

## **Copyright Warning & Restrictions**

The copyright law of the United States (Title 17, United States Code) governs the making of photocopies or other reproductions of copyrighted material.

Under certain conditions specified in the law, libraries and archives are authorized to furnish a photocopy or other reproduction. One of these specified conditions is that the photocopy or reproduction is not to be “used for any purpose other than private study, scholarship, or research.” If a user makes a request for, or later uses, a photocopy or reproduction for purposes in excess of “fair use” that user may be liable for copyright infringement,

This institution reserves the right to refuse to accept a copying order if, in its judgment, fulfillment of the order would involve violation of copyright law.

**Please Note: The author retains the copyright while the New Jersey Institute of Technology reserves the right to distribute this thesis or dissertation**

Printing note: If you do not wish to print this page, then select “Pages from: first page # to: last page #” on the print dialog screen

The Van Houten library has removed some of the personal information and all signatures from the approval page and biographical sketches of theses and dissertations in order to protect the identity of NJIT graduates and faculty.

## ABSTRACT

### PERFORMANCE MODELING OF HIGH DATA RATE RING NETWORKS USING STOCHASTIC PETRI NETS

by  
Le Zhong

This thesis has proposed a performance analysis method for Carrier Sensing Multiple Access Ring Network (CSMA) based on Petri net methodologies. The method can offer more accurate results than traditional queuing methods, but less than simulation results. In the CSMA network access protocol is studied in this thesis, when a passing message is detected by a station, it stops transmitting immediately in order to avoid a collision.

A generalized stochastic Petri nets model is made first, then the SPNP package is used to do the analysis. The analysis results show that even at a very heavy load fraction of near 2.0, the average delay of packets is at a very low level by transmitting the packages at the gap between two packages in the ring. The network performance over a wide range of network parameters including various packet sizes and numbers of stations are analyzed. The limitations and future research related to CSMA performance studies are discussed.

**PERFORMANCE MODELING OF HIGH DATA RATE RING NETWORKS  
USING STOCHASTIC PETRI NETS**

by  
**Le Zhong**

*Cotton*

**A Thesis  
Submitted to the Faculty of  
New Jersey Institute of Technology  
in Partial Fullfillment of the Requirements for the Degree of  
Master of Science in Electrical Engineering**

**Department of Electrical and Computer Engineering**

**January 1994**

Blank Page

APPROVAL PAGE

PERFORMANCE MODELING OF HIGH DATA RATE RING NETWORKS  
USING STOCHASTIC PETRI NETS

Le Zhong

~~\_\_\_\_\_~~ 1/6/94  
Dr. Mengchu Zhou, Dissertation Advisor Date  
Assistant Professor of Electrical and Computer Engineering, NJIT

~~\_\_\_\_\_~~ 1/6/94  
Dr. Anthony Robbi, Committee Member Date  
Associate Professor of Electrical and Computer Engineering, NJIT

~~\_\_\_\_\_~~ 1/6/94  
Dr. David T. Wang, Committee Member Date  
Assistant Professor of Computer and Information, NJIT

## BIOGRAPHICAL SKETCH

**Author:** Le Zhong

**Degree:** Master of Science in Electrical Engineering

**Date:** January 1994

**Date of Birth:**

**Place of Birth:**

### Undergraduate and Graduate Education

- Master of Science in Electrical Engineering,  
New Jersey Institute of Technology,  
Newark, New Jersey, 1994
- Bachelor of Science in Electrical Engineering,  
China Textile University,  
Shanghai, China, 1986

**Major:** Electrical Engineering

This dissertation is dedicated to  
my family



## ACKNOWLEDGMENT

I wish to express my sincere thanks to Dr. Mengchu Zhou for his ingenious guidance and encouragement throughout the progress of this thesis. Special thanks to Professors Anthony Robbi and David Wang for serving as members of the committee.

## TABLE OF CONTENTS

Chapter	Pages
1. INTRODUCTION .....	1
2. BRIEF REVIEW OF PETRI NET .....	5
2.1 Basic Concept .....	5
2.2 Definition .....	5
2.3 Firing of a Transition .....	6
2.4 Generalized Stochastic Petri Nets .....	6
2.5 Properties of SPNP .....	8
3. PROTOCOL AND MODEL DESCRIPTION .....	10
3.1 A Review of CSMA .....	10
3.2 The Carrier Sensing Controller .....	11
3.3 Protocol Description .....	13
3.4 GSPN Model for CSMA/RN .....	15
3.5 The Model Operation .....	18
4. PERFORMANCE ANALYSIS .....	20
4.1 Analysis Assumption and Condition .....	20
4.2 An Example to Run SPNP Program .....	23
4.3 The Description of Petri Net Model Parameters .....	27
4.4 Analysis Result V.S. Different Offered Load .....	29
4.5 Analysis Result V.S. Different Number of Stations .....	31
4.6 Analysis Result V.S. Different Average Packet Size .....	33
4.7 Analysis Result for Small Buffer Size .....	35
4.8 Comparison with Simulation Result .....	37
5. IMPROVED GSPN MODEL .....	39
5.1 The Idea of the Improvement and Improved GSPN Model .....	39

**TABLE OF CONTENTS**  
(Continued)

<b>Chapter</b>	<b>Pages</b>
5.2 The Improved Model Description .....	39
5.3 Performance Analysis Result .....	42
6. CONCLUSION .....	46
APPENDIX A. SPNP Source Code .....	49
APPENDIX B. SPNP Output Data .....	54
BIBLIOGRAPHY .....	78

## LIST OF TABLE

Figure	Pages
3-1 Parameters Associated with Transitions.....	18
4-1 Parameters List .....	24
4-2 Parameters List .....	28
4-3 Parameters for GSPN Model .....	29
4-4 Parameters for GSPN Model .....	31
4-5 Parameters for GSPN Model .....	33
4-6 Parameters for GSPN Model .....	35
5-1 Improved Model Parameters .....	42

## LIST OF FIGURE

Figure	Pages
1-1 OSI Seven Layer Model .....	2
2-1 Transition Rule .....	7
3-1 CSMA/RN Logic Diagram .....	12
3-2 Network Operation Condition .....	14
3-3 Petri Net Model .....	17
4-1 Network Ring .....	21
4-2 Analysis Result V.S. Different Offered Load .....	30
4-3 Analysis Result V.S. Different Number of Stations .....	32
4-4 Analysis Result V.S. Different Average Packet Size.....	34
4-5 Analysis Result V.S. for Small Buffer Size .....	36
4-6 A Simulation Result .....	38
5-1 Improved Petri Net Model .....	40
5-2 Average Delay Performance for Improved Model .....	43
5-3 Throughput Performance for Improved Model .....	44

# CHAPTER 1

## Introduction

Over the last decade, computer networking has changed enormously. Ten years ago, computer networks were exotic research tools used only by few specialists. Today, computers ranging from personal computer to supercomputers are more likely to be part of a network than not. Most organizations that use computers either already have or are planning to install one or more local area networks.

Only a few years ago, the design of a computer network was something of a black art. All that has changed now. A series of International Standards for describing network architecture have been issued. These standards are known as the OSI Reference Model.

OSI (Open System International) Reference Model has seven layers, as shown in Figure 1-1. Base on this seven layer model, IEEE has produced several standards for local area networks (LANs). These standards, collectively known as IEEE 802, include CSMA/CD (802.3), token bus (802.4), and token ring (802.5). The various standards differ at the physical layer and medium access control (MAC) sublayer, but compatible at the data link layer.

There are three generations [12] of networks based on the underlying physical-level technology employed. Networks built before the emergence of fiber optic technology (i.e., those based on copper-wire or microwave-radio technology) are referred to as the first-generation. The second-generation networks employ fiber in traditional architecture. An excellent example of this generation is the upgrade of long-haul trunks in a wide area network (WAN) from copper or microwave-radio to fiber connections. Although

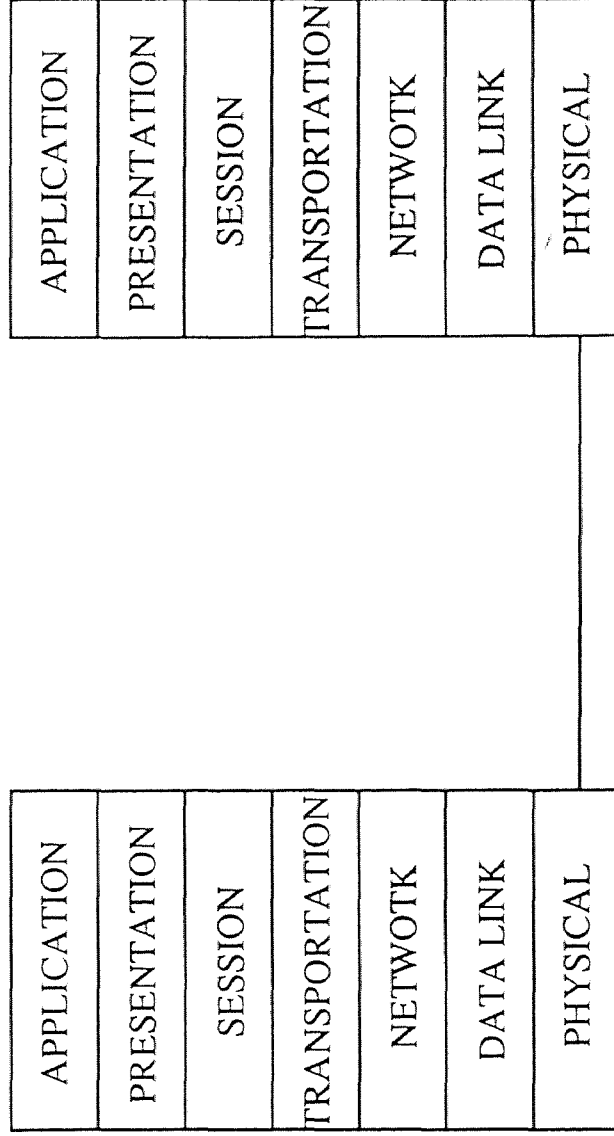


Figure 1-1 OSI Seven Layer Model

some improved performance can be achieved by employing fiber (e.g., higher data rates, lower error rates), the limitation of this generation is due to the electronic front ends employed at the network nodes. In the third-generation networks, fiber is used because of its unique properties. These networks employ totally new approaches to exploit the unique properties of fibers in order to meet the needs of emerging high-bandwidth applications.

Examples of applications that will demand true gigabit per-second networks are those involving high-definition three-dimensional imaging, where a user needs to interact with the image in real time. For instance, the user may want to rotate the image on the screen or vary some parameters manually while simultaneously seeing the effects of this change. Providing such interactive human-in-the-loop performance will require the transmission of several screens per second with latencies of fractions of seconds. The need for gigabit-per-second networks occurs where these images must be generated remotely.

Because of high bandwidth of fiber optics, fiber networks are becoming increasingly important. In this paper, we are going to present a significant extension of the carrier sensing multiple access (CSMA) ring network access protocol according to high performance of fibers, which was originally proposed in [7]. It is a collision free and a multiple access protocol. This network belongs to the third generation. Petri nets are used to construct its mathematical and graphical models. Then SPNP software package [4] is used to do our analysis, and the result will show that even near at a load fraction of 2.0, the performance of the network keeps at an excellent level.

In this thesis, we analyze a new format of CSMA (we call it CSMA/Ring Network). Analytical methods are very popular in network performance modeling and analysis. By this method, Petri nets (PNs),



especially generalized stochastic Petri nets are much better suited to the modeling of the system. The exact results can be easily obtained for the class of systems with Poisson arrivals and exponentially distributed service times. A stochastic Petri net package SPNP [3] [4] is utilized to automatically generate and solve the Markov models. To analyze a CSMA/RN system, it is necessary to make a PN model which describes the system protocol and operation conditions. Due to the complexity of the CSMA/RN system, how to obtain a group of parameters that are proper to the stochastic Petri net model is a main topic to be researched in this thesis. With the discussion of the proper parameters, a group of analysis results will be given.

Accordingly, the objectives of this thesis are as follows:

- To present an extension of CSMA network which is suited for high data rate communication between computers.
- To model CSMA/RN network with generalized stochastic Petri nets.
- To present and analyze the performance of this CSMA/RN which is suited for both LANs and WANs.

This thesis is organized as follows. The next chapter provides basic concepts and properties of Petri nets, introduction to Generalized Stochastic Petri Nets and SPNP software [4]. Chapter 3 provides a protocol description and the GSPN models. Chapter 4 provides the performance analysis based on the GSPN model. In Chapter 5, an improved model is provided. Finally, Chapter 6 summarizes the contributions of this thesis and discusses the limitation of the current models. Future extensions of the current PN models are also to be discussed.

## CHAPTER 2

### Introduction of Petri Nets

#### 2.1 Basic Concept

Petri nets are a graphical and mathematical modeling tool applicable to many systems. They are a powerful instrument for the modeling and analysis of systems featuring process synchronization, choices, sequences, and competition among processes. A Petri Net (PN) comprises a set of places  $P$ , a set of transitions  $T$ . In its graphical representation, places are drawn as circles and transitions as bars. Places may contain tokens, drawn as black dots. Places are connected to the transitions by direct arcs. A place is an input of a transition if there is a direct arc from the place to the transition. Vice versa, a place is an output of a transition if there is a direct arc from the transition to the place.

#### 2.2 Definition

A generalized stochastic Petri net (GSPN) can be defined as an eight-tuple [9] [10]:

$$PN = (P, T, I, O, m, H, F, Pr)$$

where:

$P = \{P_1, P_2, \dots, P_m\}$  is a finite set of places.

$T = \{t_1, t_2, \dots, t_n\}$  is a finite set of transitions with  $P \cup T \neq \emptyset, P \cap T = \emptyset$ ;

$I: P \times T \rightarrow N$  and is an input function where  $N = \{0, 1, 2, \dots\}$ ;

$O: P \times T \rightarrow N$  and is an output function;

$m: P \rightarrow N$  and is a marking whose  $i^{\text{th}}$  component is the number of tokens in the  $i^{\text{th}}$  place. An initial marking is denoted by  $M_0$ ; and

$H: P \times T \rightarrow N$  and is an inhibitor function;

$F: T \rightarrow (0, \infty)$  is a vector whose  $i^{\text{th}}$  component is the exponential firing rate if the  $i^{\text{th}}$  transition is timed or otherwise undefined or  $\infty$  if the  $i^{\text{th}}$  transition is immediate.

$Pr: P \rightarrow R^{\sum_{t \in P^*} |P^*|}$ , such that  $\sum_{t \in P^*} Pr(t) = 1$  and  $Pr(t) \geq 0, \forall p \in P$ , where  $P^* = \{t \in P: t \text{ is an immediate transition}\}$ , and  $p = \{t \in T: I(p, t) \neq 0\}$ .

In the definition, the first five tuples define an ordinary Petri net.

### 2.3 Firing of a Transition

The behavior of many systems can be described in terms of system states and their changes. In order to simulate the dynamic behavior of a system, a state or marking in a Petri net is changed according to the following transition rules [1] [13]:

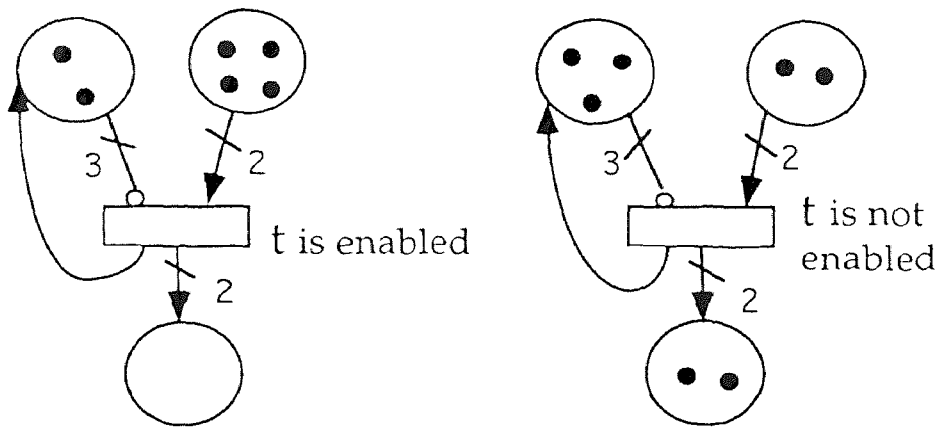
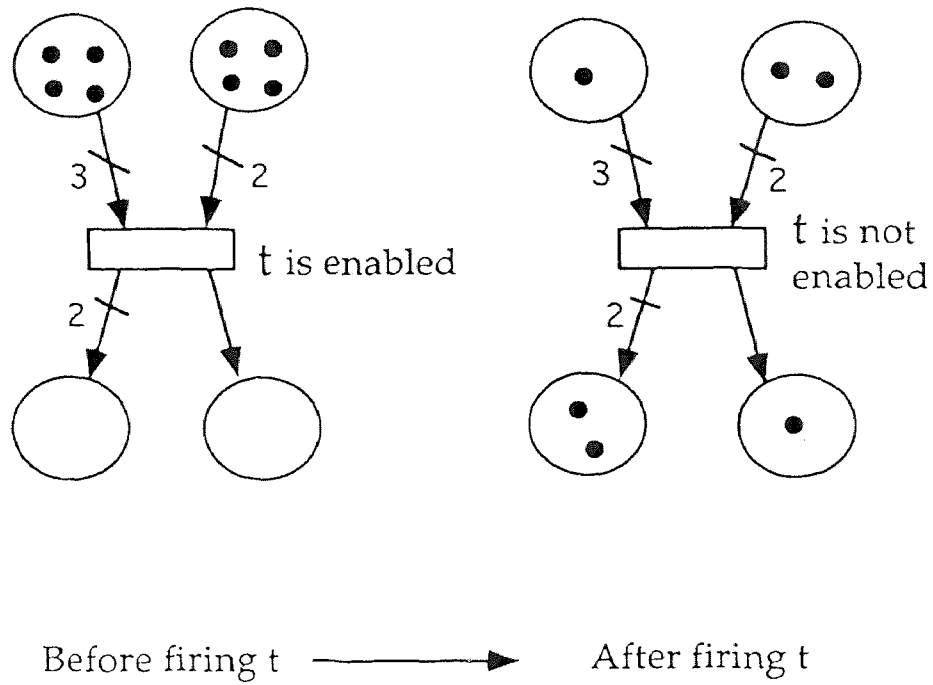
1. A transition  $t \in T$  is enabled if and only if,  $\forall p \in P, t \in T, m(p) \geq I(p, t)$  and if  $H(p, t) \neq 0, m(p) < H(p, t)$ ;
2. An enabled transition  $t$  may fire at marking  $m$ , yielding the new marking,  $m'(p_i) = m(p_i) + O(p_i, t) - I(p_i, t)$  for  $i=1, 2, \dots, n$ .

Figure 2-1 shows examples of firing rule for ordinary PN (5-tuple) and extended 8-tuple PN. After time is introduced into the ordinary PN, an enabled transition can fire only after the associated delay.

### 2.4 General Stochastic Petri Nets

There is no concept of time provided by the ordinary Petri nets, thus it is only possible to model the logical structure of the system and not its evolution in time. It is for this reason that the concept of time has been associated with a parameter that specifies the delay between the enabling of a transition and its firing resulting in timed transition Petri nets (TTPN's). There are three

Ordinary PN (5 tuples)



PN with Inhibitor Arcs

Figure 2-1 Petri Net Execution Rules

different classes of TTPN's: stochastic Petri nets (SPN's), generalized stochastic Petri nets (GSPN's), and deterministic and stochastic Petri nets (DSPN's).

Stochastic Petri nets feature each transition with an exponentially distributed delay. In generalized stochastic Petri nets, there are two kinds of transitions, timed and immediate. The immediate transitions are denoted as black bars. The timed transitions are represented by empty rectangular boxes and associated with exponentially distributed delay. When both immediate and timed transitions are enabled in a marking, only the former can fire, while the latter actually behave as if they were not enabled. It is necessary to specify a discrete probability distribution function to each of immediate transitions in conflict in order to select which transition gets fire. If several immediate transitions have input arcs from the same place, these transitions must have different probabilities in order to avoid their concurrent firing.

## 2.5 Properties of GSPN

The same behavioral properties can be defined for GSPNs as those for the ordinary Petri net [1]:

- **Safeness:** A place in a Petri net is a safe if the number of tokens in that place never exceeds one. A Petri net is safe if all of its places are safe.
- **Boundedness:** Boundedness is a generalization of safeness of a net with the situation that the places can hold a particular number of tokens. A place is  $k$ -bounded, if the number of tokens in that place cannot exceed an integer  $k$ . A Petri net is defined to be  $k$ -bounded if every place in it is  $k$ -bounded.
- **Liveness:** A transition is live, if and only if for any marking in the reachability set there is a firing sequence whose firing enables the

transition. A Petri net is live if each of its transition is live. Liveness of a Petri net implies freedom from deadlock.

- Reachability: Reachability is a fundamental basis for studying the dynamic properties of any system. A marking  $M_n$  is said to be reachable from a marking  $M_0$  if there exist a sequence of firings that transforms  $M_0$  to  $M_n$ .

## CHAPTER 3

### Protocol and Model Description

In this chapter, we first discuss the carrier sensing controller in optical ring network and network operation conditions. Then we provide an overview of protocol and define the assumption used to analyze the protocol. Finally, a generalized stochastic Petri net model is created.

#### 3.1 A Brief Review of CSMA

In many multiaccess systems, such as local area networks, a node can hear whether other nodes are transmitting after a very small propagation and detection delay relative to a packet transmission time. The detection delay is the time required for a physical receiver to determine whether or not some other node is currently transmitting. This delay differs somewhat from the delay, first, in detecting the beginning of a new transmission, second, in synchronizing on the reception of a new transmission, and third, in detecting the end of an old transmission.

If nodes can detect idle periods quickly, it is reasonable to terminate idle periods quickly and to allow nodes to initiate packet transmission after such idle detection. This type of strategy, called carrier sense multiple access (CSMA), does not necessarily imply the use of a carrier but simply the ability to detect idle periods quickly.

There are two kinds of CSMA techniques. They are CSMA/Collision Detection (CSMA/CD) and CSMA/Collision Avoid (CSMA/CA).

A number of nodes are all connected onto a common cable so that when one node transmits a packet (and the others are silent), all the other

nodes hear that packet. In addition, as in carrier sensing, a node can listen to the cable before transmitting. Because of the physical property of cable, it is possible for a node to listen to the cable while transmitting. Thus, if two nodes start to transmit almost at the same time, they will shortly detect a collision in process and both cease transmitting. On the other hand, if one node starts transmitting and no other nodes start before the first node's signal has propagated throughout the cable, the first node is guaranteed to finish its packet without collision. This technique is called CSMA/CD.

In order to reduce the token size in token ring and token bus networks, CSMA/Collision Avoid (CSMA/CA) is developed. One common approach is to replace the token with an implicit token represented by the channel becoming idle. When a node completes a packet transmission, it simply goes idle. The next node in sequence, upon detecting the idle channel, starts transmission if it has a packet or otherwise remains idle. Successive nodes in the sequence wait for successively longer times, to hear an idle, and start transmission, thus giving each of the earlier stations an opportunity to transmit if it has packets.

### 3. 2 The Carrier Sensing Controller

In the above section, we introduce the CSMA protocol. This section is for the implementation of a carrier sensing controller in CSMA/RN. Figure 3-1 illustrates the characteristics of a station in the carrier sensed ring network. First, the incoming signal is split into two streams. One enters the delay area, and the other one goes to the controller. The delay can be created by a piece of optical fiber line. For example, when the channel capacity is 1Gbps, a 100 bit delay is created by a 20 (m) piece of fiber: Delay time =  $\frac{100 \text{ bits}}{1 \text{ Gbps}} = 100 \text{ (nanosec)}$ .



## CSMA/RN Logic Diagram

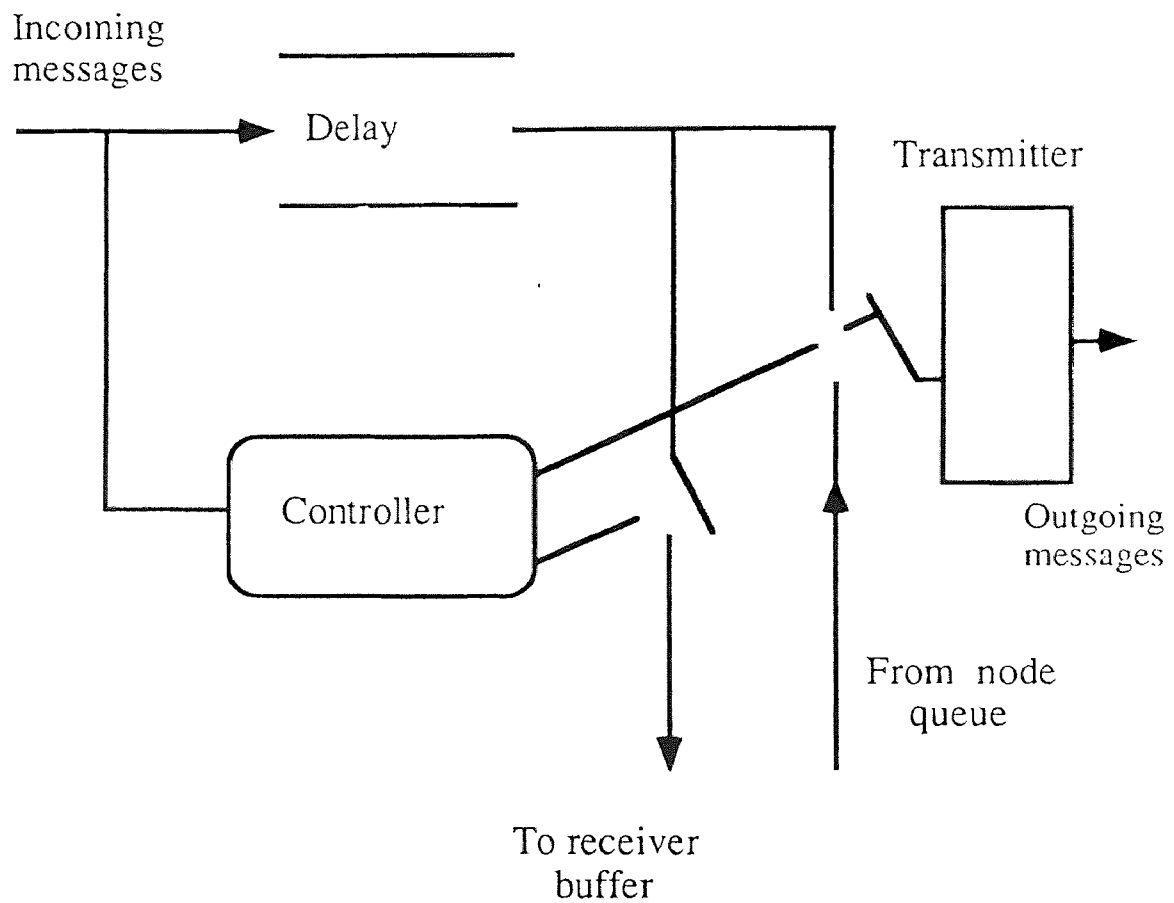


Figure 3-1 CSMA/RN Logic Diagram [7]

The controller is required to decide what to do within the delay time. When it detects that there is no incoming data, the station is allowed to transmit its queue if it is not empty. Whenever the controller detects the presence of incoming data, it will decide whether the packet is destined for this station or not. If the destination of the incoming packet is right for this station, the controller just puts the data into the receiving buffer of the station, and offers service to its queue at the same time. Otherwise, it must pass the incoming information to the next station on the ring.

The most important thing here is the speed match between the delay time and the response time of the controller.

### 3.3 Protocol Description

By implementing the carrier sensing control unit into each station, the network operation condition is shown by Figure 3-2.

The state of the ring is represented by two kinds of packets. One is the data packet, that represents the busy state of the ring. The other one is the empty packet, which represents the idle state of ring. Each station in the ring is able to decide either transmitting packets (empty packet for idle station, data packets for waiting station) or receiving data packet by sensing the state of the ring.

Protocol description and conditions:

- Packets are removed at the destination and the empty space used by this or a subsequent station to transmit messages in queues.
- Stations are uniformly spaced around the ring.
- Messages arrivals are uniformly distributed among the stations.
- Each station is allowed to transmit messages whenever an empty packet is passing or a packet is removed at this station.

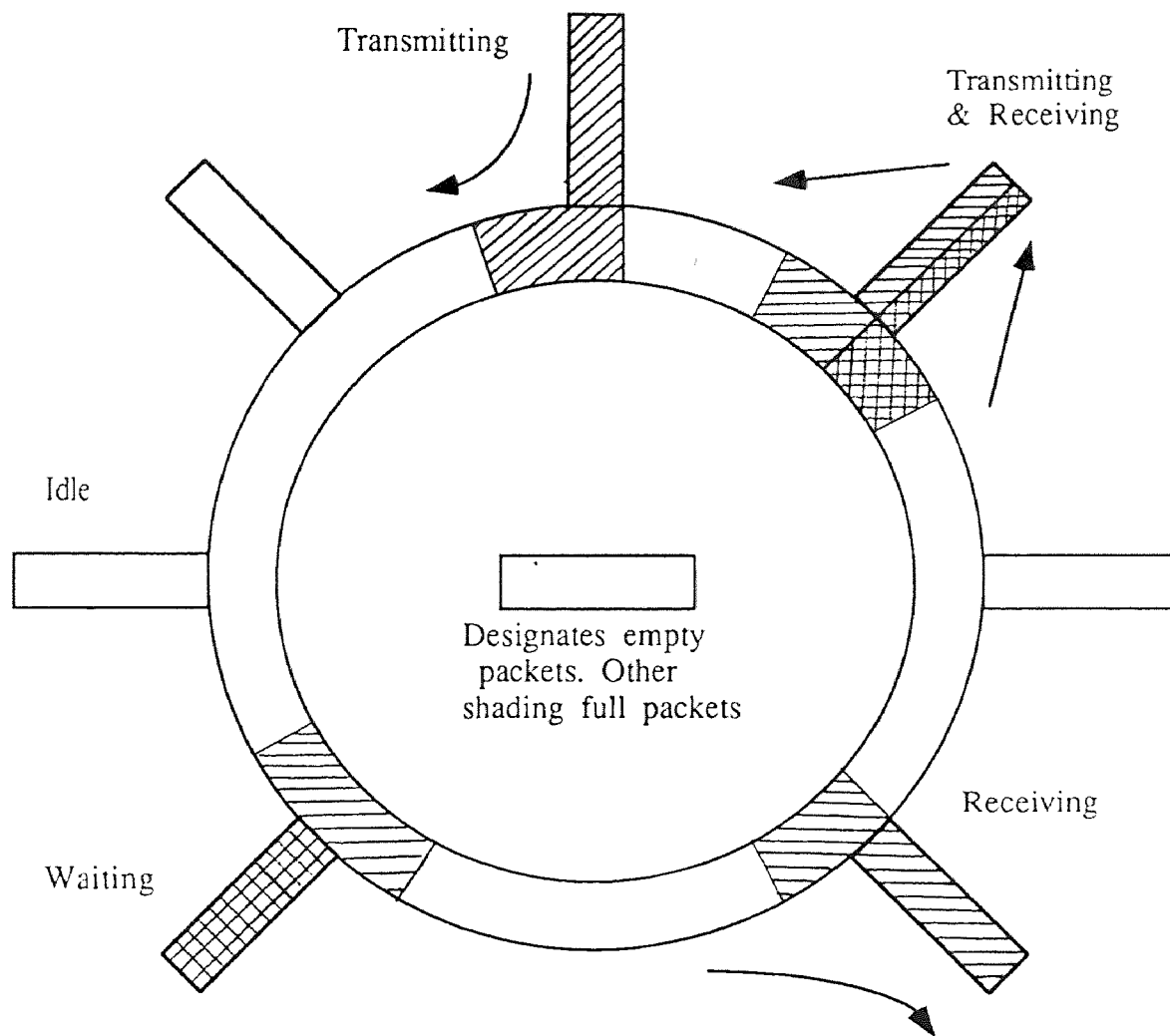


Figure 3-2 Network Operation Condition [7]

The above description for each station is very simple and the logic operations at each station are independent, when a number of stations work together in a network, the situation becomes very complex. When the load of the network increases, the empty space on the ring tends to fracture. It is quite difficult to express the whole situation.

### 3. 4 GSPN Model for CSMA/RN

The model we created is a generalized stochastic Petri net. There are two kinds of transitions in the model, timed and immediate. The network to be modeled is a CSMA/RN. Let's first consider how many kinds of delays are involved in a packet transmission. As we assume that the arrival process is a Poisson process with the rate  $\lambda$  (packets/ $\mu$ sec) and service rate is distributed with an average value of  $\mu = C/L$  ( $L$  denotes the average packet length,  $C$  is channel capacity), both of them can be represented by timed transitions.

Once a packet enters the ring, the propagation delay is deterministic, the GSPN can not be used to model this kind of delay. Thus we have to find a way to avoid it. In other words, a ring-like GSPN model is not possible to make as is done for the token-bus LAN using GSPNs [19].

According to the protocol of CSMA/RN, messages in the ring have the highest priority . Once a packet enters the ring, it will keep on traveling until it reaches the destination. Two states are enough to represent the ring (idle and busy). When you keep an eye on any point of the ring, the appearance of messages can be viewed as a Poisson process with an unknown rate. All stations connected to the ring have the same lower priority, they decide whether or not to transmit a packet depended on the state of the ring when their queues are not empty. The carrier sensing controller is responsible to split the messages into two parts when they are passing a station. One part is

destined for this station and the other part is just passing it. In our model, two immediate transitions with probabilities are used to model the two possibilities. The operations at each station are same and independent.

Based on the above analysis, to build an analytical model for CSMA/RN operation based on discrete event analysis model for this network, only a single station need be modeled since logic operations at each station are independent.

The message traffic of the ring can be represented by a Poisson arrival process based on the network load. Figure 3-3 shows the GSPN model. The single station is modeled by  $p_1, p_2, t_1 - t_3$ . The ring is modeled by  $p_3 - p_7, t_4 - t_8$ .

The model description:

$p_1$ : Buffer of a station.

$p_2$ : Queue of a station.

$p_3$ : Represents an incoming packet destined for this station when  $p_3$  holds a token.

$p_4$ : There is no message passing the station when it holds a token.

$p_5$ : Represents that there is a data packet passing the station when it holds a token.

$p_6$ : Represents the state of ring with respect to the station. Holding a token means the portion of the ring at the station is idle.

$t_1$ : Associated with the packet arrival rate  $\lambda$ .

$t_2$ : Station server, enabled when the ring is idle and the queue of the station is not empty.

$t_3$ : Station server with rate  $\mu$ , enabled when the ring is busy and the queue of the station is not empty.

$t_4$ : Associated with a time delay to the packet destined for this station.

$t_5$ : Associated with a time delay to the packet passing this station.

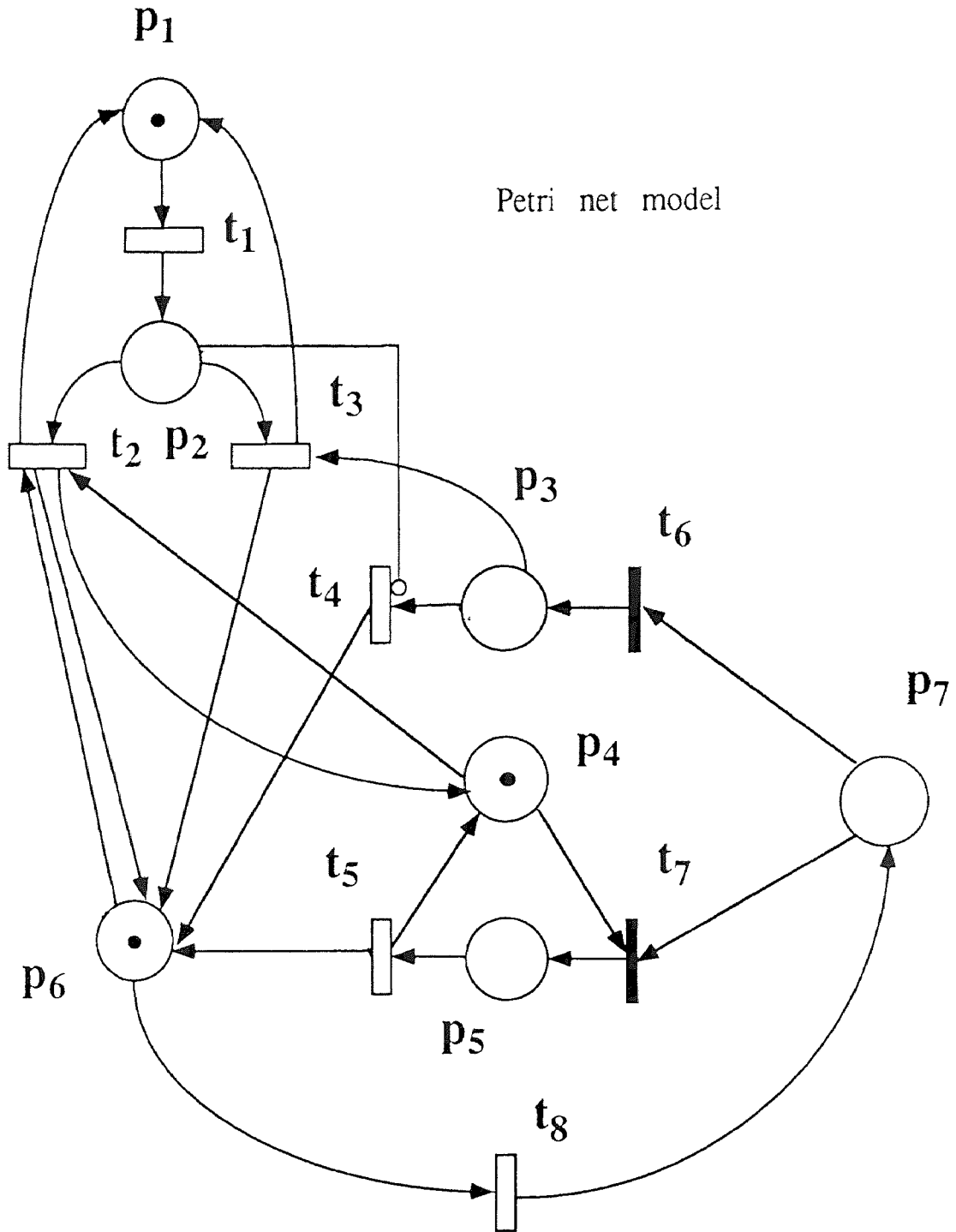


Figure 3-3 Generalized Stochastic Petri Net Model

$t_5$ : Associated with a time delay to the packet passing this station.

$t_6$ : Immediate transition whose firing probability is  $1/N$ , implying 1 out of  $N$  messages is destined to this station.

$t_7$ : Immediate transition whose firing probability is  $(N-1)/N$ , implying  $(N-1)$  out of  $N$  messages are passing the station.

$t_8$ : The ring arrival rate associated with this transition.

**Table 3-1 Parameters Associated with Transitions**

Transition	Parameter Value
$t_1$	Timed with firing rate $\lambda$
$t_2$	Timed with rate to be determined
$t_3$	Timed with firing rate $\mu$
$t_4$	Timed with rate to be determined
$t_5$	Timed with rate to be determined
$t_6$	Immediate with firing probability $1/N$
$t_7$	Immediate with firing probability $(N-1)/N$
$t_8$	Timed with ring arrival rate to be determined

### 3. 5 The Model Operation

Initially, there is one token in both places  $p_4$  and  $p_6$ , i.e.,  $m(p_4) = m(p_6) = 1$ , respectively. The queue of the station is not empty, i.e.,  $m(p_2) \geq 1$ .  $t_2$  and  $t_8$  are enabled. If the idle state keeps long enough, one token is sent back to  $p_1$ , that means one packet is transmitted. Otherwise,  $t_8$  fires, that means the station senses an incoming data packet. Once  $t_8$  fired, two things might happen.

1. There is a data packet destined for this station.
2. The coming data packet is passing the station.

For the first case, one token arrives at  $p_3$ ,  $m(p_3) = 1$ , and  $t_3$  is enabled. After the firing of  $t_3$ , a token returns to place  $p_6$ ,  $m(p_6) = 1$ . This process means the station receives a data packet and transmits a data packet from its queue at the same time, and after this process, the ring becomes idle again.

In this case, if the queue of station is empty,  $m(p_2) = 0$ ,  $t_4$  is enabled and fired, that means the station has received a data packet and nothing transmitted.

For the second case, one token is sent to place  $p_5$ ,  $m(p_5) = 1$ ,  $t_5$  is enabled. After the firing of  $t_5$ , a token returns to place  $p_6$  and place  $p_4$  respectively. Ring becomes idle, waiting for the next round of competition between  $t_2$  and  $t_8$ . If the station queue is empty, then  $t_2$  is disabled and  $t_8$  is enabled. This implies the fact that if the stations has nothing to transmit, it does not care whether the portion of the ring at the station busy or not. When a packet destined for this station,  $t_4$  is enabled and fired. This means the station can only receive messages from other stations.



## CHAPTER 4

### Performance Analysis

In the last chapter, we have created a stochastic Petri net model. This chapter discusses the parameters which are associated to the model we created based on the assumptions and conditions discussed in Chapter 3. Then discuss the performance analysis results.

#### 4.1 Analysis Assumptions and Conditions

Protocol assumptions:

1. The arrival processes to each station are statistically equivalent Poisson processes with average arrival rate  $\lambda$  (packets/ $\mu$ sec).
2. Number of stations is  $N$ .
3. The packet length is exponentially distributed with an average length of  $L$ .
4. Channel capacity is  $C = 1$  Gbps.
5. Average service rate,  $\mu$ , is supposed to be Poisson distributed with average value  $\mu = C/L$  (packets/ $\mu$ sec),
6. Offered load,  $\rho$  is defined as  $\rho = N \lambda / \mu$  where  $N$  is the number of stations,  $\lambda$  is average arrival rate, and  $\mu$  is average service rate.

Protocol Conditions have been discussed in Section 3.3.

Based on these assumptions and conditions, an intuitive analysis can be given in the following paragraphs.

At any point of the ring, the arrival packets can be viewed as a statistically equivalent Poisson processes with an unknown rate  $\lambda'$ . When the ring connects  $N$  station, in a random chosen point  $A$ , as shown in Figure 4-1,  $1/N$  of packets transmitted by Station 1 go through point  $A$ ,  $2/N$  of packets

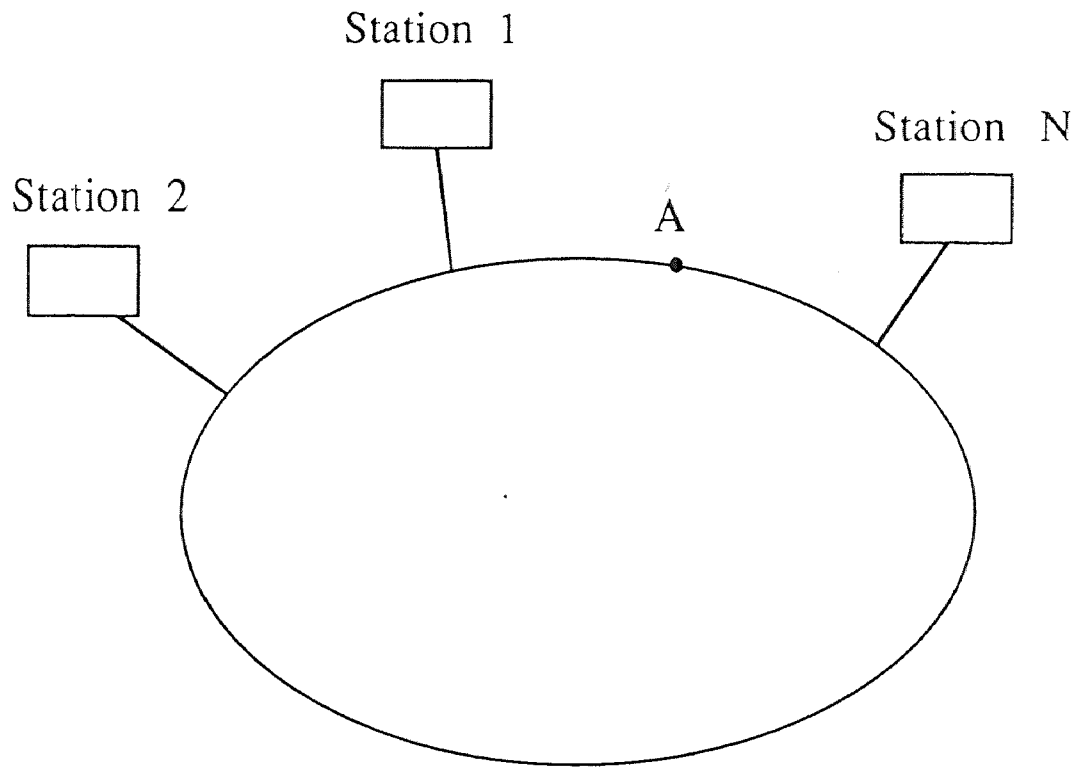


Figure 4-1 Ring Throughput

transmitted by Station 2 go through point A, ..., N/N of packets transmitted by Station N pass point A. thus at point A, throughput is:

$$\begin{aligned}
 H &= \left( \frac{1}{N} + \frac{2}{N} + \frac{3}{N} + \dots + \frac{N}{N} \right) \cdot H' \\
 &= \left( \frac{1+2+3+\dots+N}{N} \right) \cdot H' \\
 &= \frac{1}{N} \cdot \frac{N(N+1)}{2} \cdot H' \\
 &= \frac{N+1}{2} \cdot H' \tag{4.1}
 \end{aligned}$$

Where H represents the throughput of ring, H' represents the throughput of a station. The throughput is defined as number of packets per  $\mu\text{sec}$ .

By this calculation, we know that the throughput of ring is a fixed number depending on throughput of stations, and the throughput of each stations depends on the offered load.

The next step is to find out what is the maximum offered load before the queue of each station is accumulated. This means when the offered load less than some value, the queues of stations are always very short. Then the throughput of stations  $H' \approx \lambda$ .

By definition, the arrival rate of each station  $\lambda$ :

$$\lambda = \frac{\rho \cdot \mu}{N} \tag{4.2}$$

Throughput of a station:

$$H' = \frac{\rho \cdot \mu}{N} \quad (4.3)$$

$$\text{The ring throughput: } H = \frac{N+1}{2} \cdot \frac{\rho \cdot \mu}{N} \leq \mu \quad (4.4)$$

$\mu$  is the maximum number of packets the channel can handle.

From equation (4.4), we have:

$$\rho \leq \frac{2N}{N+1} \quad (4.5)$$

$$\text{or } N \geq \frac{\rho}{2-\rho} \quad (4.6)$$

When  $N=10$ ,  $\rho \leq 20/11 \approx 2$ , it is very close to two. This shows that the network can handle a load fraction near to two, when  $N \rightarrow \infty$ ,  $\rho = 2$ . This means the more stations connected in the ring, a better overall performance can be obtained.

#### 4.2 An Example to Run SPNP Program

In this section, we give an example of running SPNP program. Let packet length  $L = 2k$ ,  $N = 10$ , channel capacity  $C = 1$  Gbps, mean server rate  $\mu = C/L = 0.5$  packets/ $\mu\text{sec}$ , load fraction  $\rho = 1.5$ , and initial model marking  $m_0 = (120, 0, 0, 1, 0, 1, 0)^T$ . Table 4-1 shows all the parameters associated to the model.

After starting the program, assign two guessed value to  $t_2$  (server rate) and  $t_8$  (ring arrival rate) by experience. At first, the two value we guess are 0.3

Table 4-1 Parameters list

Transition	Parameter Value
$t_1$	Timed with firing rate $\lambda = \rho * \mu / N = 0.075$
$t_2$	Timed with rate to be determined
$t_3$	Timed with firing rate $\mu = C / L = 0.5$
$t_4$	Timed with firing rate 10
$t_5$	Timed with firing rate 10
$t_6$	Immediate with firing probability 0.1
$t_7$	Immediate with firing probability 0.9
$t_8$	Timed with ring arrival rate to be determined

and 0.3. Then we obtain result-1 as shown below:

INPUT: Number of stations: = 10  
 INPUT: Channel capacity(Gbps): = 1  
 INPUT: Ave. length of packet(k): = 2  
 INPUT: Load fraction of ring: = 1.5  
 INPUT: Ring arrival rate: = 0.3  
 INPUT: Guessed service rate: = 0.3  
 EXPECTED: throughput = = 0.0750002174963  
 EXPECTED: Average queue size = = 0.32628532141

```

=====
PLACE                Pr[nonempty]          Av[tokens]
0: p1                1.000000000000e+00   1.196737146786e+02
1: p2                2.469818393207e-01   3.262853214095e-01
2: p3                1.611072143516e-02   1.611072143516e-02
3: p4                9.741333854007e-01   9.741333854007e-01
4: p5                2.586661459927e-02   2.586661459927e-02
5: p6                9.580226639656e-01   9.580226639656e-01
6: p7                0.000000000000e+00   0.000000000000e+00
  TRANSITION        Pr[enabled]          Av[throughput]
0: t1                1.000000000000e+00   7.500000000000e-02
1: t2                2.267785264575e-01   6.803355793724e-02
4: t5                2.586661459927e-02   2.586661459927e-01
7: t8                9.580226639656e-01   2.874067991897e-01
=====

```

From this result, the throughput of station is 0.075 packets/ $\mu$ sec and throughput of ring ( $t_8$ ) is 0.2874 packets/ $\mu$ sec. By equation 4.7,  $0.075 \times 5.5 = 0.4125$  packets/ $\mu$ sec, calculated value is greater than 0.2874 packets/ $\mu$ sec. Thus the guessed values of  $t_8$  is too low and  $t_2$  is too high. We have to increase firing rate of  $t_8$  and decrease firing rate of  $t_2$  by 0.1. This means 0.4 packets/ $\mu$ sec is associated with  $t_8$  and 0.2 packets/ $\mu$ sec is associated with  $t_2$ . Then we obtain result-2 as shown below:

```

INPUT: Number of stations: = 10
INPUT: Channel capacity(Gbps): = 1
INPUT: Ave. length of packet(k): = 2
INPUT: Load fraction of ring: = 1.5
INPUT: Ring arrival rate: = 0.4
INPUT: Guessed service rate: = 0.2
EXPECTED: throughput = = 0.075000464693
EXPECTED: Average queue size = = 0.527427760165

```

```

=====

```

PLACE	Pr[nonempty]	Av[tokens]
0: p1	1.000000000000e+00	1.194725722398e+02
1: p2	3.484256022360e-01	5.274277601648e-01
2: p3	2.779927630741e-02	2.779927630741e-02
5: p6	9.384176838039e-01	9.384176838039e-01
6: p7	0.000000000000e+00	0.000000000000e+00
TRANSITION	Pr[enabled]	Av[throughput]
0: t1	1.000000000000e+00	7.500000000000e-02
1: t2	3.117244149155e-01	6.234488298309e-02
2: t3	2.531116341985e-02	1.265558170993e-02
7: t8	9.384176838039e-01	3.753670735216e-01

```

=====

```

Based on Result-2, the throughput of station is still 0.075 packets/ $\mu$ sec and throughput of ring ( $t_8$ ) is 0.3753 packets/ $\mu$ sec. By equation 4.7,  $0.075 \times 5.5 = 0.4125$  packets/ $\mu$ sec, calculated value is greater than 0.3753 packets/ $\mu$ sec. So the guessed values of  $t_8$  is too low and  $t_2$  is too high. This time we increase firing rate of  $t_8$  by 0.05 and decrease firing rate of  $t_2$  by 0.1. This means 0.45 packets/ $\mu$ sec is associated with  $t_8$  and 0.1 packets/ $\mu$ sec is associated with  $t_2$ . Then we obtain result-3 as shown below:

INPUT: Number of stations: = 10  
 INPUT: Channel capacity(Gbps): = 1  
 INPUT: Ave. length of packet(k): = 2  
 INPUT: Load fraction of ring: = 1.5  
 INPUT: Ring arrival rate: = 0.45  
 INPUT: Guessed service rate: = 0.1  
 EXPECTED: throughput = = 0.0750011973721  
 EXPECTED: Average queue size = = 1.35819634775  
 AVERAGE

```

=====
PLACE                Pr[nonempty]                Av[tokens]
0: p1                1.000000000000e+00        1.186418036523e+02
1: p2                5.840977050086e-01        1.358196347748e+00
2: p3                4.851461895420e-02        4.851461895420e-02
5: p6                9.144501484246e-01        9.144501484246e-01
6: p7                0.000000000000e+00        0.000000000000e+00
TRANSITION          Pr[enabled]                Av[throughput]
0: t1                1.000000000000e+00        7.500000000000e-02
1: t2                5.163299044667e-01        5.163299044667e-02
2: t3                4.673641385084e-02        2.336820692542e-02
7: t8                9.144501484246e-01        4.115025667911e-01
=====
  
```

Based on Result-3, the throughput of station is 0.075 packets/ $\mu$ sec and throughput of ring ( $t_8$ ) is 0.4115 packets/ $\mu$ sec. By equation 4.7,  $0.075 \cdot 5.5 = 0.4125$  packets/ $\mu$ sec, calculated value is very close to the output result. The deviation = 0.001, implying this result is good enough. The average queue size is then 1.36. By Little's Law Theorem, we obtain the following result:

$$\begin{aligned}
 E\{T\} &= \frac{E\{n\}}{\lambda} \\
 &= \frac{E\{n\} \cdot N}{\rho \cdot \mu} \\
 &= \frac{1.36 \cdot 10}{1.5 \cdot 0.5} \approx 18 \mu\text{sec}
 \end{aligned}$$

### 4.3 The Description of Petri Net Model Parameters

We have already created the GSPN model for the CSMA/RN. The derived Petri net model is first translated to C-based code. The SPNP package [4] was developed at Duke University is used to derived the throughput and delay for the system with different offered load.

In this model, the sum of the throughputs of  $t_2$  and  $t_3$  represents the throughput of a station, throughput of  $t_8$  represents the throughput of the ring. For an N station network, the throughput relationship between N stations and the ring is:

$$H = \frac{N + 1}{2} \cdot H' \quad (4.7)$$

Considering this point, we have to assign different firing rate to  $t_2$  and  $t_8$ . It is very difficult to know what firing rate should be correct for these two transitions, because they both follow the load of the ring.

When we start to run the program, two guessed value of firing rates are assigned to  $t_2$  and  $t_8$ . Both of them can not exceed the value  $C/L$  which is the maximum number of packets the ring can handle. At a low load condition, the firing rate of  $t_2$  is a relatively large value. The firing rate of  $t_8$  should be less than  $C/L$  because at the low load condition, the stations have more chances to transmit messages. The throughput of the ring is low, if a too large value is assigned to  $t_8$ , the throughput of  $t_8$  will be too high to fit the relationship between the throughput of the station and ring. At the heavy load condition, the situation is different. The stations have less and less chance to transmit messages in their queues. The throughput of  $t_2$  will decrease significantly and the throughput of ring will increase near to  $C/L$ , the maximum firing rate of the ring. So the firing rate of  $t_2$  will decrease with



increasing of the load. The firing rate of  $t_8$  will increase until it reaches the maximum value  $C/L$ , then fixed at this value. We decrease the firing rate of  $t_2$  until Equation 4.7 is matched, this is the result we are looking for.

1.  $t_1$  is associated with the arrival rate  $\lambda$ , its value depends on the offered load.

By definition,  $\lambda = \frac{\rho \cdot \mu}{N}$  where  $\mu$  is service rate.

2.  $t_2$  is a guessed value depending on the offered load.

3.  $t_3$  is associated with service rate,  $\mu = C/L$ .

4.  $t_4$  and  $t_5$  are associated with a large firing rate.

5.  $t_6$  and  $t_7$  are immediate transitions,  $t_6$  is associated with a probability  $1/N$ , and  $t_7$  with a probability of  $(N-1)/N$ . That means the incoming messages are split into two parts.  $1/N$  part of messages is destined for this station and  $(N-1)/N$  parts of messages are passing it.

6.  $t_8$  is a guessed value depending on the offered load.

**Table 4-2 Parameters list**

Transition	Parameter Value
$t_1$	Timed with firing rate $\lambda = \rho \cdot \mu / N$
$t_2$	Timed with rate to be determined
$t_3$	Timed with firing rate $\mu = C / L$
$t_4$	Timed with firing rate 10
$t_5$	Timed with firing rate 10
$t_6$	Immediate with firing probability 0.1
$t_7$	Immediate with firing probability 0.9
$t_8$	Timed with ring arrival rate to be determined

The maximum number of tokens that can be put initially to a place in SPNP Software Package is 120. Thus the maximum buffer size is 120 and the maximum queue of each station is also 120. Thus, the initial marking is  $m_0 = (120, 0, 0, 1, 0, 1, 0)^T$ . Based on the above discussion, we obtain the SPNP source code as shown in Appendix A.

#### 4.4 Analysis Result V.S. Different Offered Load

In this section, we assume the average packet length is 2k, so  $\mu = C/L = 0.5$  (packets/ $\mu$ ). The offered load  $\rho$  is from 0 to 2.0. The number of stations are 10.

The analysis results v.s. different offered load for packet size of 2k and 10-station network are shown in Figure 4-2. The average delay is increased with increasing of the offered load. The delay is very low when the load fraction is less than 1.5 and the delay increases very fast after the offered load over 1.5. For example, when  $\rho = 1.75$ ,  $E(T) = 1234$  ( $\mu\text{sec}$ ). From Equation 4.5:

$$\rho \leq \frac{2N}{N+1} \quad (4.8)$$

Table 4-3 Parameters for GSPN Model

Transition	Parameter Value
$t_1$	Timed with firing rate $\lambda = 0.05\rho$
$t_2$	Timed with rate to be determined
$t_3$	Timed with firing rate $\mu = C/L = 0.5$
$t_4$	Timed with firing rate 10
$t_5$	Timed with firing rate 10
$t_6$	Immediate with firing probability 0.1
$t_7$	Immediate with firing probability 0.9
$t_8$	Timed with ring arrival rate to be determined

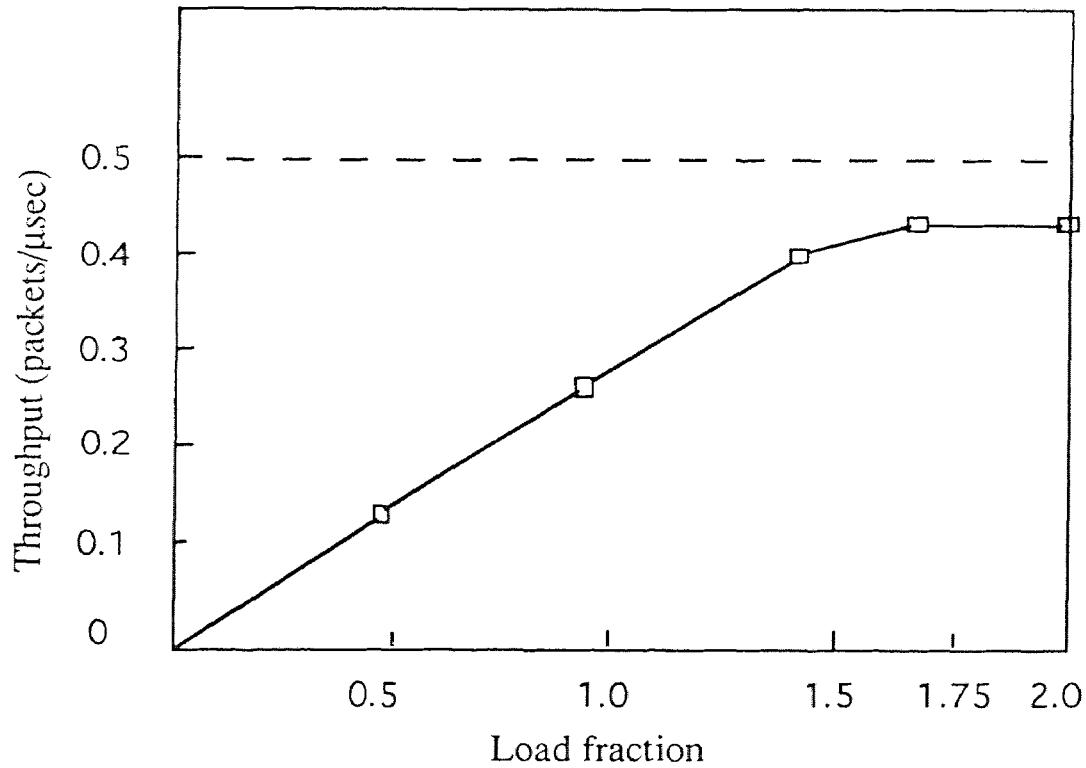
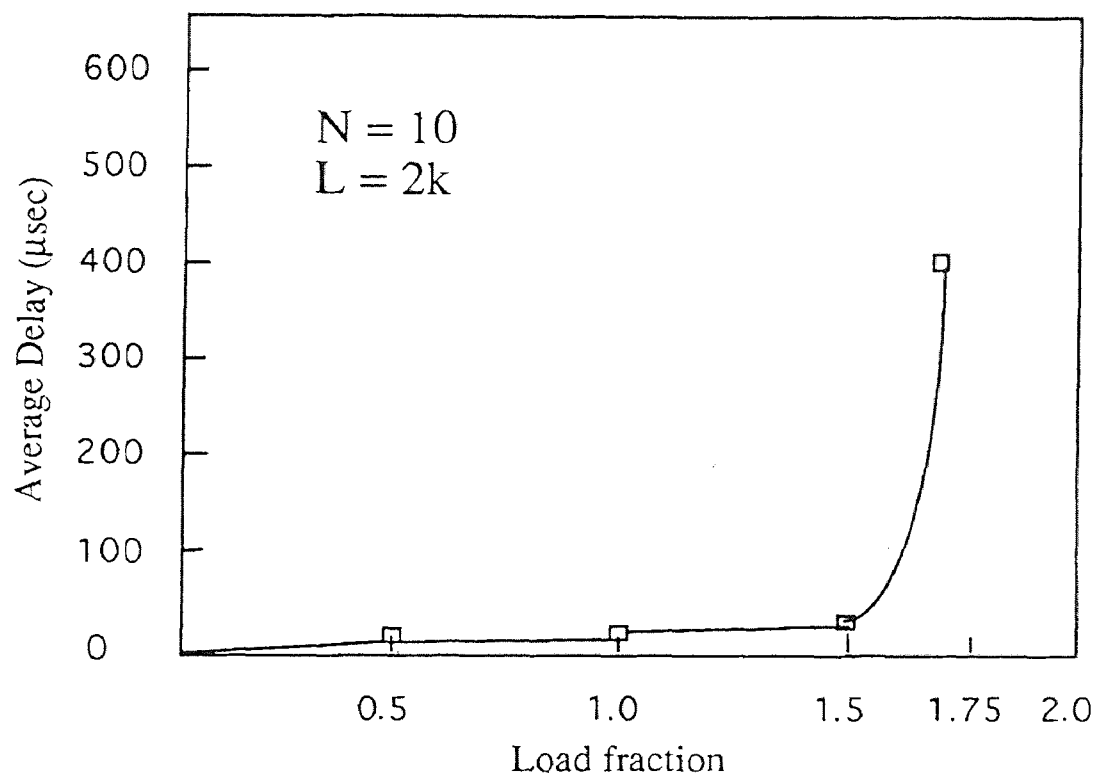


Figure 4-2 Analysis Result V.S. Different offered Load

When  $N = 10$ ,  $\rho < 1.8$ , so our result is very close to the analysis given in Section 4.1. The throughput keeps on increasing until it reach 0.437 (packets/ $\mu$ sec). Over 87% of the ring capability (0.5 packets/ $\mu$ sec is the highest) is utilized.

#### 4.5 Analysis Result V.S. Different Numbers of Stations

In this section, we compare the performance of the network between  $N = 10$  and  $N = 20$  based on average packet length  $L = 2k$ .

Table 4-4 Parameters for GSPN Model

Transition	Number of Stations $N = 10$	Number of Stations $N = 20$
$t_1$	$\lambda = 0.05\rho$	$\lambda = 0.025\rho$
$t_2$	To be determined	To be determined
$t_3$	0.5	0.5
$t_4$	10	10
$t_5$	10	10
$t_6$	0.1	0.05
$t_7$	0.9	0.95
$t_8$	To be determined	To be determined

Figure 4-3 shows the result. The delay performance for  $N = 10$  and  $N = 20$  is almost the same when load fraction is less than 1.5. When load fraction is greater than 1.5, the performance of network connected 20 stations is much better than that of the network connected 10 stations. From equation 4.5, we have  $\rho < 1.95$  for  $N = 20$ . Thus the SPNP result is very close to the analysis

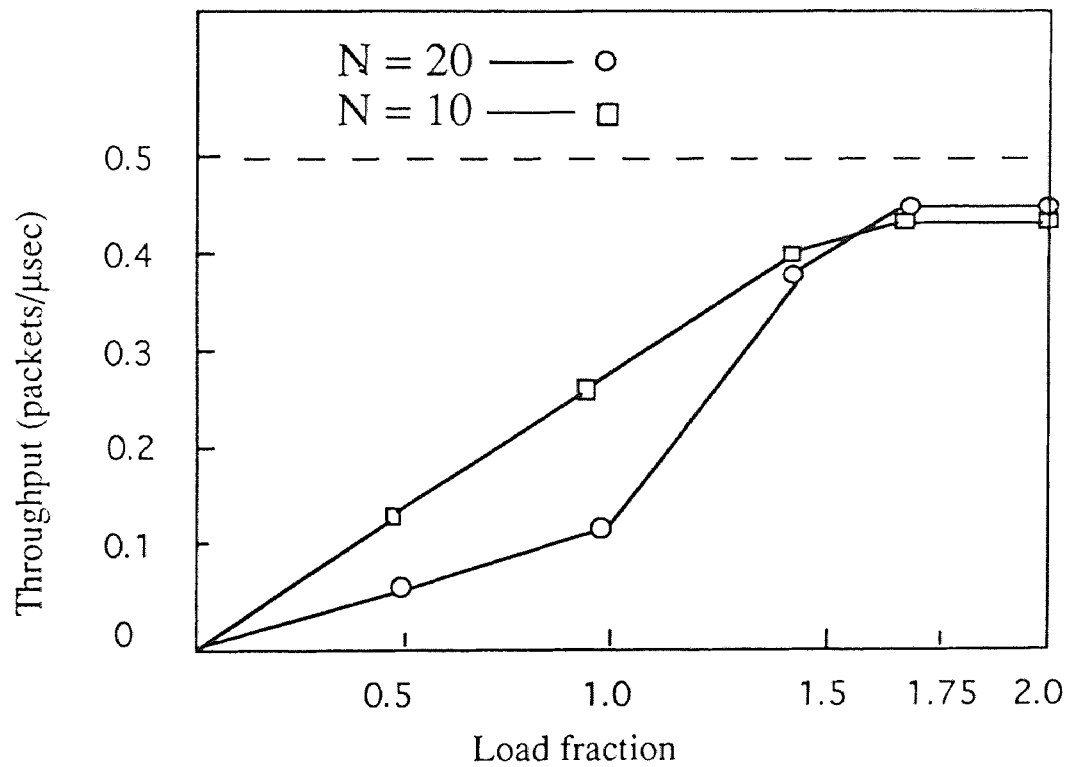
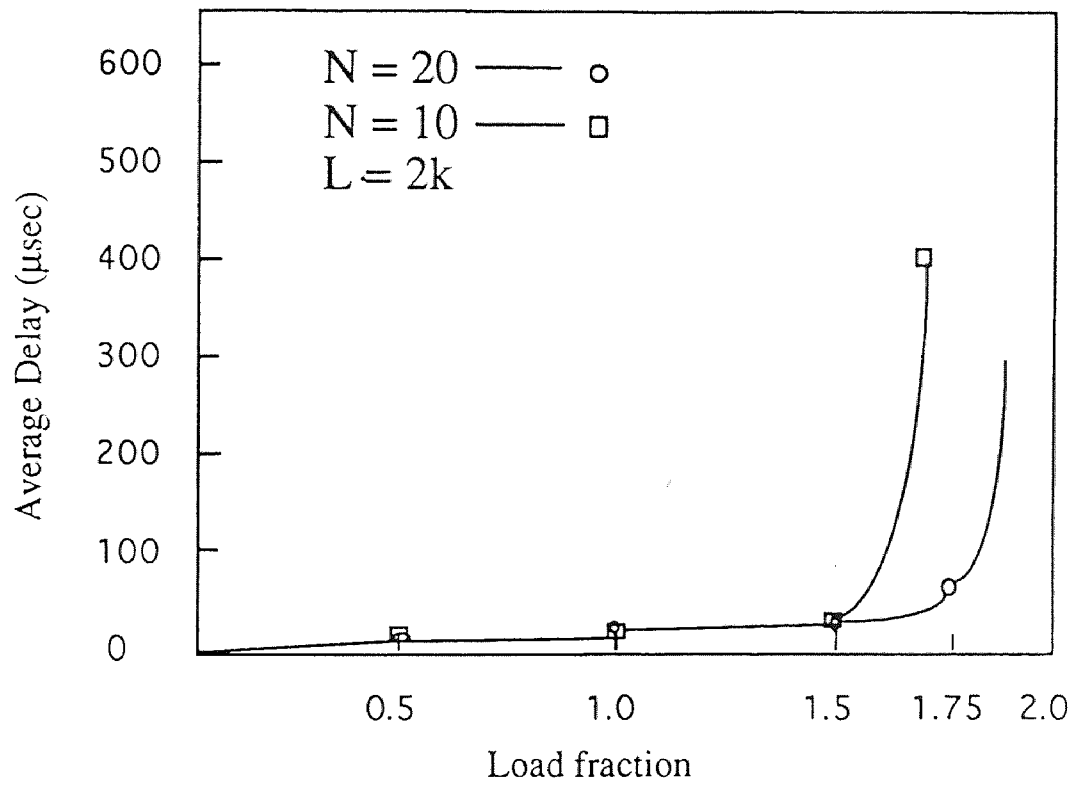


Figure 4-3 Analysis Result V.S. Different Number of Stations

result given at Section 4.1.

For the heavy load situation, the performance of throughput for  $N = 20$  is better than that for  $N = 10$ . For example, when  $\rho = 1.75$ , the throughput is 4.61 (packets/ $\mu$ sec) for  $N = 20$  and the throughput is 4.37 (packets/ $\mu$ sec) for  $N = 10$ .

#### 4.6 Analysis Result V.S. Different Average Packet Size

In this section, we compare the performance v.s. different average packet size of 2k, 4k, 8k with number of stations  $N = 10$ .

Table 4-5 Parameters for GSPN Model

Transition	Packet Size L = 2k	Packet Size L = 4k	Packet Size L = 8k
$t_1$	$\lambda = 0.05\rho$	$\lambda = 0.025\rho$	$\lambda = 0.0125$
$t_2$	To be determined	To be determined	To be determined
$t_3$	0.5	0.25	0.125
$t_4$	10	10	10
$t_5$	10	10	10
$t_6$	0.1	0.1	0.1
$t_7$	0.9	0.9	0.9
$t_8$	To be determined	To be determined	To be determined

Figure 4-4 shows the result. The analysis result shows that the performance of network decreases with the increasing of packet size. Because it takes longer time to transmit a larger packet. when the offered load increases, the empty packet tends to fracture. The probability for stations to catch a big enough empty packet for their queued customer becomes less and less. This is the reason that makes the difference.

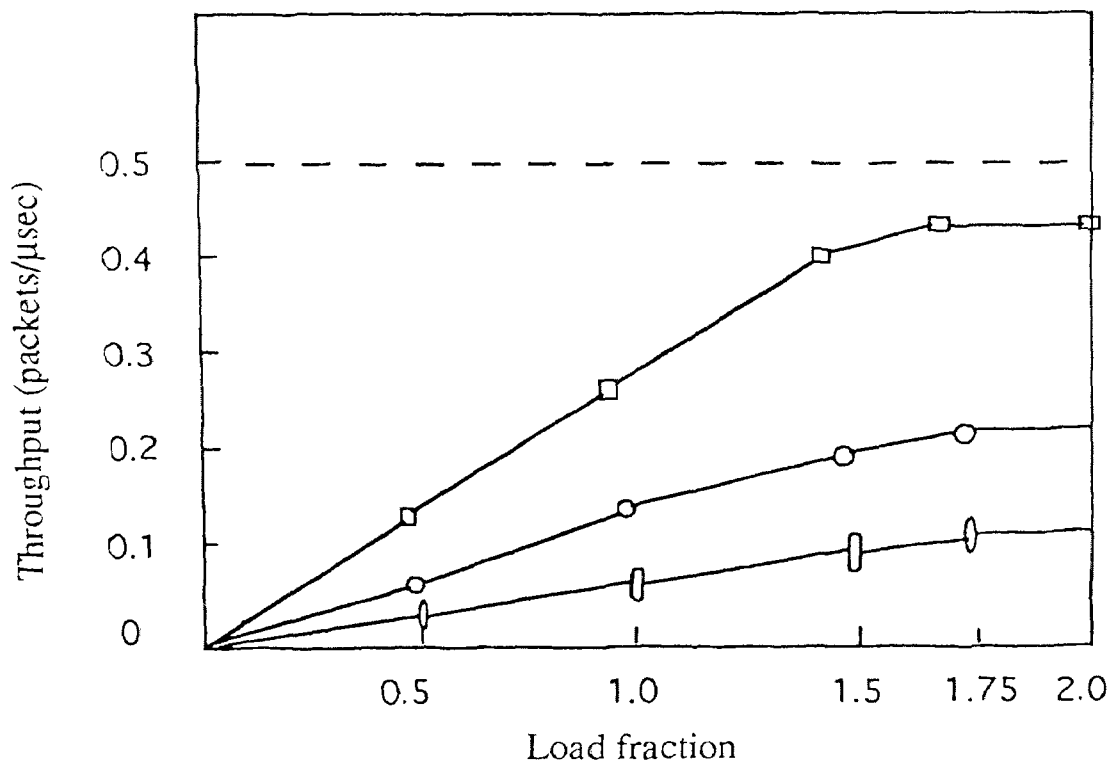
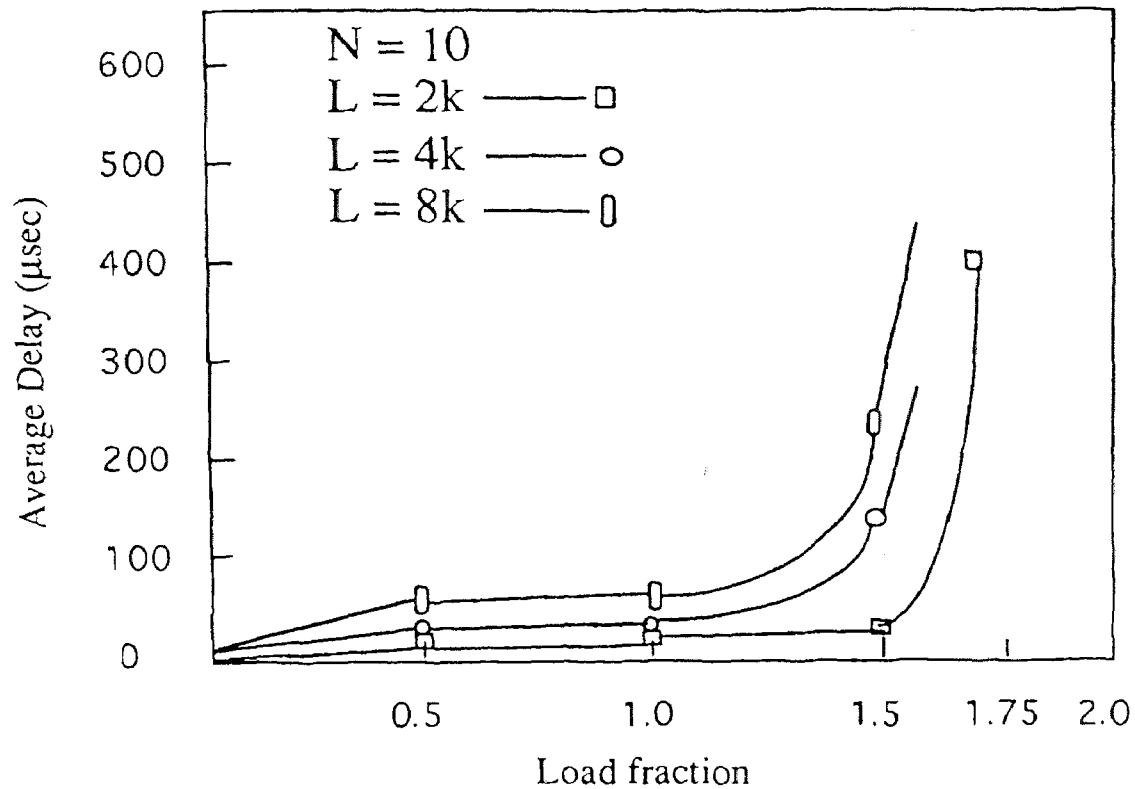


Figure 4-4 Analysis result V.S. Different Average Packet Size

#### 4.7 Analysis Result for Small Buffer Size

In many network applications, the buffer size is small because an unlimited buffer is not the case. In this section, the buffer size is assigned to 5. The average packet length  $L = 2k$ . Number of stations  $N = 10$ . We are going to compare the results between the buffer size of 120 and the buffer size of 5. Thus, the initial marking  $m_0 = (5, 0, 0, 1, 0, 1, 0)^T$ .

Table 4-6 Parameters for GSPN Model

Transition	Parameter Value
$t_1$	Timed with firing rate $\lambda = 0.5\rho$
$t_2$	Timed with rate to be determined
$t_3$	Timed with firing rate $\mu = C/L = 0.5$
$t_4$	Timed with firing rate 10
$t_5$	Timed with firing rate 10
$t_6$	Immediate with firing probability 0.1
$t_7$	Immediate with firing probability 0.9
$t_8$	Timed with ring arrival rate to be determined

The analysis result is shown in Figure 4-5. When buffer size is 5, no matter how high the load fraction is, the average delay is always small. The performance of throughput is also improved at load fraction of upper 1.75. As shown in appendix B, it is the different firing rate of the station that makes this difference. For the buffer size of 120, firing rate of station is 0.04, throughput of station is 0.0786, throughput of ring is 0.437. For the buffer size of 5, firing rate of station is 0.055, throughput of station is 0.0802, throughput of ring is 0.442. These properties are very significant in the application.



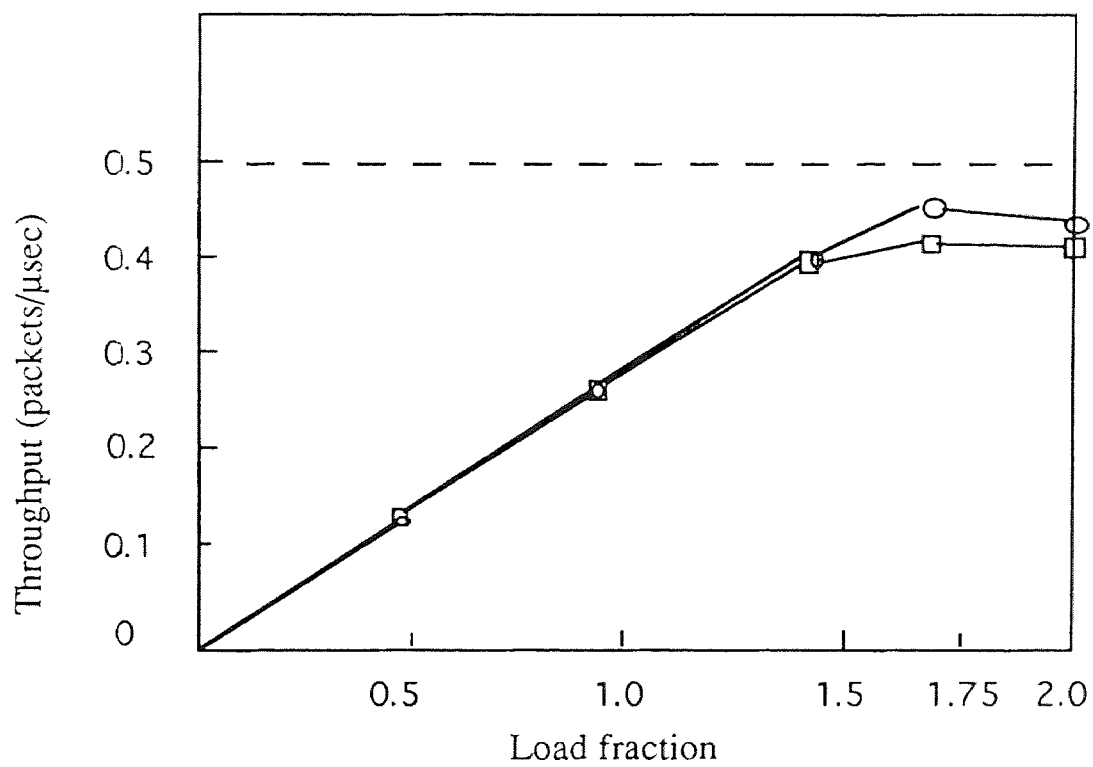
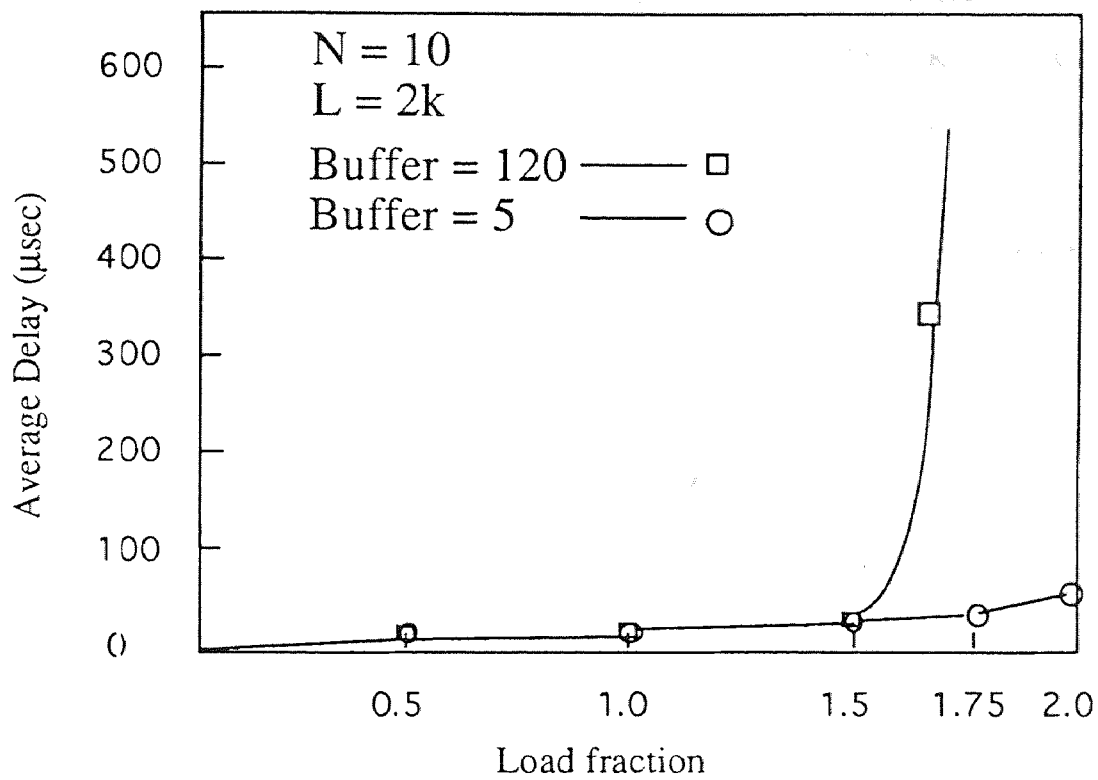


Figure 4-5 Average Delay Performance for Small Buffer Size

#### 4.8 Comparison with Queuing Method and Simulation

Fondriat and Maly [7] presented a paper of CSMA/RN in Computer Communication Review, they used the simulation method to obtain the results. First, they gave an analytical result. Based on average service time, average wait time for a message is calculated using the Pollaczek-Kintchine formula [7]:

$$E\{W\} = \frac{\lambda E\{S^2\}}{2(1 - \rho)} \quad (4.9)$$

Where  $\rho = \lambda E\{S\}$ ,  $E\{S\}$  is the mean service time and  $E\{S^2\}$  is the second moment of  $E\{S\}$ . The analytical result is shown in Figure 4-6. Plotted are the calculated values of wait time only up to 0.95 load fraction since the value goes unstable for load fractions that approach  $\rho = 1.0$ .

Next, they used a real time simulation model to obtain a simulation result. Their system is a real time system where the concept of message truncated was implemented. In their system, a large packet can be divided into several smaller pieces and transmitted.

Comparing the results we obtain in Section 4.3 with their results shown in Figure 4-6, we find when the load fraction is less than 1.5, the two results are almost the same. At the load fraction upper 1.5, the performance of our Petri net model degrades faster than that of their simulation model. The reason is that at the heavy load situation, the empty packets tends to fracture. For our discrete event system model, the efficiency of the ring is lower than that of their real time simulation model.

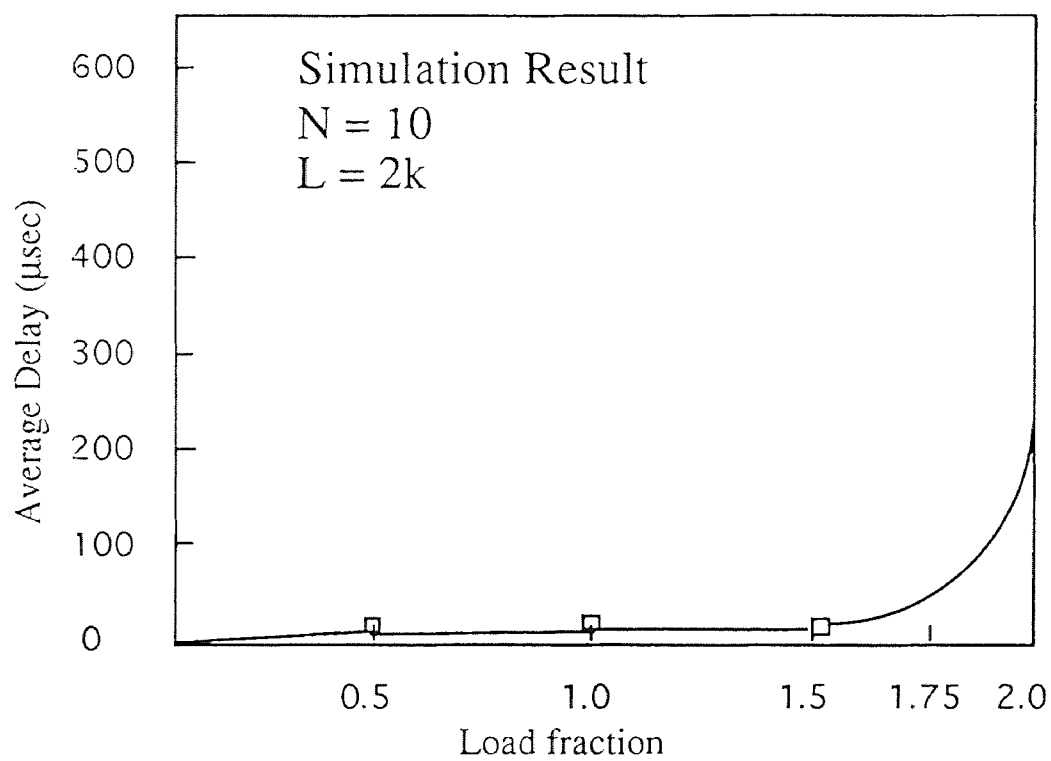
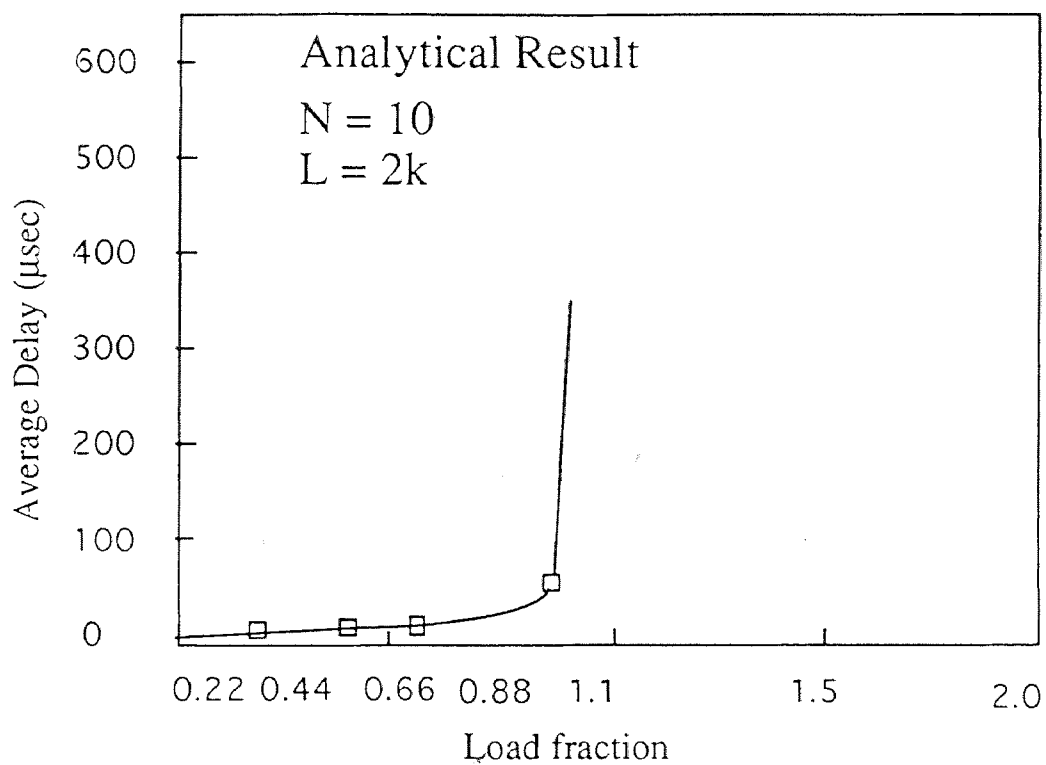


Figure 4-6 A Simulation Result [7]

## CHAPTER 5

### Improved GSPN Model

In Chapter 4, we have discussed the performance of CSMA/RN. Although the analysis results are very good, the network performance v.s. different packet size is not an ideal one. For example, at the load fraction of 1.5, the average delay is 18  $\mu$ sec, 163.7  $\mu$ sec and 252.8  $\mu$ sec for  $L = 2k, 4k, 8k$  respectively. Because the network model is sensitive to the packet size, it is necessary to improve it by renewing the GSPN model. This chapter is contributed to this improvement.

#### 5.1 The Idea of the Improvement and the Improved GSPN Model

In section 4.5, we know when the offered load increases, the empty packets tend to be fracture, the probability for stations to catch a big enough empty packet for their queued customers becomes less and less, especially for big packet size. In order to overcome such a situation for big packet size, we can randomly split one packet into two or three small packets. Our idea is to transmit the smaller packets through the optical fiber.

In our GSPN model, we can use the multiple arc to represent the packet splitting. The model is given in Figure 5-1.

#### 5.2 The Improved Model Description

There are not a lot of changes for the improved model. Only two places and two transitions are added to the GSPN model developed in Chapter 3, they are  $p_{01}$ ,  $p_0$ ,  $t_{01}$  and  $t_0$ . The model description only concentrates on the differences from the old one.



$p_{01}$ : Represents one split packet.

$p_0$ : When the number of tokens in this place reach  $k$  (an integer), this means the service for one customer is finished.

$t_{01}$ : Packet divider.

$t_0$ : Immediate transition. When it fires, the service for one customer is done.

Now, let's see the model operations. Initially, there is one token in both places  $p_4$  and  $p_6$  respectively. Assume the queue of the station is not empty,  $M(p_2) \geq 1$ . The immediate transition  $t_{01}$  fires when  $p_{01} = 0$  and is disabled when  $p_{01} > 0$ . Thus the number of tokens in  $p_{01}$  can never exceed  $k$ . The weight of output arc from  $t_{01}$  to  $p_{01}$  is  $k$  which can be any integer. Once  $t_{01}$  fires, a packet is split into  $k$  small packets. There is an inhibitor arc between  $p_{01}$  and  $t_{01}$ . This inhibitor arc guarantees that before the  $k$  small packets be sent out, all customers in the queue have to wait. When  $M(p_{01}) = 0$ ,  $t_{01}$  fires and sends  $k$  tokens into  $p_{01}$ . This means one customer in the queue is served. The tokens in  $p_{01}$  can be sent out either by  $t_2$  or  $t_3$  depend on the state of the ring. When the ring is idle, tokens from  $p_{01}$  go through  $t_2$ . Otherwise they go through  $t_3$ . When the  $k$  tokens in  $p_{01}$  are all transmitted, the number of tokens in  $p_0$  becomes  $k$ . The immediate transition  $t_0$  fires, drains  $k$  tokens out of  $p_0$  and sends one token back to  $p_1$ . At this moment, the service for one customer is finished. The operations of other part of this model are the same as what we discussed in Section 3.5.

Another interesting question is about the value of  $k$ . In our analysis, we define  $k = \text{average packet length} / \text{average minimum packet length}$ . In order to compare results for deferent packet size of  $2k$ ,  $4k$  and  $8k$ , the average minimum packet length is  $2k$ .

### 5.3 Performance Analysis Result

The processes to obtain the result is the same as what we did in section 4.5 except the different arrival rate associated with  $t_1$ . Because the working environments of the ring are the same for different packet size, so the guessed firing rates of  $t_1$  and  $t_8$  are always the same. Table 5-1 gives the parameters used for this SPNP analysis. Figure 5-2 and Figure 5-3 show the results.

Table 5-1 Improved Model Parameters

Transition	Parameters Value
$t_0$	Immediate with firing probability 1.0
$t_{01}$	Immediate with firing probability 1.0
$t_1$	Timed with firing rate $\lambda$
$t_2$	Timed with rate to be determined
$t_3$	Timed with firing rate $\mu$
$t_4$	Timed with rate 10
$t_5$	Timed with rate 10
$t_6$	Immediate with firing probability $1/N$
$t_7$	Immediate with firing probability $(N-1)/N$
$t_8$	Timed with ring arrival rate to be determined

The new results of the improved model show a very significant improvement on the performance of average delay. For  $L = 2k, 4k$  and  $8k$ ,  $k = 1, 2, 3$  respectively. The results shows that for  $L = 4k$  and  $L = 8k$ , the delay performance is very close to that of  $L = 2k$ . For example, at load fraction of 1.5, the average delays are  $11.7 \mu\text{sec}$ ,  $18.3 \mu\text{sec}$ ,  $31.25 \mu\text{sec}$  respectively. The throughput performance for all of them are the same. Base on this result.

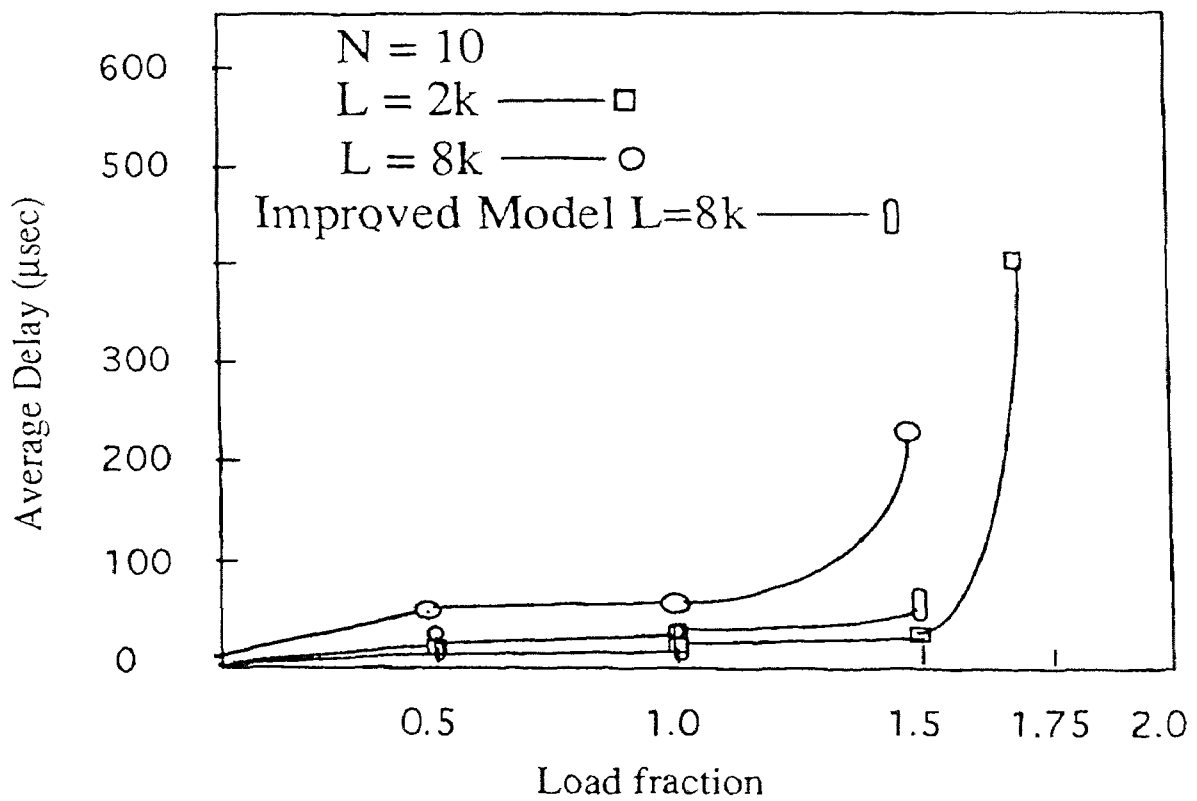
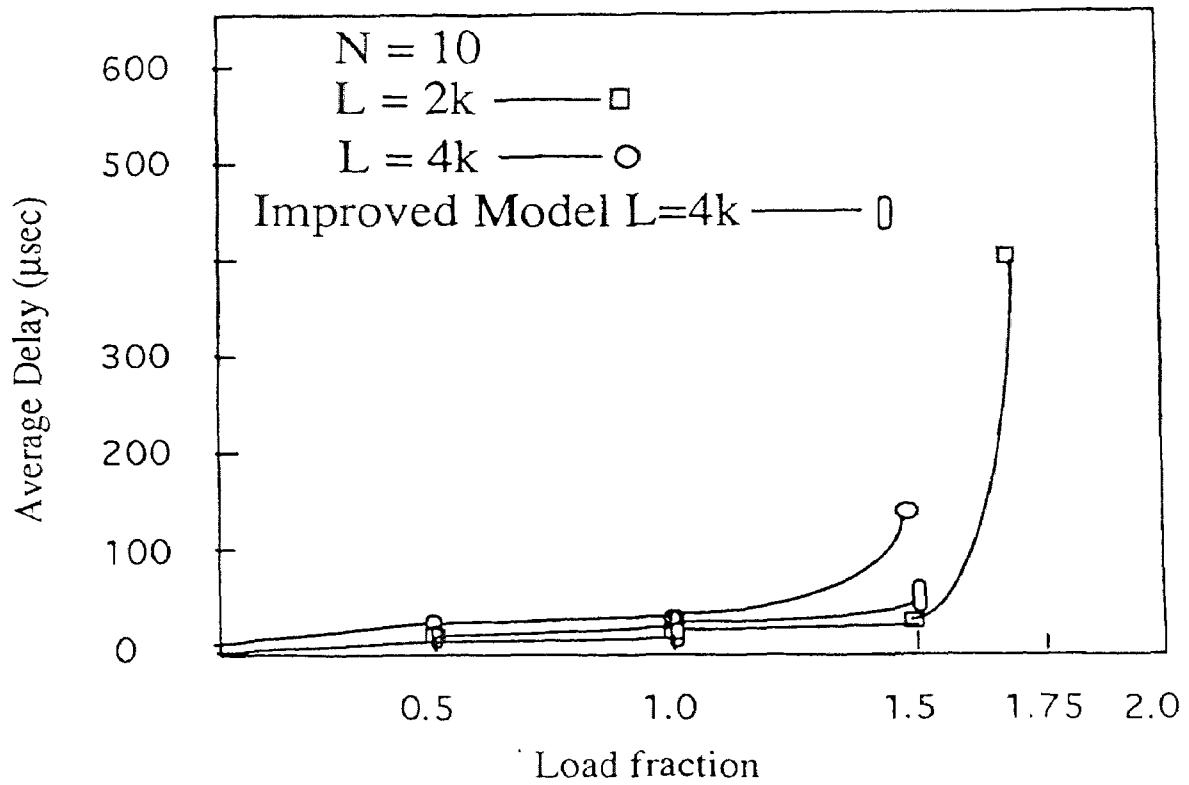


Figure 5-2 Average Delay Performance For Improved Model



For the three different packet size,  
the throughputs do not change

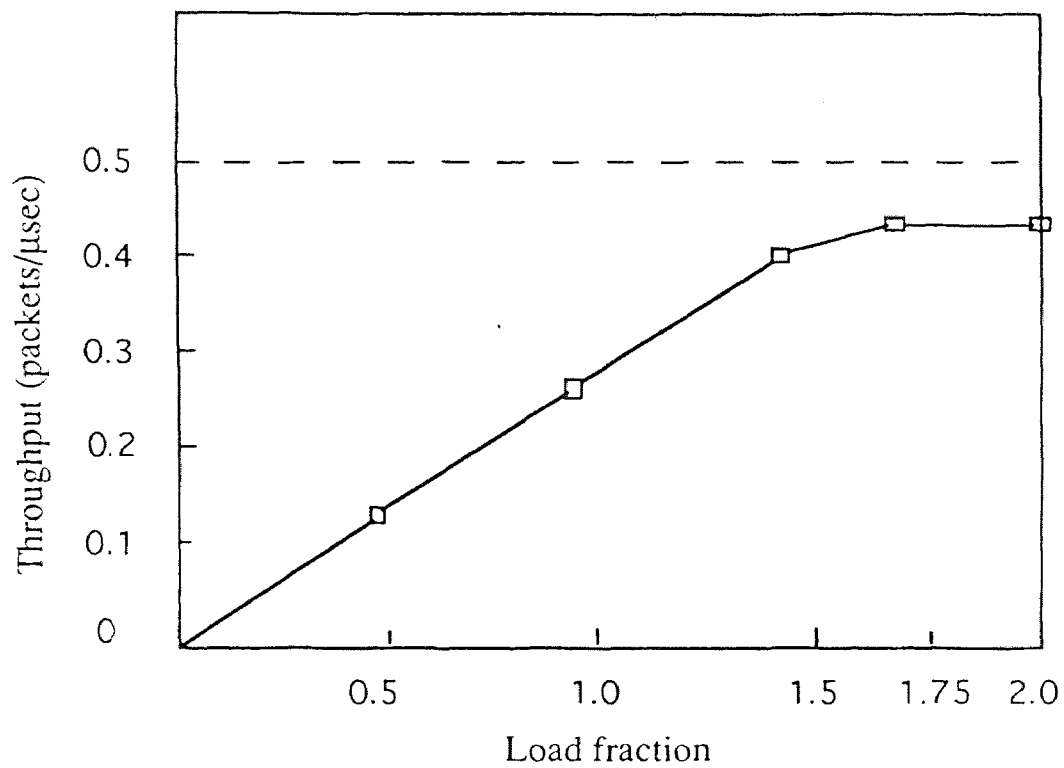


Figure 5-3 Throughput Performance for Improved Model

we can say the packet-size sensitive problem can be overcome.

Here we need declare that the extra header and tailer need to be considered when  $k$  is a big integer. The analysis results of this section are obtained by neglecting the extra header and tailer because  $k$  (1, 2 or 3) is small integer.

In this chapter, we create an improved GSPN model and obtain our results by SPNP software package. Comparing with the simulation results [7], our results are similar to theirs at the load fraction below 1.5. At the load fraction higher than 1.5, our results are not as good as theirs because of our discrete event system model.

## CHAPTER 6

### Conclusions

This thesis has developed generalized stochastic Petri nets (GSPN) models of carrier sensing multiple access ring network (CSMA/RN) and its improved model. Based on these models, the performance of CSMA/RN has been investigated. For performance studies, the GSPN models developed in this thesis are not only appropriate for small number of stations, but also appropriate for any number of stations. By running SPNP program, the numerical results of network performance can be obtained in an interactive way. Small modifications of the GSPN model can represent big changes of network operations. Compared with simulation methods, Petri net methods take much less time. This thesis concludes that generalized stochastic Petri nets are a very useful analytical tool. Accordingly, the contributions of this thesis are as follows:

1. To provide basic concept of carrier sensing multiple access (CSMA) and detail about its extended version — carrier sensing multiple access ring network (CSMA/RN).
2. To provide an introduction to generalized stochastic Petri nets.
3. To model carrier sensing multiple access ring network (CSMA/RN) using GSPN.
4. To present and analyze the performance of CSMA/RN v.s. different load fractions, different number of stations, different packet size and small buffer size for applications.
5. To present an improved generalized stochastic Petri net model and the performance analysis.

6. To make a brief comparison among analysis method, PN, and simulation for the same network performance.

In summary, the method and models developed in this thesis are successfully used to performance analysis of the carrier sensing multiple access ring network system.

However, there is no perfect model particularly with such a complex system as CSMA/RN needs to be modeled. The following paragraphs discuss the limitations of the models developed in this thesis.

Although generalized stochastic Petri net (GSPN) have many advantages in system modeling and performance evaluation, they still have some limitations for CSMA/RN modeling. There are two typical cases:

1. When the average packet length increases significantly, the performance results for carrier sensing multiple access ring network with them are worse than that of the real-time CSMA/RN's performance.
2. The performance analysis v.s. different ring size cannot obtain with models obtained in this thesis.

For the first case, an improved model is given in Chapter 5, but when  $k$  increases to a very large integer, the penalty of transmitting a large number of extra header and tailer becomes very heavy. Especially at the low load situation, the extra header and tailer could create a certain amount of unnecessary transmitting time. Compared with the continuous-time Petri net, our generalized stochastic Petri net model developed in Chapter 3 is not able to deal with interruptions of a message. The improved generalized stochastic Petri net model developed in Chapter 5 can deal with interruptions of a message, but only from the point view of stochastic and may create unnecessary transmission delay at both low load and high load situations.

As we use generalized stochastic Petri nets to model the CSMA/RN, a propagation delay which is nearly deterministic cannot be included in a GSPN model. The performance analysis v.s. different ring size cannot be obtained by the models developed in this thesis.

Based on the above discussions, future research shall be extended along the following directions:

1. Using different values of  $k$  for different load fractions v.s. different network parameters, i. e., packet size, number of stations, to obtain a group of results for each load fraction. Select the best result from the group of results. For example, at the low load situation, the ring is always available and  $k$  should be a small integer or just 1. At the heavy load situation, the empty packet tends to fracture and  $k$  should be a large integer. By doing so, the packet is split into  $k$  number of small packets in order to catch the fracture empty packet. This method shall enable one to obtain more accurate average times of interruptions of a transmitted packet at any load fraction, without calculating the average times of interruptions in a packet transmission at a designated load fraction. It is conjectured that in this way, the performance results should be very close to that of the real time models.
2. Using deterministic and stochastic petri nets (DSPN's) model to obtain the performance analysis v.s. different ring size. The DSPN's driven simulation may need to be conducted.

## APPENDIX A

### SPNP SOURCE CODE

```
/* Model1.c */
# include      "user.h"
# include      <stdio.h>
int n,x;
double lamda, a, b, c, d, pl, a1;
parameters () {
    iopt (IOP_PR_FULL_MARK, VAL_YES);
    iopt (IOP_PR_MC_, VAL_YES);
    iopt (IOP_PR_RGRAPH, VAL_YES);
    iopt (IOP_PR_RSET, VAL_YES);
    iopt (IOP_PR_PROB, VAL_YES);
    n = input ("Number of Stations:");
    c = input ("Channel Capacity (Gbps):");
    pl= input ("Ave. Length of Packet(k):");
    a = input ("Load Fraction of Ring:");
    lamda=input ("Ring Arrival Rate:");
    a1=input ("Guessed Service Rate:");
    d=c/pl;
    b=a*d/n;
}
net () {
    place ("p1"); init ("p1",120);
    place ("p2"); init ("p2", 0);
```

```

place ("p3"); init ("p3", 0);
place ("p4"); init ("p4", 1);
place ("p5"); init ("p5", 0);
place ("p6"); init ("p6", 1);
place ("p7"); init ("p7", 0);

trans ("t1");      rateval ("t1", b);
trans ("t2");      rateval ("t2", a1);
trans ("t3");      rateval ("t3", d);
trans ("t4");      rateval ("t4", 10);
trans ("t5");      rateval ("t5", 10);
trans ("t6");      priority ("t6",1);      probval ("t6", 1.0/n);
trans ("t7");      priority ("t7", 1);      probval ("t7", (n-1.)/n);
trans ("t8");      rateval ("t8", lamda);

iarc ("t1", "p1");      oarc ("t1", "p2");      oarc ("t2", "p1");
iarc ("t2", "p2");      oarc ("t2", "p6");      iarc ("t2", "p6");
iarc ("t2", "p4");      oarc ("t2", "p4");      oarc ("t3", "p6");
iarc ("t3", "p3");      iarc ("t3", "p2");      oarc ("t3", "p1");
iarc ("t4", "p3");      oarc ("t4", "p6");      oarc ("t5", "p4");
iarc ("t5", "p5");      oarc ("t5", "p6");      iarc ("t6", "p7");
oarc ("t6", "p3");      iarc ("t7", "p7");      oarc ("t7", "p5");
iarc ("t7", "p4");      oarc ("t8", "p7");      iarc ("t8", "p6");
harc ("t4", "p2");

}

/* The following lines should appear in all program */
assert () {return (RES_NOERR);}
ac_init () { }

```

```

ac_reach () {fprintf (stderr, "/n The reachibility graph has been generated/n/
n"); }
/* User defined output functions */
reward_type ef0() {return (rate ("t2") + rate ("t3")); }
/* throughput */
reward_type ef2() {return (mark ("p2")); }
/* robot utilization */
/* output */
ac_final () {pr_expected ("throughput = ", ef0);
              pr_expected ("Average queue size = ", ef2);
              pr_std_average (); }

/* Model2.c for improved model */
# include      "user.h"
# include      <stdio.h>
int n,x;
double lamda, a, b, c, d, pl, a1;
parameters () {
              iopt (IOP_PR_FULL_MARK, VAL_YES);
              iopt (IOP_PR_MC_, VAL_YES);
              iopt (IOP_PR_RGRAPH, VAL_YES);
              iopt (IOP_PR_RSET, VAL_YES);
              iopt (IOP_PR_PROB, VAL_YES);
              n = input ("Number of Stations:");
              c = input ("Channel Capacity (Gbps):");
              pl= input ("Ave. Length of Packet(k):");

```



```

    a = input ("Load Fraction of Ring:");
    lamda=input ("Ring Arrival Rate:");
    a1=input ("Guessed Service Rate:");
d=c/pl;
b=a*d/n;
x=pl/2;
}
net () {
    place ("p0");
    place ("p01");
    place ("p1"); init ("p1",120);
    place ("p2"); init ("p2", 0);
    place ("p3"); init ("p3", 0);
    place ("p4"); init ("p4", 1);
    place ("p5"); init ("p5", 0);
    place ("p6"); init ("p6", 1);
    place ("p7"); init ("p7", 0);
    trans ("t0");          priority ("t0", 1);          probval ("t0", 1.);
    trans ("t01");         priority ("t01",1);          probval ("t01", 1.);
    trans ("t1");          rateval ("t1", b);
    trans ("t2");          rateval ("t2", a1);
    trans ("t3");          rateval ("t3", 0.5);
    trans ("t4");          rateval ("t4", 10.);
    trans ("t5");          rateval ("t5", 10.);
    trans ("t6");          priority ("t6",1);          probval ("t6", 1.0/n);
    trans ("t7");          priority ("t7", 1);          probval ("t7", (n-1.)/n);
    trans ("t8");          rateval ("t8", lamda);

```

```

        miarc ("t0", "p0", x);    oarc ("t0", "p1");
        iarc ("t01", "p2");      moarc ("t01", "p01", x);
        iarc ("t1", "p1");        oarc ("t1", "p2");          oarc ("t2", "p1");
        iarc ("t2", "p2");        oarc ("t2", "p6");          iarc ("t2", "p6");
        iarc ("t2", "p4");        oarc ("t2", "p4");          oarc ("t3", "p6");
        iarc ("t3", "p3");        iarc ("t3", "p2");          oarc ("t3", "p1");
        iarc ("t4", "p3");        oarc ("t4", "p6");          oarc ("t5", "p4");
        iarc ("t5", "p5");        oarc ("t5", "p6");          iarc ("t6", "p7");
        oarc ("t6", "p3");        iarc ("t7", "p7");          oarc ("t7", "p5");
        iarc ("t7", "p4");        oarc ("t8", "p7");          iarc ("t8", "p6");
        harc ("t4", "p2");
        harc ("t01", "p2");
    }

    /* The following lines should appear in all program */
    assert () {return (RES_NOERR);}
    ac_init () { }
    ac_reach () {fprintf (stderr, "/n The reachability graph has been generated/n/n"); }
    /* User defined output functions */
    reward_type ef0() {return (rate ("t2") + rate ("t3")); }
    /* throughput */
    reward_type ef2() {return (mark ("p2")); }
    /* robo: utilization */
    /* output */
    ac_final () {pr_expected ("throughput = ", ef0);
                 pr_expected ("Average queue size = ", ef2);
                 pr_std_average (); }

```

## APPENDIX B

### SPNP OUTPUT DATA

#### 1. Buffer Size is 120

INPUT: Number of stations: = 10  
 INPUT: Channel capacity(Gbps): = 1  
 INPUT: Ave. length of packet(k): = 2  
 INPUT: Load fraction of ring: = 0.5  
 INPUT: Ring arrival rate: = 0.14  
 INPUT: Guessed service rate: = 0.1  
 EXPECTED: throughput = = 0.0250001212623  
 EXPECTED: Average queue size = = 0.294045653666  
 AVERAGE:

```

=====
PLACE                Pr[nonempty]                Av[tokens]
0: p1                1.000000000000e+00        1.197059543463e+02
1: p2                2.280373248035e-01        2.940456536659e-01
2: p3                7.250215781621e-03        7.250215781621e-03
3: p4                9.876469991047e-01        9.876469991047e-01
4: p5                1.235300089528e-02        1.235300089528e-02
5: p6                9.803967833231e-01        9.803967833231e-01
6: p7                0.000000000000e+00        0.000000000000e+00
  TRANSITION        Pr[enabled]                Av[throughput]
0: t1                1.000000000000e+00        2.500000000000e-02
1: t2                2.190661593037e-01        2.190661593037e-02
2: t3                6.187010663906e-03        3.093505331953e-03
3: t4                1.063205117715e-03        1.063205117715e-02
4: t5                1.235300089528e-02        1.235300089528e-01
7: t8                9.803967833231e-01        1.372555496652e-01
=====
  
```

INPUT: Number of stations: = 10  
 INPUT: Channel capacity(Gbps): = 1  
 INPUT: Ave. length of packet(k): = 2  
 INPUT: Load fraction of ring: = 1  
 INPUT: Ring arrival rate: = 0.29  
 INPUT: Guessed service rate: = 0.1  
 EXPECTED: throughput = = 0.0500006003851  
 EXPECTED: Average queue size = = 0.711592143924  
 AVERAGE:

```

=====
PLACE                Pr[nonempt]                Av[tokens]
  
```

0: p1	1.000000000000e+00	1.192884078561e+02
1: p2	4197510119437e-01	7.115921439239e-01
2: p3	2.423443123933e-02	2.423443123933e-02
3: p4	9.751803109280e-01	9.751803109280e-01
4: p5	2.481968907199e-02	2.481968907199e-02
5: p6	9.509458796887e-01	9.509458796887e-01
6: p7	0.000000000000e+00	0.000000000000e+00
TRANSITION	Pr[enabled]	Av[throughput]
0: t1	1.000000000000e+00	5.000000000000e-02
1: t2	3.869708036769e-01	3.869708036769e-02
2: t3	2.260704003488e-02	1.130352001744e-02
3: t4	1.627391204447e-03	1.627391204447e-02
4: t5	2.481968907199e-02	2.481968907199e-01
7: t8	9.509458796887e-01	2.757743051097e-01

=====

INPUT: Number of stations: = 10  
 INPUT: Channel capacity(Gbps): = 1  
 INPUT: Ave. length of packet(k): = 2  
 INPUT: Load fraction of ring: = 1.5  
 INPUT: Ring arrival rate: = 0.46  
 INPUT: Guessed service rate: = 0.1  
 EXPECTED: throughput = = 0.0750012038854  
 EXPECTED: Average queue size = = 1.34351542473  
 AVERAGE:

PLACE	Pr[nonempty]	Av[tokens]
0: p1	1.000000000000e+00	1.186564845753e+02
1: p2	5.815690266775e-01	1.343515424729e+00
2: p3	4.927577253289e-02	4.927577253289e-02
3: p4	9.622047391671e-01	9.622047391671e-01
4: p5	3.779526083292e-02	3.779526083292e-02
5: p6	9.129289666342e-01	9.129289666342e-01
6: p7	0.000000000000e+00	0.000000000000e+00
TRANSITION	Pr[enabled]	Av[throughput]
0: t1	1.000000000000e+00	7.500000000000e-02
1: t2	5.127683590675e-01	5.127683590675e-02
2: t3	4.744873595730e-02	2.372436797865e-02
3: t4	1.827036575588e-03	1.827036575588e-02
4: t5	3.779526083292e-02	3.779526083292e-01
7: t8	9.129289666342e-01	4.199473246517e-01

=====

INPUT: Number of stations: = 10  
 INPUT: Channel capacity(Gbps): = 1  
 INPUT: Ave. length of packet(k): = 2  
 INPUT: Load fraction of ring: = 1.75  
 INPUT: Ring arrival rate: = 0.5

INPUT: Guessed service rate: = 0.04  
 EXPECTED: throughput = = 0.078601400612  
 EXPECTED: Average queue size = = 108.971344135  
 AVERAGE:

```
=====
PLACE                Pr[nonempty]          Av[tokens]
0: p1                9.073437937185e-01   1.102865586456e+01
1: p2                9.999844223376e-01   1.089713441354e+02
2: p3                8.733496885700e-02   8.733496885700e-02
3: p4                9.606986350226e-01   9.606986350226e-01
4: p5                3.930136497745e-02   3.930136497745e-02
5: p6                8.733636661656e-01   8.733636661656e-01
6: p7                0.000000000000e+00   0.000000000000e+00
  TRANSITION        Pr[enabled]          Av[throughput]
0: t1                9.073437937185e-01   7.939258195036e-02
1: t2                8.733488241661e-01   3.493395296664e-02
2: t3                8.733489529071e-02   4.366744764535e-02
3: t4                7.356629238712e-08   7.356629238712e-07
4: t5                3.930136497745e-02   3.930136497745e-01
7: t8                8.733636661656e-01   4.366818330828e-01
=====
```

INPUT: Number of stations: = 10  
 INPUT: Channel capacity(Gbps): = 1  
 INPUT: Ave. length of packet(k): = 2  
 INPUT: Load fraction of ring: = 2  
 INPUT: Ring arrival rate: = 0.5  
 INPUT: Guessed service rate: = 0.038  
 EXPECTED: throughput = = 0.0768558940353  
 EXPECTED: Average queue size = = 116.911271488  
 AVERAGE:

```
=====
PLACE                Pr[nonempty]          Av[tokens]
0: p1                7.685852347380e-01   3.088728512158e+00
1: p2                9.999999999932e-01   1.169112714878e+02
2: p3                8.733624199247e-02   8.733624199247e-02
3: p4                9.606986903957e-01   9.606986903957e-01
4: p5                3.930130960431e-02   3.930130960431e-02
5: p6                8.733624484032e-01   8.733624484032e-01
6: p7                0.000000000000e+00   0.000000000000e+00
  TRANSITION        Pr[enabled]          Av[throughput]
0: t1                7.685852347380e-01   7.685852347380e-02
1: t2                8.733624483967e-01   3.318777303908e-02
2: t3                8.733624199244e-02   4.366812099622e-02
3: t4                3.191367189458e-14   3.191367189458e-13
4: t5                3.930130960431e-02   3.930130960431e-01
7: t8                8.733624484032e-01   4.366812242016e-01
=====
```

```

=====
INPUT: Number of stations: = 10
INPUT: Channel capacity(Gbps): = 1
INPUT: Ave. length of packet(k): = 4
INPUT: Load fraction of ring: = 0.5
INPUT: Ring arrival rate: = 0.07
INPUT: Guessed service rate: = 0.04
EXPECTED: throughput = = 0.0125000658116
EXPECTED: Average queue size = = 0.377004680564
AVERAGE:
=====

```

PLACE	Pr[nonempty]	Av[tokens]
0: p1	1.000000000000e+00	1.196229953194e+02
1: p2	2.749736104171e-01	3.770046805642e-01
2: p3	7.971631481761e-03	7.971631481761e-03
3: p4	9.937893477256e-01	9.937893477256e-01
4: p5	6.210652274416e-03	6.210652274416e-03
5: p6	9.858177162438e-01	9.858177162438e-01
6: p7	0.000000000000e+00	

0.000000000000e+00

TRANSITION	Pr[enabled]	Av[throughput]
0: t1	1.000000000000e+00	1.250000000000e-02
1: t2	2.658249849733e-01	1.063299939893e-02
2: t3	7.468265650527e-03	1.867066412632e-03
3: t4	5.033658312343e-04	5.033658312343e-03
4: t5	6.210652274416e-03	6.210652274416e-02
7: t8	9.858177162438e-01	6.900724013707e-02

```

=====
INPUT: Number of stations: = 10
INPUT: Channel capacity(Gbps): = 1
INPUT: Ave. length of packet(k): = 4
INPUT: Load fraction of ring: = 1
INPUT: Ring arrival rate: = 0.145
INPUT: Guessed service rate: = 0.04
EXPECTED: throughput = = 0.0250003591099
EXPECTED: Average queue size = = 0.944908802505
AVERAGE:
=====

```

PLACE	Pr[nonempty]	Av[tokens]
0: p1	1.000000000000e+00	1.190550911975e+02
1: p2	4.910467951463e-01	9.449088025049e-01
2: p3	2.738478721112e-02	2.738478721112e-02
3: p4	9.874708758480e-01	9.874708758480e-01
4: p5	1.252912415199e-02	1.252912415199e-02
5: p6	9.600860886369e-01	9.600860886369e-01
6: p7	0.000000000000e+00	0.000000000000e+00

TRANSITION	Pr[enabled]	Av[throughput]
0: t1	1.000000000000e+00	2.500000000000e-02
1: t2	4.583893475007e-01	1.833557390003e-02
2: t3	2.665914083937e-02	6.664785209844e-03
3: t4	7.256463717459e-04	7.256463717459e-03
4: t5	1.252912415199e-02	1.252912415199e-01
7: t8	9.600860886369e-01	1.392124828523e-01

```

=====
INPUT: Number of stations: = 10
INPUT: Channel capacity(Gbps): = 1
INPUT: Ave. length of packet(k): = 4
INPUT: Load fraction of ring: = 1.5
INPUT: Ring arrival rate: = 0.22
INPUT: Guessed service rate: = 0.027
EXPECTED: throughput = = 0.0377473135861
EXPECTED: Average queue size = = 6.14232760463
AVERAGE:

```

PLACE	Pr[nonempty]	Av[tokens]
0: p1	9.999995610661e-01	1.138576723954e+02
1: p2	8.531883320018e-01	6.142327604628e+00
2: p3	6.813199695465e-02	6.813199695465e-02
3: p4	9.819072499899e-01	9.819072499899e-01
4: p5	1.809275001010e-02	1.809275001010e-02
5: p6	9.137752530353e-01	9.137752530353e-01
6: p7	0.000000000000e+00	0.000000000000e+00

TRANSITION	Pr[enabled]	Av[throughput]
0: t1	9.999995610661e-01	3.749998353998e-02
1: t2	7.701123607581e-01	2.079303374047e-02
2: t3	6.781711938253e-02	1.695427984563e-02
3: t4	3.148775721142e-04	3.148775721142e-03
4: t5	1.809275001010e-02	1.809275001010e-01
7: t8	9.137752530353e-01	2.010305556678e-01

```

=====
INPUT: Number of stations: = 10
INPUT: Channel capacity(Gbps): = 1
INPUT: Ave. length of packet(k): = 8
INPUT: Load fraction of ring: = 0.5
INPUT: Ring arrival rate: = 0.035
INPUT: Guessed service rate: = 0.02
EXPECTED: throughput = = 0.00625003036516
EXPECTED: Average queue size = = 0.375496419128
AVERAGE:

```

PLACE	Pr[nonempty]	Av[tokens]
0: p1	1.000000000000e+00	1.196245035809e+02

1: p2	2.741880546513e-01	3.754964191275e-01
2: p3	7.710599143314e-03	7.710599143314e-03
3: p4	9.968841031545e-01	9.968841031545e-01
4: p5	3.115896845547e-03	3.115896845547e-03
5: p6	9.891735040111e-01	9.891735040111e-01
6: p7	0.000000000000e+00	0.000000000000e+00
TRANSITION	Pr[enabled]	Av[throughput]
0: t1	1.000000000000e+00	6.250000000000e-03
1: t2	2.658914651959e-01	5.317829303918e-03
2: t3	7.457608489972e-03	9.322010612465e-04
3: t4	2.529906533424e-04	2.529906533424e-03
4: t5	3.115896845547e-03	3.115896845547e-02
7: t8	9.891735040111e-01	3.462107264039e-02

```

=====
INPUT: Number of stations: = 10
INPUT: Channel capacity(Gbps): = 1
INPUT: Ave. length of packet(k): = 8
INPUT: Load fraction of ring: = 1
INPUT: Ring arrival rate: = 0.071
INPUT: Gussed service rate: = 0.02
EXPECTED: throughput = = 0.0125001650361
EXPECTED: Average queue size = = 0.942058123798
AVERAGE:
=====

```

PLACE	Pr[nonempty]	Av[tokens]
0: p1	1.000000000000e+00	1.190579418762e+02
1: p2	4.902717061890e-01	9.420581237978e-01
2: p3	2.658500190351e-02	2.658500190351e-02
3: p4	9.938193720361e-01	9.938193720361e-01
4: p5	6.180627963872e-03	6.180627963872e-03
5: p6	9.672343701326e-01	9.672343701326e-01
6: p7	0.000000000000e+00	0.000000000000e+00
TRANSITION	Pr[enabled]	Av[throughput]
0: t1	1.000000000000e+00	1.250000000000e-02
1: t2	4.610951792124e-01	9.221903584249e-03
2: t3	2.622609161443e-02	3.278261451804e-03
3: t4	3.589102890825e-04	3.589102890825e-03
4: t5	6.180627963872e-03	6.180627963872e-02
7: t8	9.672343701326e-01	6.867364027942e-02

```

=====
INPUT: Number of stations: = 10
INPUT: Channel capacity(Gbps): = 1
INPUT: Ave. length of packet(k): = 8
INPUT: Load fraction of ring: = 1.5
INPUT: Ring arrival rate: = 0.11
INPUT: Gussed service rate: = 0.01
=====

```



EXPECTED: throughput = = 0.0188360700342  
 EXPECTED: Average queue size = = 47.3904343277  
 AVERAGE:

```
=====
PLACE                Pr[nonempty]          Av[tokens]
0: p1                9.965570407252e-01   7.260956567228e+01
1: p2                9.847542129993e-01   4.739043432772e+01
2: p3                7.894984284876e-02   7.894984284876e-02
3: p4                9.909709906369e-01   9.909709906369e-01
4: p5                9.029009363103e-03   9.029009363103e-03
5: p6                9.120211477881e-01   9.120211477881e-01
6: p7                0.000000000000e+00   0.000000000000e+00
  TRANSITION        Pr[enabled]          Av[throughput]
0: t1                9.965570407252e-01   1.868544451360e-02
1: t2                8.969409327060e-01   8.969409327060e-03
2: t3                7.893328565690e-02   9.866660707113e-03
3: t4                1.655719185567e-05   1.655719185567e-04
4: t5                9.029009363103e-03   9.029009363103e-02
7: t8                9.120211477881e-01   1.003223262567e-01
=====
```

INPUT: Number of stations: = 20  
 INPUT: Channel capacity(Gbps): = 1  
 INPUT: Ave. length of packet(k): = 2  
 INPUT: Load fraction of ring: = 0.5  
 INPUT: Ring arrival rate: = 0.07  
 INPUT: Guessed service rate: = 0.08  
 EXPECTED: throughput = = 0.0125000204461  
 EXPECTED: Average queue size = = 0.178698067037  
 AVERAGE:

```
=====
PLACE                Pr[nonempty]          Av[tokens]
0: p1                1.000000000000e+00   1.198213019330e+02
1: p2                1.517139323733e-01   1.786980670375e-01
2: p3                1.349429666290e-03   1.349429666290e-03
3: p4                9.934028442178e-01   9.934028442178e-01
4: p5                6.597155782236e-03   6.597155782236e-03
5: p6                9.920534145515e-01   9.920534145515e-01
6: p7                0.000000000000e+00   0.000000000000e+00
  TRANSITION        Pr[enabled]          Av[throughput]
0: t1                1.000000000000e+00   1.250000000000e-02
1: t2                1.496567625428e-01   1.197254100342e-02
2: t3                1.054958885392e-03   5.274794426962e-04
3: t4                2.944707808979e-04   2.944707808979e-03
4: t5                6.597155782236e-03   6.597155782236e-02
7: t8                9.920534145515e-01   6.944373901860e-02
=====
```

INPUT: Number of stations: = 20  
 INPUT: Channel capacity(Gbps): = 1  
 INPUT: Ave. length of packet(k): = 2  
 INPUT: Load fraction of ring: = 1  
 INPUT: Ring arrival rate: = 0.14  
 INPUT: Guessed service rate: = 0.08  
 EXPECTED: throughput = = 0.0250001516266  
 EXPECTED: Average queue size = = 0.41732413737  
 AVERAGE:

```

=====
PLACE                Pr[nonempty]          Av[tokens]
0: p1                1.000000000000e+00   1.195826758626e+02
1: p2                2.951069771185e-01   4.173241373696e-01
2: p3                4.530745715284e-03   4.530745715284e-03
3: p4                9.869340347557e-01   9.869340347557e-01
4: p5                1.306596524433e-02   1.306596524433e-02
5: p6                9.824032890404e-01   9.824032890404e-01
6: p7                0.000000000000e+00   0.000000000000e+00
  TRANSITION        Pr[enabled]          Av[throughput]
0: t1                1.000000000000e+00   2.500000000000e-02
1: t2                2.872185839087e-01   2.297748671269e-02
2: t3                4.045329827796e-03   2.022664913898e-03
3: t4                4.854158874880e-04   4.854158874880e-03
4: t5                1.306596524433e-02   1.306596524433e-01
7: t8                9.824032890404e-01   1.375364604657e-01
=====
  
```

INPUT: Number of stations: = 20  
 INPUT: Channel capacity(Gbps): = 1  
 INPUT: Ave. length of packet(k): = 2  
 INPUT: Load fraction of ring: = 1.5  
 INPUT: Ring arrival rate: = 0.42  
 INPUT: Guessed service rate: = 0.06  
 EXPECTED: throughput = = 0.0375009330796  
 EXPECTED: Average queue size = = 0.986487714551  
 AVERAGE:

```

=====
PLACE                Pr[nonempty]          Av[tokens]
0: p1                1.000000000000e+00   1.190135122854e+02
1: p2                5.005364291734e-01   9.864877145508e-01
2: p3                2.050363298443e-02   2.050363298443e-02
3: p4                9.624176300168e-01   9.624176300168e-01
4: p5                3.758236998325e-02   3.758236998325e-02
5: p6                9.419139970323e-01   9.419139970323e-01
6: p7                0.000000000000e+00   0.000000000000e+00
  TRANSITION        Pr[enabled]          Av[throughput]
0: t1                1.000000000000e+00   3.750000000000e-02
=====
  
```

1: t2	4.625101694919e-01	2.775061016952e-02
2: t3	1.950064582013e-02	9.750322910063e-03
3: t4	1.002987164309e-03	1.002987164309e-02
4: t5	3.758236998325e-02	3.758236998325e-01
7: t8	9.419139970323e-01	3.956038787536e-01

```

=====
INPUT: Number of stations: = 20
INPUT: Channel capacity(Gbps): = 1
INPUT: Ave. length of packet(k): = 2
INPUT: Load fraction of ring: = 1.75
INPUT: Ring arrival rate: = 0.5
INPUT: Guessed service rate: = 0.04
EXPECTED: throughput = = 0.0439969352181
EXPECTED: Average queue size = = 3.00396814743
AVERAGE:

```

PLACE	Pr[nonempty]	Av[tokens]
0: p1	9.999999991475e-01	1.169960318526e+02
1: p2	7.425245222000e-01	3.003968147427e+00
2: p3	3.448716215916e-02	3.448716215916e-02
3: p4	9.562177949428e-01	9.562177949428e-01
4: p5	4.378220505722e-02	4.378220505722e-02
5: p6	9.217306327836e-01	9.217306327836e-01
6: p7	0.000000000000e+00	0.000000000000e+00
TRANSITION	Pr[enabled]	Av[throughput]
0: t1	9.999999991475e-01	4.374999996270e-02
1: t2	6.764650175942e-01	2.705860070377e-02
2: t3	3.387666902863e-02	1.693833451431e-02
3: t4	6.104931305276e-04	6.104931305276e-03
4: t5	4.378220505722e-02	4.378220505722e-01
7: t8	9.217306327836e-01	4.608653163918e-01

```

=====
INPUT: Number of stations: = 20
INPUT: Channel capacity(Gbps): = 1
INPUT: Ave. length of packet(k): = 2
INPUT: Load fraction of ring: = 2
INPUT: Ring arrival rate: = 0.5
INPUT: Guessed service rate: = 0.022
EXPECTED: throughput = = 0.042823625714
EXPECTED: Average queue size = = 105.056883164
AVERAGE:

```

PLACE	Pr[nonempty]	Av[tokens]
0: p1	8.970228523718e-01	1.494311683599e+01
1: p2	9.999771648448e-01	1.050568831640e+02
2: p3	4.555710527659e-02	4.555710527659e-02

3: p4	9.567197732703e-01	9.567197732703e-01
4: p5	4.328022672970e-02	4.328022672970e-02
5: p6	9.111626679937e-01	9.111626679937e-01
6: p7	0.000000000000e+00	0.000000000000e+00
TRANSITION	Pr[enabled]	Av[throughput]
0: t1	8.970228523718e-01	4.485114261859e-02
1: t2	9.111409150734e-01	2.004510013161e-02
2: t3	4.555705116484e-02	2.277852558242e-02
3: t4	5.411174208554e-08	5.411174208554e-07
4: t5	4.328022672970e-02	4.328022672970e-01
7: t8	9.111626679937e-01	4.555813339969e-01

```

=====
INPUT: Number of stations: = 20
INPUT: Channel capacity(Gbps): = 1
INPUT: Ave. length of packet(k): = 4
INPUT: Load fraction of ring: = 0.5
INPUT: Ring arrival rate: = 0.065
INPUT: Gussed service rate: = 0.065
EXPECTED: throughput = = 0.0062499853634
EXPECTED: Average queue size = = 0.102829937603
AVERAGE:

```

PLACE	Pr[nonempty]	Av[tokens]
0: p1	1.000000000000e+00	1.198971700624e+02
1: p2	9.331430714184e-02	1.028299376029e-01
2: p3	1.490027925880e-03	1.490027925880e-03
3: p4	9.938720404864e-01	9.938720404864e-01
4: p5	6.127959513606e-03	6.127959513606e-03
5: p6	9.923820125605e-01	9.923820125605e-01
6: p7	0.000000000000e+00	0.000000000000e+00
TRANSITION	Pr[enabled]	Av[throughput]
0: t1	1.000000000000e+00	6.250000000000e-03
1: t2	9.154808353618e-02	5.950625429851e-03
2: t3	1.197439734189e-03	2.993599335472e-04
3: t4	2.925881916911e-04	2.925881916911e-03
4: t5	6.127959513606e-03	6.127959513606e-02
7: t8	9.923820125605e-01	6.450483081643e-02

```

=====
INPUT: Number of stations: = 20
INPUT: Channel capacity(Gbps): = 1
INPUT: Ave. length of packet(k): = 4
INPUT: Load fraction of ring: = 1
INPUT: Ring arrival rate: = 0.135
INPUT: Gussed service rate: = 0.06
EXPECTED: throughput = = 0.012500103113
EXPECTED: Average queue size = = 0.240826335038

```

AVERAGE:

```
=====
PLACE                Pr[nonempty]                Av[tokens]
0: p1                1.000000000000e+00        1.197591736650e+02
1: p2                1.946605897366e-01        2.408263350378e-01
2: p3                5.615998173377e-03        5.615998173377e-03
3: p4                9.874085098090e-01        9.874085098090e-01
4: p5                1.259149019103e-02        1.259149019103e-02
5: p6                9.817925116356e-01        9.817925116356e-01
6: p7                0.000000000000e+00        0.000000000000e+00
  TRANSITION        Pr[enabled]                Av[throughput]
0: t1                1.000000000000e+00        1.250000000000e-02
1: t2                1.871671537370e-01        1.123002922422e-02
2: t3                5.080295555135e-03        1.270073888784e-03
3: t4                5.357026182415e-04        5.357026182415e-03
4: t5                1.259149019103e-02        1.259149019103e-01
7: t8                9.817925116356e-01        1.325419890708e-01
=====
```

INPUT: Number of stations: = 20

INPUT: Channel capacity(Gbps): = 1

INPUT: Ave. length of packet(k): = 4

INPUT: Load fraction of ring: = 1.5

INPUT: Ring arrival rate: = 0.2

INPUT: Guessed service rate: = 0.04

EXPECTED: throughput = = 0.0187503824786

EXPECTED: Average queue size = = 0.650501266809

AVERAGE:

```
=====
PLACE                Pr[nonempty]                Av[tokens]
0: p1                1.000000000000e+00        1.193494987332e+02
1: p2                3.969640496234e-01        6.505012668091e-01
2: p3                1.562583651735e-02        1.562583651735e-02
3: p4                9.816456231171e-01        9.816456231171e-01
4: p5                1.835437688291e-02        1.835437688291e-02
5: p6                9.660197865997e-01        9.660197865997e-01
6: p7                0.000000000000e+00        0.000000000000e+00
  TRANSITION        Pr[enabled]                Av[throughput]
0: t1                1.000000000000e+00        1.875000000000e-02
1: t2                3.747863780544e-01        1.499145512217e-02
2: t3                1.503570942579e-02        3.758927356448e-03
3: t4                5.901270915620e-04        5.901270915620e-03
4: t5                1.835437688291e-02        1.835437688291e-01
7: t8                9.660197865997e-01        1.932039573199e-01
=====
```

INPUT: Number of stations: = 20

INPUT: Channel capacity(Gbps): = 1

INPUT: Ave. length of packet(k): = 4  
 INPUT: Load fraction of ring: = 1.75  
 INPUT: Ring arrival rate: = 0.245  
 INPUT: Guessed service rate: = 0.02  
 EXPECTED: throughput = = 0.0219480982502  
 EXPECTED: Average queue size = = 2.72529125654  
 AVERAGE:

```

=====
PLACE                Pr[nonempty]                Av[tokens]
0: p1                9.999999998062e-01        1.172747087435e+02
1: p2                7.295556989875e-01        2.725291256544e+00
2: p3                3.368769070399e-02        3.368769070399e-02
3: p4                9.780206503639e-01        9.780206503639e-01
4: p5                2.197934963608e-02        2.197934963608e-02
5: p6                9.443329596599e-01        9.443329596599e-01
6: p7                0.000000000000e+00        0.000000000000e+00
TRANSITION          Pr[enabled]                Av[throughput]
0: t1                9.999999998062e-01        2.187499999576e-02
1: t2                6.803423121446e-01        1.360684624289e-02
2: t3                3.336500802913e-02        8.341252007283e-03
3: t4                3.226826748551e-04        3.226826748551e-03
4: t5                2.197934963608e-02        2.197934963608e-01
7: t8                9.443329596599e-01        2.313615751167e-01
=====
  
```

INPUT: Number of stations: = 20  
 INPUT: Channel capacity(Gbps): = 1  
 INPUT: Ave. length of packet(k): = 4  
 INPUT: Load fraction of ring: = 2  
 INPUT: Ring arrival rate: = 0.25  
 INPUT: Guessed service rate: = 0.0115  
 EXPECTED: throughput = = 0.0223498314763  
 EXPECTED: Average queue size = = 102.272075801  
 AVERAGE:

```

=====
PLACE                Pr[nonempty]                Av[tokens]
0: p1                9.226484089847e-01        1.772792419873e+01
1: p2                9.999220557487e-01        1.022720758013e+02
2: p3                4.656224827526e-02        4.656224827526e-02
3: p4                9.778811754789e-01        9.778811754789e-01
4: p5                2.211882452109e-02        2.211882452109e-02
5: p6                9.313189272036e-01        9.313189272036e-01
6: p7                0.000000000000e+00        0.000000000000e+00
TRANSITION          Pr[enabled]                Av[throughput]
0: t1                9.226484089847e-01        2.306621022462e-02
1: t2                9.312428794061e-01        1.070929311317e-02
2: t3                4.656215345257e-02        1.164053836314e-02
=====
  
```

3: t4	9.482269019574e-08	9.482269019574e-07
4: t5	2.211882452109e-02	2.211882452109e-01
7: t8	9.313189272036e-01	2.328297318009e-01

```

=====
INPUT: Number of stations: = 10
INPUT: Channel capacity(Gbps): = 1
INPUT: Ave. length of packet(k): = 4
INPUT: Load fraction of ring: = 1.75
INPUT: Ring arrival rate: = 0.25
INPUT: Guessed service rate: = 0.02
EXPECTED: throughput = = 0.0400822168806
EXPECTED: Average queue size = = 98.8710335162
AVERAGE:
=====

```

PLACE	Pr[nonempty]	Av[tokens]
0: p1	9.402625485441e-01	2.112896648378e+01
1: p2	9.998282664311e-01	9.887103351622e+01
2: p3	8.907204258991e-02	8.907204258991e-02
3: p4	9.799551305215e-01	9.799551305215e-01
4: p5	2.004486947846e-02	2.004486947846e-02
5: p6	8.908830879316e-01	8.908830879316e-01
6: p7	0.000000000000e+00	0.000000000000e+00
TRANSITION	Pr[enabled]	Av[throughput]
0: t1	9.402625485441e-01	4.113648649880e-02
1: t2	8.907155251841e-01	1.781431050368e-02
2: t3	8.907162550778e-02	2.226790637694e-02
3: t4	4.170821345621e-07	4.170821345621e-06
4: t5	2.004486947846e-02	2.004486947846e-01
7: t8	8.908830879316e-01	2.227207719829e-01

## 2. Buffer Size is 5.

```

INPUT: Number of stations: = 10
INPUT: Channel capacity(Gbps): = 1
INPUT: Ave. length of packet(k): = 2
INPUT: Load fraction of ring: = 0.5
INPUT: Ring arrival rate: = 0.14
INPUT: Guessed service rate: = 0.1
EXPECTED: throughput = = 0.0249889964236
EXPECTED: Average queue size = = 0.293260491618
AVERAGE:
=====

```

PLACE	Pr[nonempty]	Av[tokens]
-------	--------------	------------

0: p1	9.995565821510e-01	4.706739508382e+00
1: p2	2.279357522720e-01	2.932604916177e-01
2: p3	7.247629946572e-03	7.247629946572e-03
3: p4	9.876469677787e-01	9.876469677787e-01
4: p5	1.235303222130e-02	1.235303222130e-02
5: p6	9.803993378321e-01	9.803993378321e-01
6: p7	0.000000000000e+00	0.000000000000e+00
TRANSITION	Pr[enabled]	Av[throughput]
0: t1	9.995565821510e-01	2.498891455378e-02
1: t2	2.189685389021e-01	2.189685389021e-02
2: t3	6.184285066774e-03	3.092142533387e-03
3: t4	1.063344879799e-03	1.063344879799e-02
4: t5	1.235303222130e-02	1.235303222130e-01
7: t8	9.803993378321e-01	1.372559072965e-01

```

=====
INPUT: Number of stations: = 10
INPUT: Channel capacity(Gbps): = 1
INPUT: Ave. length of packet(k): = 2
INPUT: Load fraction of ring: = 1
INPUT: Ring arrival rate: = 0.29
INPUT: Gussed service rate: = 0.1
EXPECTED: throughput = = 0.0496569920838
EXPECTED: Average queue size = = 0.682040813299
AVERAGE:

```

PLACE	Pr[nonempty]	Av[tokens]
0: p1	9.931357758625e-01	4.317959186701e+00
1: p2	4.168611356721e-01	6.820408132986e-01
2: p3	2.408867510129e-02	2.408867510129e-02
3: p4	9.751766040349e-01	9.751766040349e-01
4: p5	2.482339596509e-02	2.482339596509e-02
5: p6	9.510879289336e-01	9.510879289336e-01
6: p7	0.000000000000e+00	0.000000000000e+00
TRANSITION	Pr[enabled]	Av[throughput]
0: t1	9.931357758625e-01	4.965678879312e-02
1: t2	3.843040260571e-01	3.843040260571e-02
2: t3	2.245317895615e-02	1.122658947808e-02
3: t4	1.635496145140e-03	1.635496145140e-02
4: t5	2.482339596509e-02	2.482339596509e-01
7: t8	9.510879289336e-01	2.758154993907e-01

```

=====
INPUT: Number of stations: = 10
INPUT: Channel capacity(Gbps): = 1
INPUT: Ave. length of packet(k): = 2
INPUT: Load fraction of ring: = 1.5
INPUT: Ring arrival rate: = 0.44

```



INPUT: Guessed service rate: = 0.1  
 EXPECTED: throughput = = 0.0729794677134  
 EXPECTED: Average queue size = = 1.14711228857  
 AVERAGE:

```
=====
PLACE                Pr[nonempty]          Av[tokens]
0: p1                9.730549900342e-01   3.852887711428e+00
1: p2                5.708056590073e-01   1.147112288572e+00
2: p3                4.658160233949e-02   4.658160233949e-02
3: p4                9.636827919231e-01   9.636827919231e-01
4: p5                3.631720807694e-02   3.631720807694e-02
5: p6                9.171011895836e-01   9.171011895836e-01
6: p7                0.000000000000e+00   0.000000000000e+00
  TRANSITION        Pr[enabled]          Av[throughput]
0: t1                9.730549900342e-01   7.297912425257e-02
1: t2                5.058664822831e-01   5.058664822831e-02
2: t3                4.478563897026e-02   2.239281948513e-02
3: t4                1.795963369228e-03   1.795963369228e-02
4: t5                3.631720807694e-02   3.631720807694e-01
7: t8                9.171011895836e-01   4.035245234168e-01
=====
```

INPUT: Number of stations: = 10  
 INPUT: Channel capacity(Gbps): = 1  
 INPUT: Ave. length of packet(k): = 2  
 INPUT: Load fraction of ring: = 1.75  
 INPUT: Ring arrival rate: = 0.5  
 INPUT: Guessed service rate: = 0.09  
 EXPECTED: throughput = = 0.0825827716266  
 EXPECTED: Average queue size = = 1.56545896408  
 AVERAGE:

```
=====
PLACE                Pr[nonempty]          Av[tokens]
0: p1                9.438001936564e-01   3.434541035918e+00
1: p2                6.746995568465e-01   1.565458964082e+00
2: p3                6.069676635228e-02   6.069676635228e-02
3: p4                9.595515344384e-01   9.595515344384e-01
4: p5                4.044846556158e-02   4.044846556158e-02
5: p6                8.988547680861e-01   8.988547680861e-01
6: p7                0.000000000000e+00   0.000000000000e+00
  TRANSITION        Pr[enabled]          Av[throughput]
0: t1                9.438001936564e-01   8.258251694494e-02
1: t2                5.889168050954e-01   5.300251245859e-02
2: t3                5.916051833600e-02   2.958025916800e-02
3: t4                1.536248016278e-03   1.536248016278e-02
4: t5                4.044846556158e-02   4.044846556158e-01
7: t8                8.988547680861e-01   4.494273840431e-01
=====
```

```

=====
INPUT: Number of stations: = 10
INPUT: Channel capacity(Gbps): = 1
INPUT: Ave. length of packet(k): = 2
INPUT: Load fraction of ring: = 2
INPUT: Ring arrival rate: = 0.5
INPUT: Guessed service rate: = 0.055
EXPECTED: throughput = = 0.0802829688745
EXPECTED: Average queue size = = 2.76181831688
AVERAGE:
=====

```

PLACE	Pr[nonempty]	Av[tokens]
0: p1	8.028312481526e-01	2.238181683116e+00
1: p2	8.749946514370e-01	2.761818316884e+00
2: p3	7.711139196803e-02	7.711139196803e-02
3: p4	9.602583861346e-01	9.602583861346e-01
4: p5	3.974161386542e-02	3.974161386542e-02
5: p6	8.831469941665e-01	8.831469941665e-01
6: p7	0.000000000000e+00	0.000000000000e+00
TRANSITION	Pr[enabled]	Av[throughput]
0: t1	8.028312481526e-01	8.028312481526e-02
1: t2	7.640381220597e-01	4.202209671328e-02
2: t3	7.652174432242e-02	3.826087216121e-02
3: t4	5.896476456084e-04	5.896476456084e-03
4: t5	3.974161386542e-02	3.974161386542e-01
7: t8	8.831469941665e-01	4.415734970833e-01

```

=====
INPUT: Number of stations: = 10
INPUT: Channel capacity(Gbps): = 1
INPUT: Ave. length of packet(k): = 2
INPUT: Load fraction of ring: = 2.25
INPUT: Ring arrival rate: = 0.5
INPUT: Guessed service rate: = 0.0455
EXPECTED: throughput = = 0.0782702215699
EXPECTED: Average queue size = = 3.3577325019
AVERAGE:
=====

```

PLACE	Pr[nonempty]	Av[tokens]
0: p1	6.957374673591e-01	1.642267498095e+00
1: p2	9.381249686294e-01	3.357732501905e+00
2: p3	8.228111786623e-02	8.228111786623e-02
3: p4	9.604810064969e-01	9.604810064969e-01
4: p5	3.951899350309e-02	3.951899350309e-02
5: p6	8.781998886307e-01	8.781998886307e-01
6: p7	0.000000000000e+00	0.000000000000e+00
TRANSITION	Pr[enabled]	Av[throughput]

0: t1	6.957374673591e-01	7.827046507790e-02
1: t2	8.192400502608e-01	3.727542228687e-02
2: t3	8.198959856616e-02	4.099479928308e-02
3: t4	2.915193000725e-04	2.915193000725e-03
4: t5	3.951899350309e-02	3.951899350309e-01
7: t8	8.781998886307e-01	4.390999443153e-01

### 3. Data for Improved Model

INPUT: Number of stations: = 10  
 INPUT: Channel capacity(Gbps): = 1  
 INPUT: Ave. length of packets(k): = 2  
 INPUT: Load fraction of ring: = 0.5  
 INPUT: Ring arrival rate: = 0.14  
 INPUT: Guessed service rate: = 0.1  
 EXPECTED: throughput = = 0.0250001222936  
 EXPECTED: robot utilization = = 0.0723310350726  
 AVERAGE:

PLACE	Pr[nonempty]	Av[tokens]
0: p0	0.000000000000e+00	0.000000000000e+00
1: p01	2.459711455966e-01	2.459711455966e-01
2: p1	1.000000000000e+00	1.196816978193e+02
3: p2	5.609295496155e-02	7.233103507257e-02
4: p3	2.811446056143e-03	2.811446056143e-03
5: p4	9.875917665474e-01	9.875917665474e-01
6: p5	1.240823345262e-02	1.240823345262e-02
7: p6	9.847803204912e-01	9.847803204912e-01
8: p7	0.000000000000e+00	0.000000000000e+00
TRANSITION	Pr[enabled]	Av[throughput]
2: t1	1.000000000000e+00	2.500000000000e-02
3: t2	2.411365192220e-01	2.411365192220e-02
4: t3	1.772940742725e-03	8.864703713624e-04
5: t4	1.290045563689e-03	1.290045563689e-02
6: t5	1.240823345262e-02	1.240823345262e-01
9: t8	9.847803204912e-01	1.378692448688e-01

INPUT: Number of stations: = 10  
 INPUT: Channel capacity(Gbps): = 1  
 INPUT: Ave. length of packets(k): = 2  
 INPUT: Load fraction of ring: = 1  
 INPUT: Ring arrival rate: = 0.29  
 INPUT: Guessed service rate: = 0.1  
 EXPECTED: throughput = = 0.050000605438  
 EXPECTED: robot utilization = = 0.332028839219

## AVERAGE:

```

=====
PLACE                Pr[nonempty]          Av[tokens]
0: p0                0.000000000000e+00  0.000000000000e+00
1: p01               4.665504242260e-01  4.665504242260e-01
2: p1                1.000000000000e+00  1.192014207366e+02
3: p2                1.958467963729e-01  3.320288392186e-01
4: p3                1.277155309565e-02  1.277155309565e-02
5: p4                9.748887399529e-01  9.748887399529e-01
6: p5                2.511126004706e-02  2.511126004706e-02
7: p6                9.621171868573e-01  9.621171868573e-01
8: p7                0.000000000000e+00  0.000000000000e+00
TRANSITION          Pr[enabled]          Av[throughput]
2: t1                1.000000000000e+00  5.000000000000e-02
3: t2                4.436289667203e-01  4.436289667203e-02
4: t3                1.127541753194e-02  5.637708765972e-03
5: t4                2.226369091814e-03  2.226369091814e-02
6: t5                2.511126004706e-02  2.511126004706e-01
9: t8                9.621171868573e-01  2.790139841886e-01
=====

```

```

INPUT: Number of stations: = 10
INPUT: Channel capacity(Gbps): = 1
INPUT: Ave. length of packets(k): = 2
INPUT: Load fraction of ring: = 1.5
INPUT: Ring arrival rate: = 0.45
INPUT: Guessed service rate: = 0.1
EXPECTED: throughput = = 0.0750014658661
EXPECTED: robot utilization = = 0.880901352165
AVERAGE:

```

```

=====
PLACE                Pr[nonempty]          Av[tokens]
0: p0                0.000000000000e+00  0.000000000000e+00
1: p01               6.484810148522e-01  6.484810148522e-01
2: p1                1.000000000000e+00  1.184706176330e+02
3: p2                3.788010196804e-01  8.809013521653e-01
4: p3                3.291534898556e-02  3.291534898556e-02
5: p4                9.623575881270e-01  9.623575881270e-01
6: p5                3.764241187299e-02  3.764241187299e-02
7: p6                9.294422391415e-01  9.294422391415e-01
8: p7                0.000000000000e+00  0.000000000000e+00
TRANSITION          Pr[enabled]          Av[throughput]
2: t1                1.000000000000e+00  7.500000000000e-02
3: t2                5.929525734721e-01  5.929525734721e-02
4: t3                3.141241703771e-02  1.570620851885e-02
5: t4                2.611869269292e-03  2.611869269292e-02
6: t5                3.764241187299e-02  3.764241187299e-01
=====

```



5: p4	9.875883417279e-01	9.875883417279e-01
6: p5	1.241165827209e-02	1.241165827209e-02
7: p6	9.850521929035e-01	9.850521929035e-01
8: p7	0.000000000000e+00	0.000000000000e+00
TRANSITION	Pr[enabled]	Av[throughput]
2: t1	1.000000000000e+00	1.250000000000e-02
3: t2	2.425124830007e-01	2.425124830007e-02
4: t3	1.497890824986e-03	7.489454124929e-04
5: t4	1.304178596708e-03	1.304178596708e-02
6: t5	1.241165827209e-02	1.241165827209e-01
9: t8	9.850521929035e-01	1.379073070065e-01

```

=====
INPUT: Number of stations: = 10
INPUT: Channel capacity(Gbps): = 1
INPUT: Ave. length of packets(k): = 4
INPUT: Load fraction of ring: = 1
INPUT: Ring arrival rate: = 0.29
INPUT: Guessed service rate: = 0.1
EXPECTED: throughput = = 0.0500008502711
EXPECTED: robot utilization = = 0.254957762433
AVERAGE:
=====

```

PLACE	Pr[nonempty]	Av[tokens]
0: p0	2.330754637059e-01	2.330754637059e-01
1: p01	4.719980826721e-01	7.109207016383e-01
2: p1	1.000000000000e+00	1.192730441549e+02
3: p2	1.700822718140e-01	2.549577624331e-01
4: p3	1.143337192202e-02	1.143337192202e-02
5: p4	9.748547026145e-01	9.748547026145e-01
6: p5	2.514529738555e-02	2.514529738555e-02
7: p6	9.634213306924e-01	9.634213306924e-01
8: p7	0.000000000000e+00	0.000000000000e+00
TRANSITION	Pr[enabled]	Av[throughput]
2: t1	1.000000000000e+00	2.500000000000e-02
3: t2	4.502638728363e-01	4.502638728363e-02
4: t3	9.948925974847e-03	4.974462987424e-03
5: t4	2.296475608431e-03	2.296475608431e-02
6: t5	2.514529738555e-02	2.514529738555e-01
9: t8	9.634213306924e-01	2.793921859008e-01

```

=====
INPUT: Number of stations: = 10
INPUT: Channel capacity(Gbps): = 1
INPUT: Ave. length of packets(k): = 4
INPUT: Load fraction of ring: = 1.5
INPUT: Ring arrival rate: = 0.45
INPUT: Guessed service rate: = 0.1
=====

```

EXPECTED: throughput = = 0.0750030156918  
 EXPECTED: robot utilization = = 0.685296650246  
 AVERAGE:

```
=====
PLACE                Pr[nonempty]          Av[tokens]
0: p0                3.234002492524e-01   3.234002492524e-01
1: p01              6.587237834783e-01   9.940473177042e-01
2: p1                1.000000000000e+00   1.186559795663e+02
3: p2                3.467023487710e-01   6.852966502455e-01
4: p3                3.043092682452e-02   3.043092682452e-02
5: p4                9.622608863014e-01   9.622608863014e-01
6: p5                3.773911369859e-02   3.773911369859e-02
7: p6                9.318299594769e-01   9.318299594769e-01
8: p7                0.000000000000e+00   0.000000000000e+00
TRANSITION          Pr[enabled]          Av[throughput]
2: t1                1.000000000000e+00   3.750000000000e-02
3: t2                6.051973026355e-01   6.051973026355e-02
4: t3                2.896657085649e-02   1.448328542824e-02
5: t4                2.744906270806e-03   2.744906270806e-02
6: t5                3.773911369859e-02   3.773911369859e-01
9: t8                9.318299594769e-01   4.193234817646e-01
=====
```

INPUT: Number of stations: = 10  
 INPUT: Channel capacity(Gbps): = 1  
 INPUT: Ave. length of packets(k): = 4  
 INPUT: Load fraction of ring: = 1.75  
 INPUT: Ring arrival rate: = 0.5  
 INPUT: Guessed service rate: = 0.04  
 EXPECTED: throughput = = 0.0785995174164  
 EXPECTED: robot utilization = = 100.428867712  
 AVERAGE:

```
=====
PLACE                Pr[nonempty]          Av[tokens]
0: p0                4.999881482539e-01   4.999881482539e-01
1: p01              9.999848524897e-01   1.499981556726e+00
2: p1                9.333883565937e-01   1.857114743590e+01
3: p2                9.999346403718e-01   1.004288677116e+02
4: p3                8.733085258547e-02   8.733085258547e-02
5: p4                9.606984577668e-01   9.606984577668e-01
6: p5                3.930154223316e-02   3.930154223316e-02
7: p6                8.733676051814e-01   8.733676051814e-01
8: p7                0.000000000000e+00   0.000000000000e+00
TRANSITION          Pr[enabled]          Av[throughput]
2: t1                9.333883565937e-01   4.083574060097e-02
3: t2                8.733531759881e-01   3.493412703953e-02
4: t3                8.733078075377e-02   4.366539037688e-02
=====
```

5: t4	2.989882184430e-07	2.989882184430e-06
6: t5	3.930154223316e-02	3.930154223316e-01
9: t8	8.733676051814e-01	4.366838025907e-01

```

=====
INPUT: Number of stations: = 10
INPUT: Channel capacity(Gbps): = 1
INPUT: Ave. length of packets(k): = 8
INPUT: Load fraction of ring: = 0.5
INPUT: Ring arrival rate: = 0.14
INPUT: Guessed service rate: = 0.1
EXPECTED: throughput = = 0.0250004458722
EXPECTED: robot utilization = = 0.0458189831791
AVERAGE:

```

PLACE	Pr[nonempty]	Av[tokens]
0: p0	1.853194071829e-01	3.700627659633e-01
1: p01	2.477023225042e-01	6.207465240537e-01
2: p1	1.000000000000e+00	1.197064786943e+02
3: p2	3.961638701883e-02	4.581898317913e-02
4: p3	2.381385210283e-03	2.381385210283e-03
5: p4	9.875864166586e-01	9.875864166586e-01
6: p5	1.241358334141e-02	1.241358334141e-02
7: p6	9.852050314483e-01	9.852050314483e-01
8: p7	0.000000000000e+00	0.000000000000e+00
TRANSITION	Pr[enabled]	Av[throughput]
2: t1	1.000000000000e+00	6.250000000000e-03
3: t2	2.432877113886e-01	2.432877113886e-02
4: t3	1.343349466673e-03	6.716747333365e-04
5: t4	1.312119561869e-03	1.312119561869e-02
6: t5	1.241358334141e-02	1.241358334141e-01
9: t8	9.852050314483e-01	1.379287044028e-01

```

=====
INPUT: Number of stations: = 10
INPUT: Channel capacity(Gbps): = 1
INPUT: Ave. length of packets(k): = 8
INPUT: Load fraction of ring: = 1
INPUT: Ring arrival rate: = 0.29
INPUT: Guessed service rate: = 0.1
EXPECTED: throughput = = 0.0500014200416
EXPECTED: robot utilization = = 0.215646085416
AVERAGE:

```

PLACE	Pr[nonempty]	Av[tokens]
0: p0	3.539690042125e-01	7.048550849638e-01
1: p01	4.753807253500e-01	1.196667816436e+00
2: p1	1.000000000000e+00	1.193089731892e+02



3: p2	1.540972136634e-01	2.156460854164e-01
4: p3	1.060405977157e-02	1.060405977157e-02
5: p4	9.748336089911e-01	9.748336089911e-01
6: p5	2.516639100889e-02	2.516639100889e-02
7: p6	9.642295492195e-01	9.642295492195e-01
8: p7	0.000000000000e+00	0.000000000000e+00
TRANSITION	Pr[enabled]	Av[throughput]
2: t1	1.000000000000e+00	1.250000000000e-02
3: t2	4.543775285466e-01	4.543775285466e-02
4: t3	9.127334373873e-03	4.563667186937e-03
5: t4	2.339898934298e-03	2.339898934298e-02
6: t5	2.516639100889e-02	2.516639100889e-01
9: t8	9.642295492195e-01	2.796265692737e-01

```

=====
INPUT: Number of stations: = 10
INPUT: Channel capacity(Gbps): = 1
INPUT: Ave. length of packets(k): = 8
INPUT: Load fraction of ring: = 1.5
INPUT: Ring arrival rate: = 0.45
INPUT: Guessed service rate: = 0.1
EXPECTED: throughput = = 0.0750118300481
EXPECTED: robot utilization = = 0.586255814694
AVERAGE:
=====

```

PLACE	Pr[nonempty]	Av[tokens]
0: p0	4.936131327575e-01	9.808278942917e-01
1: p01	6.656032547816e-01	1.681585124835e+00
2: p1	1.000000000000e+00	1.187481409305e+02
3: p2	3.253770399462e-01	5.862558146943e-01
4: p3	2.878281870975e-02	2.878281870975e-02
5: p4	9.621967363361e-01	9.621967363361e-01
6: p5	3.780326366387e-02	3.780326366387e-02
7: p6	9.334139176264e-01	9.334139176264e-01
8: p7	0.000000000000e+00	0.000000000000e+00
TRANSITION	Pr[enabled]	Av[throughput]
2: t1	1.000000000000e+00	1.875000000000e-02
3: t2	6.133912414562e-01	6.133912414562e-02
4: t3	2.734541180494e-02	1.367270590247e-02
5: t4	2.833092039072e-03	2.833092039072e-02
6: t5	3.780326366387e-02	3.780326366387e-01
9: t8	9.334139176264e-01	4.200362629319e-01

```

=====
INPUT: Number of stations: = 10
INPUT: Channel capacity(Gbps): = 1
INPUT: Ave. length of packets(k): = 8
INPUT: Load fraction of ring: = 1.75
=====

```

INPUT: Ring arrival rate: = 0.5  
 INPUT: Guessed service rate: = 0.04  
 EXPECTED: throughput = = 0.0785941542434  
 EXPECTED: robot utilization = = 88.2495407444  
 AVERAGE:

```

=====
PLACE                Pr[nonempty]          Av[tokens]
0: p0                7.499618717400e-01   1.499911791277e+00
1: p01               9.999666442216e-01   2.499954785609e+00
2: p1                9.575680282626e-01   3.075049261137e+01
3: p2                9.998134792957e-01   8.824954074441e+01
4: p3                8.732083320184e-02   8.732083320184e-02
5: p4                9.606980263101e-01   9.606980263101e-01
6: p5                3.930197368987e-02   3.930197368987e-02
7: p6                8.733771931083e-01   8.733771931083e-01
8: p7                0.000000000000e+00   0.000000000000e+00
  TRANSITION        Pr[enabled]          Av[throughput]
2: t1                9.575680282626e-01   2.094680061824e-02
3: t2                8.733454223980e-01   3.493381689592e-02
4: t3                8.732067469502e-02   4.366033734751e-02
5: t4                8.522307902606e-07   8.522307902606e-06
6: t5                3.930197368987e-02   3.930197368987e-01
9: t8                8.733771931083e-01   4.366885965541e-01
=====
  
```

## BIBLIOGRAPHY

1. T., Murata, "Petri Nets: Properties Analysis and Application." *Proc. of the IEEE*. 77(4) (1989) : 541-579.
2. ANSI/IEEE Standard 802.4, *Token-Passing Bus Access Method*, New York: IEEE Press (1985).
3. G. Ciardo, J. Muppala, and K. S. Trivedi. "SPNP: Stochastic Petri Net Package." in *Proc. 3rd Int. Workshop Petri Net and Performance Models*, Kyoto, Japan (1989): 142-151.
4. G., Ciardo, and J. K. Muppala. "Manual for the SPNP Package." Duke University, Durham(1990).
5. Rene David and Hassane Alla, *Petri Nets and Grafcet*. Prentice Hall International (UK) Ltd, (1992).
6. J. L., Peterson, *Petri Net Theory and the Modeling of Systems*. Englewood Cliffs, NJ: Prentice Hall (1981).
7. E. C. Foudriat, K. Maly, C.M. Overstreet, S. Khanna and F. Pattera. "A Carrier Sensed Multiple Access Protocol High Data Rate Ring Networks." in *Computer Communication Review*, (1991).
8. C., Oliver, and K. S. Trivedi. "Stochastic Petri Net Models of Polling Systems." *IEEE J. on Select Area Commu.* 8(1990): 1649-1657.
9. M. C., Zhou, K. Mcdermott, P.A. Patel, and T. Tang. "Construction of Petri Net Based Mathematical Models of an FMS Cell." *Proc. of IEEE International Conf. on Syst., Man, and Cybernetics*. Charlottesville, Virginia (1991): 367-372.
10. M. C., Zhou, and F. DiCesare. "Modeling Buffers in Automated Manufacturing Systems Using Petri Nets." *Rensselaer's Second International Conference on Computer-Integrated Manufacturing*, Troy, NY (1990): 265-272.
11. A. S., Taneubaum, *Computer Networks*, Second Edition. Prentice Hall, Englewood Cliffs, N. J. (1988).
12. Biswanath Mukherjee, "WDM-Based Local Lightware Networks." *IEEE, Network* (1992).

**BIBLIOGRAPHY**  
(Continued)

13. M. C., Zhou, Frank DiCesare, and DianLong Guo, " A Moment Generating Function Based Approach for Evaluating Extended Stochastic Petri Nets." *IEEE Trans. on Automation Control*, vol.38, No. 2 (Feb. 1993): 321-327.
14. C. Salvatore, D. S. Antonella and M. Orazio, " Optimization of Acyclic Bandwidth Allocation Exploiting the Priority Mechanism in the Fieldbus Data Link Layer," *IEEE Trans. Industrial Electron.*, June 1993. Vol. 38, No. 3.
15. M. Schwartz, *Telecommunication Networks Protocols, Modeling and Analysis*. Reading, MA: Addison-Wesley, pp. 44.
16. C. Vincenzo, P. Antonio and V. Lorenzo, " " *IEEE Trans. Industrial Electron.* Oct. 1993. Vol. 40, No. 5.
17. S. M. Ross, *Stochastic Process*. New York: Wiley, 1983.
18. J. F. Meyer, " On Evaluating the Performability of Degradable Computing Systems," *IEEE Trans. Comput.*, Vol. C-29, pp. 720-731, Aug. 1980.
19. M. C. Zhou and Z. Pan. " A Petri Net Method for Modeling and Performance of Token Bus LAN." *Proc. of 1992 Reginal Control Conference*. pp. 119-121, Brooklyn, NY. July 24-25, 1992