQ Error Control on the Satellite Channel," ICC 74, June 1974.

[51] H.Okada, et al," On Packeting Methods in Message Switching Communication Networks," Papers of Technical Group on Switching, I.E.C.E., Japan, SE-31, 1971 (in Japanese).

[52] H.Miyahara, H.Okada, et al," Packeting Methods in Computer Networks," Conf. on "Islands of Application", IEEE Computer Soc. and U.S. Dept. of Commerce, 1972.

[53] H.Okada, et al," Analysis of Intervening Probability Distribution in Packet Switching Computer Networks," Trans. I.E.C.E., Japan, 55-A, Vol.12, Dec. 1972 (in Japanese).

[54] H.Miyahara, H.Okada, et al," Packeting Methods in Store-and-Forward Computer Networks," Trans. I.E.C.E., Japan, 55-A, Vol.9,Sept. 1972 (in Japanese)

[55] H.Okada, et al," The Behaviors of Message Reassemling in Packet Switching Computer Communication Network," Papers of Technical Group on Switching, I.E.C.E., Japan, EC 72-43, 1972 (in Japanese).

[56] H.Okada, et al," Analysis of Message Traverse Time in Packet Switching Computer Metworks," Papers of Technical Group I.E.C.E Japan, SE 73-29, 1973 (in Japanese).

[57] H.Okada, et al," Packeting Methods and Transmission Efficiency in Packet Switching Computer Networks," Technology reports of the OSAKA Univ. Vol.23, No.1139, 1973.

[58] H.Okada, et al," Packet-ACK Methods in Packet Switching Computer Networks," Papers of Technical Group on Switching, I.E.C.E., Japan, SE 73-12, 1973 (in Japanese).

[59] H.Okada, et al," Block Switching System," Papers of Technical Group on Switching I.E.C.E., Japan SE 74-11, 1974(in japanese).

[60] H.Okada, et al," Analysis of the Isarithmic Method in Packet Switching Networks," Proc. of 15th Conv., IPSJ Japan, Dec. 1974 (in Japanese).

[61] H.Okada, et al," Analysis of the Behavior of Packet-ACK in Packet Switching Computer Networks," Trans. I.E.C.E., Japan 57-A 1975 To be published (in Japanese).

[62] T.Iwahashi, H.Okada," A Consideration on Packet Switching System with Satellite Communication," Papers of Technical Group on Switching I.E.C.E., Japan, Dec 1974(in Japanese).

[63] Y.Yamasaki, H.Okada, et al," Analysys of Multi-ACK Methods in Packet Switching Computer Networks," Papers of Technical Group on Switching I.E.C.E., Japan, SE 74-21 1974(in Japanese).

[64] M.Suzuki, H.Okada, et al," Analysis of Isarithmic Packet Switching Networks and its Improvement," Papers of Technical Group on Switching I.E.C.E., Japan Dec. 1974 (in Japanese).

[65] L.Takacs," Introduction to the Theory of Queue," Oxford Univ. Press, 1962.

[66] T.L.Saaty," Elements of Queueing Theory with Applications," McGraw-Hill, 1961.

[67] T.Honma," Queueing Theory," Rikogakusha, 1966(in Japanese).

[68] E.Parzen," Stochastic Processes," Holden-Day, 1962.

[69] M.Loeve," Probability Theory," D.Van Nostrand Company, 1955.

論文目録

岡田博美

主論文 Packet Switching in Computer Communication Networks
(計算機通信網におけるパケット交換方式)

(主論文のうち印刷公表したもの)

1. 蓄積交換計算機網におけるパケッティング方式
電子通信学会 論文誌A
55巻 9号
昭和47年9月25日

1. パケット交換計算機網における伝送時間割込み分布の解析　電子通信学会 論文誌A
55巻 12号
昭和47年12月25日

1. Packeting Methods and Transmission Efficiency in Packet Switching Computer Network
(パケット交換計算機網におけるパケッティング方式と伝送効率)
Technology Reports of the Osaka University
Vol. 23, No. 1139　昭和48年 10月

1. パケット交換計算機網におけるメッセージ編集時動作特性について”　電子通信学会 電子計算機研究会資料
EC 72-43
昭和47年 12月

1. パケット交換計算機網における転送時間について

電子通信学会 交換研究会 資料
SE 73−29
昭和48年 8月

1. パケット交換計算機網におけるACK方式

電子通信学会 交換研究会 資料
SE 73 - 44
昭和48年 12月

1. ブロック交換方式

電子通信学会 交換研究会 資料
SE 74−11
昭和49年 6月

1. パケット交換計算機網における ACK方式(II)

電子通信学会 交換研究会 資料
SE 74−21
昭和49年 7月

1. パケット交換計算機網における等数パケット輻輳制御方式の解析

電子通信学会 交換研究会 資料
SE 74-30
昭和49年 8月

1. パケット交換計算機網における等数パケット輻輳制御方式の解析(II)

電子通信学会 交換研究会 資料
SE 74 - 55
昭和49年 12月

（主論文のうち未公表のもの）

1. パケット交換計算機網におけるパケットACK動作の解析.
電子通信学会 論文誌 A
昭和50年2月以降掲載予定

論文採録決定通知状

昭和49年10月17日

岡田博美　殿
他2名

〒105 東京都港区芝公園 3-5-8
機械振興会館
社団法人 電子通信学会
編集委員会

論文題名　パケット交換計算機網におけるパケットACK動作の解析

ご寄稿いただきました上記論文（受付番号　177，A　分冊）は昭和49年10月15日に採録と決定いたしましたのでご通知申し上げます。

なお，掲載は2月号以降の予定になっておりますのでしばらくお待ち下さい。