우선순위 토큰링의 모델링

Modeling of Prioritized Token Ring

채 기 준 *
Ki-Joon Chae*

Abstract

Analytic and simulation models for prioritized token ring are presented in this paper. Its protocol is based on prioritized token ring with reservation (R-PTR). Since the protocol of the R-PTR is simple and the performance of the R-PTR is not inferior to that of the IEEE-PTR under almost all traffic load environments, we use the R-PTR as our token ring model. By using the properties of Markovian process, the expressions for average throughput and average packet transmission delay are derived. The results obtained from the analytic model are compared with that of the discrete event simulation model.

1. Introduction

Token ring has become one of the most popular local area networks because of its simple logic, efficient performance and fairness.

In token ring protocol, a unique message type known as the free token circulates around the ring when all stations are idle. When a station that seizes a free token does not have any waiting packets, it simply passes the free token to the next station. A station wishing to transmit must wait until it detects a free token passing by. If it seizes a free token, it changes the free token to a busy token and transmits its packet which is appended to the end of the busy token. When the station completes the data transmission, it purges its transmitted packet. The station changes the busy token to a free token when the station has finished its transmission and the busy token has returned to the station.

Although token ring access protocols without priority scheme have been analyzed in many papers [1-14], only a few attempts have been made to analyze prioritized token ring protocols. Bassiouni and Gupta [15] present a heuristic algorithm for evaluating average waiting times for asymmetric token rings with priority classes. Gianini and Manfield [16] analyzed symmetric polling systems with two priority classes. Shen et al. [17] propose two types of prioritized token ring access protocols and analyze the performances of those

* 이화여자대학교 전자계산학과

protocols.

Our token ring protocol also includes the priority operation and is based on prioritized token ring with reservation (R-PTR) [17] which is similar to IEEE standard 802.5 [18] (IEEE-PTR). In our model, each station has only one buffer while in [16] and [17] it has multiple buffers according to the number of priority levels. Comparing with the existing analytic models using the multiple buffers for priority levels to analyze the prioritized token fing, our model using only one buffer is more realistic since the real sustems have only one buffer to store the packets.

In the R-PTR, the priority operation is accomplished by assigning a priority to the free token. When a station seizes a free token whose priority level is equal to or less than that of its access waiting packet, it changes the free token to a busy token and transmits its waiting packet. During one-round circulation, a busy token collects the information about the highest priority of all access waiting packets at all stations. When the station completes the data transmission, it purges its transmitted packet. The station generates a new free token with the highest priority collected during one-round circulation of the busy token when the station has finished its transmission and the busy token has returned to the station.

Shen et al. [17] examined the fairness of transmission of each prioritized packet in the R-PTR and the IEEE-PTR and concluded that there is no difference between them about the fairness. They also proved that the performance of the R-PTR is not inferior to that of the IEEE-PTR under almost all traffic load environments. Because of the above reasons, we use the R-PTR as our token ring model.

In this paper, analytic and simulation models to compute the average throughput and packet transmission delay of the R-PTR protocol are presented.

2. Analytic Model

2.1. Model Description

The model is shown in Figure 1 and can be characterized by the following assumptions:

(1) The number of stations is N and the priority level of a packet is a uniformly distributed random integer between 0 and 7. Level 0 is the lowest priority and level 7 is the highest priority.

(2) Each station has only one buffer whose size is one.

(3) "Round-robin" packet transmission strategy is considered, i.e., a station can transmit only one packet each time it captures a free token.

(4) The state of the token ring is represented by (n_0, $\cdots$, n_7, f), where n_i is the number of stations with a level i packet and f is the priority level of a free token. The sum of all n_i's must be less than or equal to N.

(5) A stream of packets arrives at each station according to a Poisson process with mean value λ.

(6) The channel times are normalized by the transmission time of a packet, i.e., the size of a packet is 1 unit time.

(7) The propagation delay between the nearest two stations is r unit time.

In Figure 1, a free token arrives at station A, and then station B. We consider the following four time instants:

tA : the instant that station A seizes a free token

tA' : the instant that station A issues a free token

tB : the instant that station B seizes the free token issued by station A

tB' : the instant that station B issues a free token.

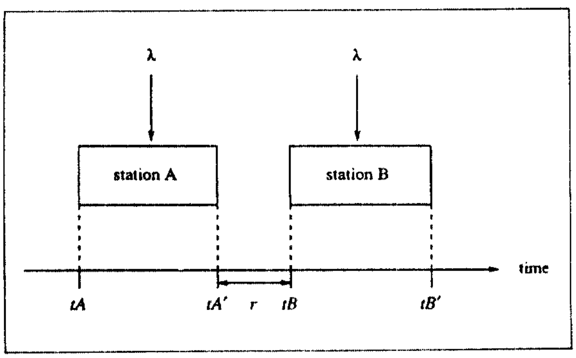

Figure 1. Tokey Ring Model

Each time instant is a Markov renewal point since the state at any instant depends on both the state at the previous instant and the number of arrivals between the current and previous instants.

2.2. Steady State Probability

2.2.1. State Transition Probability from tA' to tB

We assume that the states at instant tA' and instant tB are represented by ($n_0, \cdots, n_7, f$) and ($n_0', \cdots, n_7', g$), respectively. We also represent the state transition probability from tA' to tB by P($n_0', \cdots, n_7', g \mid n_0, \cdots, n_7, f$).

Since tB is the time instant that the station B captures a free token issued by the station A at tA', the priority level of a free token at tB, g, must be equal to the priority level of a free token at tA', f.

From assumptions (1), (5) and (7), we find ($1-e^{-\lambda r}$) / 8 and ($e^{-\lambda r}$) are the probabilities that any station has one or more arrivals of any priority level packet and no arrivals, respectively, between tA' and tB. $\left(N - \sum_{j=0}^{7} n_j\right)$ and $\left(N - \sum_{j=0}^{7} n_j'\right)$ stations are idle at time instants tA' and tB, respectively, and ($n_j' - n_j$) stations have arrivals of level j packet between tA' and tB. Therefore, by using the multinomial distribution property, we obtain the state transition probability P($n_0', \cdots, n_7', g \mid n_0, \cdots, n_7, f$) as follows:

$$\mathrm{P}(n_0', \cdots, n_7', g \mid n_0, \cdots, n_7, f) = \begin{cases} \dfrac{\left(N-\sum_{j=0}^{7} n_j\right)!}{\left[\prod_{j=0}^{7}\left(n_j'-n_j\right)!\right]\left(N-\sum_{j=0}^{7} n_j'\right)!} \left[\frac{1}{8}\left(1-e^{-\lambda r}\right)\right]^{\sum_{j=0}^{7}(n_j'-n_j)} \left(e^{-\lambda r}\right)^{N-\sum_{j=0}^{7} n_j'} & \text{if } f=g \\ 0 & \text{if } f \neq g \end{cases}$$

where $n_j \le n_j' \le N$, $j = 0, \cdots, 7$ and $f, g = 0, \cdots, 7$.

2.2.2. State Probability at tB

The probability that station B has a priority level i packet at time instant tB with state ($n_0', \cdots, n_7', f$) is denoted by P($n_0', \cdots, n_7'$) and computed as follows:

$$\mathrm{P}_i(n_0', \cdots, n_7') = \begin{cases} \dfrac{\dfrac{(N-1)!}{(n_0')!\cdots(n_i'-1)!\cdots(n_7')!\left(N-\sum_{j=0}^{7} n_j'\right)!}}{\dfrac{N!}{(n_0')!\cdots(n_7')!\left(N-\sum_{j=0}^{7} n_j'\right)!}} = \dfrac{n_i'}{N} & \text{if } n_i' \neq 0 \\ 0 & \text{if } n_i' = 0 \end{cases}$$

where $0 \le n_i' \le N$, $i = 0, \cdots, 7$.

2.2.3. State Transition Probability from tB to tB'

The state transition probability from instant tB to instant tB' depends on the priority level of a packet in station B and the level of free tokens arriving at and generated from station B. If no packet is transmitted, the time duration tB'-tB is equal to zero. If a packet is transmitted, then tB'-tB depends on the packet transmission time. If the packet transmission time is greater than the round trip propagation delay, tB'-tB is equal to the packet transmission time, 1. If not, it is equal to the propagation delay, Nr. Consequently, tB'-tB is equal to Max (1, Nr) which is represented by t.

We consider two cases. The first case is that both states at tB and tB' are the same while the second case is that the states are different. In the following equations, X denotes the probability that any idle station has a packet arrival before a busy token passes the station.

Case 1:

This case occurs when station B has a packet whose priority level is less than the level of the free token arriving at the station or does not have any packet. This case also happens when station B has a packet whose priority level is equal to or greater than the level of the free token and the priority level collected by the busy token during the packet transmission is equal to the level of the free token. We assume the distances between any two nearest idle stations are the same. The state transition probability of this case is given as follows:

$$P(n_0',\cdots,n_7',i \mid n_0',\cdots,n_7',i)=1-\sum_{j=i}^{7}P_j(n_0',\cdots,n_7')$$
$$+P_i(n_0',\cdots,n_7')\begin{pmatrix} N-\sum_{j=0}^{7}n_j'+1 \\ 1\end{pmatrix}\left[\frac{1}{8}\left(1-e^{-\lambda\tau}\right)\right]\left(e^{-\lambda\tau}\right)^{N-\sum_{j=0}^{7}n_j'}\cdot R(i)+PA$$

where

$$R(i)=\begin{cases}0 & \text{if } (i<7) \text{ and } ((n_{i+1}'\geq 1) \text{ or } \cdots \text{ or } (n_7'\geq 1))\\ 1 & \text{if } ((i=7) \text{ and } (n_7'\geq 2)) \\ & \quad \text{or } ((i<7) \text{ and } (n_i'\geq 2) \text{ and } (n_{i+1}'=\cdots=n_7'=0)) \\ X=1-\exp\left[(-\lambda)\left(\dfrac{N}{N-\sum_{j=0}^{7}n_j'+1}\cdot\tau\right)\dfrac{\left(N-\sum_{j=0}^{7}n_j'+1\right)\left(N-\sum_{j=0}^{7}n_j'+2\right)}{2}\right] & \\ \quad \text{otherwise} & \end{cases}$$

$$PA=\begin{cases}0 & \text{if } i=7\\ \sum_{k=i+1}^{7}\left[P_k(n_0',\cdots,n_7')\begin{pmatrix} N-\sum_{j=0}^{7}n_j'+1 \\ 1\end{pmatrix}\left[\frac{1}{8}\left(1-e^{-\lambda\tau}\right)\right]\left(e^{-\lambda\tau}\right)^{N-\sum_{j=0}^{7}n_j'}\cdot R'(k)\right] & \\ \quad \text{otherwise} & \end{cases}$$

where

$$R'(k)=\begin{cases}0 & \text{if } (n_i'=0) \text{ or } (n_{i+1}'\geq 1) \text{ or } \cdots \text{ or } (n_{k-1}'\geq 1) \text{ or } (n_k'\geq 2)\\ & \quad \text{or } (n_{k+1}'\geq 1) \text{ or } \cdots \text{ or } (n_7'\geq 1)\\ 1-X & \text{otherwise.}\end{cases}$$

Case 2:

This case occurs when station B has a packet whose priority level is equal to or greater than the level of a free token arriving at the station.

$$P(n_0'',\cdots,n_7'',g \mid n_0',\cdots,n_7',i)=\sum_{k=i}^{7}\left\{P_k(n_0',\cdots,n_7')\frac{\left(N-\sum_{j=0}^{7}n_j'+1\right)!}{\left[\prod_{j=0}^{7}\left(n_j''-n_j'\right)!\right]\left(n_k''-n_k'+1\right)!\left(N-\sum_{j=0}^{7}n_j''\right)!}\right.$$
$$\left.\cdot\left[\frac{1}{8}\left(1-e^{-\lambda\tau}\right)\right]^{\sum_{j=0}^{7}(n_j''-n_j')+1}\left(e^{-\lambda\tau}\right)^{N-\sum_{j=0}^{7}n_j''}\cdot R''(k)\right\}$$

where $(n_0'',\cdots,n_7'',g)\neq(n_0',\cdots,n_7',i)$. $R''(k)$ has different values depending on the priority levels of a packet and free tokens and is denoted as follows:

(1) $(g<i\leq k)$ or $(i\leq g<k)$

$$R''(k)=\begin{cases}0 & \text{if } (n_{g+1}'\geq 1) \text{ or } \cdots \text{ or } (n_{k-1}'\geq 1) \text{ or } (n_k'\geq 2)\\ & \quad \text{or } (n_{k+1}'\geq 1) \text{ or } \cdots \text{ or } (n_7'\geq 1)\\ 1-X & \text{if } (n_g'\geq 1) \text{ and } (n_k'<2)\\ & \quad \text{and } (n_{g+1}'=\cdots=n_{k-1}'=n_{k+1}'=\cdots=n_7'=0)\\ X(1-X) & \text{otherwise}\end{cases}$$

(2) $(i\leq g=k)$

(i) $k<7$

$$R''(k)=\begin{cases}0 & \text{if } (n_{g+1}'\geq 1) \text{ or } \cdots \text{ or } (n_7'\geq 1)\\ 1-X & \text{if } (n_g'\geq 2) \text{ and } (n_{g+1}'=\cdots=n_7'=0)\\ X(1-X) & \text{otherwise}\end{cases}$$

(ii) $k=7$

$$R''(k)=\begin{cases}1 & \text{if } (n_g'\geq 2)\\ X & \text{otherwise}\end{cases}$$

(3) $(i\leq k<g)$

(i) $g<7$

$$R''(k)=\begin{cases}0 & \text{if } (n_{g+1}'\geq 1) \text{ or } \cdots \text{ or } (n_7'\geq 1)\\ 1-X & \text{if } (n_g'\geq 1) \text{ and } (n_{g+1}'=\cdots=n_7'=0)\\ X(1-X) & \text{otherwise}\end{cases}$$

(ii) $g=7$

$$R''(k)=\begin{cases}1 & \text{if } (n_g'\geq 1)\\ X & \text{otherwise}\end{cases}$$

2.2.4. Steady State Probability

Using the state transition probabilities obtained above, we compute the steady state probability $\pi\ (n_0'',\ \cdots\ ,n_7'',g)$ as follows:

$$\pi(n_0'',\cdots,n_7'',g)=\sum_{\substack{0\leq\sum_{j=0}^{7}n_j,\ \sum_{j=0}^{7}n_j'\leq N\\ f=0,\cdots,7}}\{P(n_0'',\cdots,n_7'',g \mid n_0',\cdots,n_7',f)$$
$$\cdot P(n_0',\cdots,n_7',f \mid n_0,\cdots,n_7,f)\cdot\pi(n_0,\cdots,n_7,f)\}.$$

2.3. Throughput and Delay Analysis

2.3.1. Average Throughput

First, we consider the probability $P(k|n_0, \cdots, n_7, f)$ that the transmission time, $tB'-tB$, is equal to k when the state at time instant tB is $P(k|n_0, \cdots, n_7, f)$. If station B has a packet whose priority level is less than the level f of the free token arriving at the station or does not have any packet, the transmission time is 0. However, if the station has a packet whose priority level is greater than or equal to f, the packet is transmitted and the transmission time is equal to t. Thus, the probability is represented as follows:

$$g_k = P(k|n_0, \cdots, n_7, f) = \begin{cases} 1 - \sum_{j=f}^{7} P_j(n_0, \cdots, n_7) & \text{if } k = 0 \\ \sum_{j=f}^{7} P_j(n_0, \cdots, n_7) & \text{if } k = t \\ 0 & \text{otherwise} \end{cases}$$

In addition, the probability generating function of the probability g_k is represented by $G(z|n_0, \cdots, n_7, f)$ and computed as follows:

$$\begin{aligned} G(z|n_0, \cdots, n_7, f) &= \sum_k z^k g_k = z^0 g_0 + z^t g_t \\ &= 1 - \sum_{j=f}^{7} P_j(n_0, \cdots, n_7) + z^t \cdot \sum_{j=f}^{7} P_j(n_0, \cdots, n_7) \\ &= 1 + (z^t - 1) \cdot \sum_{j=f}^{7} P_j(n_0, \cdots, n_7) \end{aligned}$$

Then, the first derivative of $G(z|n_0, \cdots, n_7, f)$ evaluated at $z = 1$ yields the average transmission time $F(n_0, \cdots, n_7, f)$ and is given by

$$\begin{aligned} F(n_0, \cdots, n_7, f) &= G^{(1)}(1|n_0, \cdots, n_7, f) = \frac{dG(z|n_0, \cdots, n_7, f)}{dz}\bigg]_{z=1} \\ &= t \cdot \sum_{j=f}^{7} P_j(n_0, \cdots, n_7) = \frac{t \cdot \sum_{j=f}^{7} n_j}{N} \end{aligned}$$

From the above equations, we now derive the average throughput of any priority level packet. In the following equation, T_i denotes the average throughput of level i packet.

$$T_i = \frac{\sum_{\substack{0 \le \sum_{j=0}^{7} n_j \le N \\ f = 0, \cdots, 7}} \pi(n_0, \cdots, n_7, f) \cdot PR_i(n_0, \cdots, n_7, f)}{\sum_{\substack{0 \le \sum_{j=0}^{7} n_j \le N \\ f = 0, \cdots, 7}} \pi(n_0, \cdots, n_7, f) \cdot F(n_0, \cdots, n_7, f)} \qquad i = 0, \cdots, 7$$

where $PR_i(n_0, \cdots, n_7, f)$ is the probability that a priority level i packet is transmitted at state $(n_0, \cdots, n_7, f)$. Since a packet is transmitted when its priority level i is equal to or greater than the priority level f of a free token, we have the following equation for $PR_i(n_0, \cdots, n_7, f)$:

$$PR_i(n_0, \cdots, n_7, f) = \begin{cases} 0 & \text{if } i < f \\ P_i(n_0, \cdots, n_7) & \text{otherwise} \end{cases}$$

2.3.2. Average Transmission Delay

To derive the average transmission delay, we first consider the probability $PN_i(m_i|n_0, \cdots, n_7, f, k)$ where m_i is the number of stations which have level i packet at the time instant that k time units have elapsed after a free token was issued in the state $(n_0, \cdots, n_7, f)$. The probability is computed as follows:

$$PN_i(m_i|n_0, \cdots, n_7, f, k) = \binom{N - \sum_{j=0}^{7} n_j}{m_i - n_i} \cdot \left[\frac{1}{8}(1 - e^{-\lambda k})\right]^{m_i - n_i} \cdot (e^{-\lambda k})^{N - \sum_{j=0}^{7} n_j - m_i + n_i}$$

where $n_i \le m_i \le N - \sum_{j=0}^{7} n_j + n_i$ and $i = 0, \cdots, 7$.

Next, we calculate the average number of level i packets in the network at the instant that k time units have elapsed after a free token was issued in the state $(n_0, \cdots, n_7, f)$. This number is denoted by $N_i(n_0, \cdots, n_7, f, k)$ and given by

$$\begin{aligned} N_i(n_0, \cdots, n_7, f, k) &= \sum_{m_i = n_i}^{n_i + N - \sum_{j=0}^{7} n_j} m_i \cdot PN_i(m_i|n_0, \cdots, n_7, f, k) \\ &= \left[\frac{1}{8}(1 + 7e^{-\lambda k})\right]^{N - \sum_{j=0}^{7} n_j - 1} \cdot \left[\frac{1}{8}(1 - e^{-\lambda k}) \cdot \left(N - \sum_{j=0}^{7} n_j\right) + \frac{n_i}{8}(1 + 7e^{-\lambda k})\right] \end{aligned}$$

In addition, we consider the average number of level i packets in the network from the generation of a free token in state ($n_0, \cdots, n_7, f$) to the next generation of a free token. This number is given as follows:

$$M_i(n_0, \cdots, n_7, f) = N_i(n_0, \cdots, n_7, f, t+r) \cdot P(t|n_0, \cdots, n_7, f)$$

From the above equations, we compute the average number of level i packets in the network at any arbitrary time and this is given by the following equation.

$$L_i = \frac{\sum_{\substack{0 \le \sum_{j=0}^{7} n_j \le N \\ f=0,\cdots,7}} \pi(n_0, \cdots, n_7, f) \cdot M_i(n_0, \cdots, n_7, f)}{\sum_{\substack{0 \le \sum_{j=0}^{7} n_j \le N \\ f=0,\cdots,7}} \pi(n_0, \cdots, n_7, f) \cdot P(n_0, \cdots, n_7, f)} \qquad i = 0, \cdots, 7$$

We now know the average throughput and the average number of any priority level packets. Therefore, we can use Little's law to calculate the average transmission delay D_I as follows:

$$D_I = L_I / T_I$$

Although Di does not have the closed form, it can be easily calcalated using computer.

3. Simulation Model

We use discrete event simulation as our simulation model. Discrete event simulation concerns the modeling of a system as it evolves over time by a representation in which the state variables change only at a countable number of points in time. These points in time are the ones at which an event occurs, where an event is defined to be an instantaneous occurrence which may change the state of a system.

Because of the dynamic nature of discrete event simulation model, we need to keep track of the current value of simulated time as the simulation proceeds, and we also need a mechanism to advance simulated time from one value to another. In our simulation model, the current value of model time is maintained in TNOW.

Figure 2 shows a header file which contains the definitions used by our discrete event simulation model. Our model maintains the event list for a discrete event simulation, using the procedures GetEvent, Schedule, StartSim and CountIt. In order to make it easy to add new event in the event list, we use a linked list as a data structure of the event list.

Each event is specified by the following elements. The first and second elements 'what' and 'when' indicate what type of event takes place and when the event occurs, respectively. The third element 'info' contains the information about the station at which the event takes place.

In our simulation, model time is advanced by the procedure GetEvent, which returns the activity to take place at that time in the parameter 'action'. The activities corresponding to 'action' are assumed to take no model time, that is, time is advanced only by GetEvent.

The scheduling of future events is done by a call to the procedure Schedule. The passed parameter 'dt' indicates the amount of time from the current model time until the scheduled event is to occur.

The procedure StartSim is used to initialize the simulation. This procedure has two parameters, stopTime and limit. The first parameter indicates the model time at which the simulation is to be halted. If it is not desired to end the simulation at a particular time, then the parameter stopTime should be set to a negative number.

The second parameter, limit, is used to stop the simulation after a pre-specified number of occurrences of calls to the procedure CountIt. CountIt is called without parameters after a call to Schedule. It causes an internal counter to be incremented by one after the event scheduled is actually executed. If we wish to increment the counter by more than one, just call CountIt the appropriate number of times. The internal counter is not tied to any specific event, though typically a call to CountIt will be made only after scheduling some particular event of interest.

The simulation is also halted by a total event counter. This

```
const
    MaxNumEv = 500000;
type
    EVTYP = (recMsg, getToken, ifCollide, BusC, mtCB, CRecMes, getT, recM, ClearStats, count, stopSim);
    PARMBLK= record
      station: integer;
    end;
    EVPNT = ^EVENT;
    EVENT= record
      what : EVTYP;
      when : real;
      info : PARMBLK;
      next : EVPNT;
    end;
procedure GetEvent(var action:EVTYP; var parm:PARMBLK);
    external;
procedure Schedule(action:EVTYP; parm:PARMBLK; dt:real);
    external;
procedure StartSim(stopTime:real; limit:integer);
    external;
procedure CountIt;
    external;
```

Figure 2. Header File for Discrete Event Simulation

counter is incremented for every event, and prevents the simulation from running forever. This counter is not directly tied to the CountIt counter. The maximum number of events for this counter is stored in the constant MaxNumEv. The simulation always halts when the event list is empty.

The types of events which can take place are defined in the type definition EVTYP. The event 'stopSim' must be included in this definition. This event is returned when the simulation stopTime is reached, or when the event list is empty. It can also be used to halt the simulation at any other time, by scheduling the event 'stopSim' at that time. The event 'count' must also always be included in the event definition. This event is tied to CountIt.

The type PARMBLK is also for user definition. Each event which is scheduled has a PARMBLK associated with it. The PARMBLK can contain any information that the user wishes to define for an event. In our model, it contains only the information about the station at which the event takes place.

Statistics are collected on a quantity by a call to procedure 'Collect' with the first parameter set to the name of the quantity, and the second to the value of the observation.

We have used the above discrete event simulation models to validate the analytic model developed in Section 2.

4. Performance Measurements

In order to validate results obtained using the analytic model, the results are compared with results from the simulation model. For simalation model, we compute the average delay from 1000-1500 packets depending on the amount of arriving pacleets. For simplicity of analysis, we assume there are two levels of priority, 0 and 1, where 1 is the higher level. When different values are assigned to a certain parameter, the other parameters have the following values:

Transmission rate : 4 Mbits/sec
Number of stations : 10
Delay between two stations : 2 micro-seconds
Packet inter-arrival time : 0.05 seconds
Packet length : 1000 bytes

Figure 3 graphically shows the average transmission delay as a function of packet inter-arrival time and Figure 4 shows the delay as a function of packet length. In both Figures, we have very good results for level 1 packets while we have a small discrepancy for level 0 packets. This discrepancy is caused by longer waiting time in any station on the network for level 0 packets.

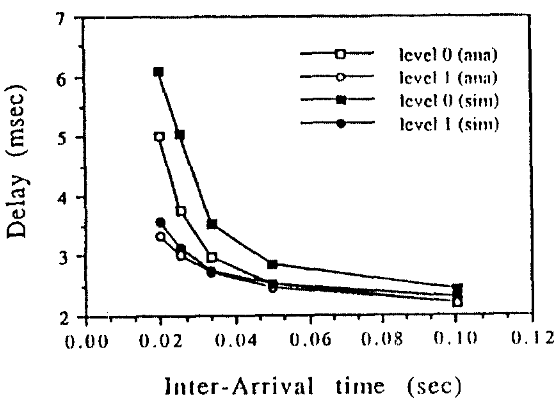

Figure 3. Transmission Delay vs. Inter-Arrival Time

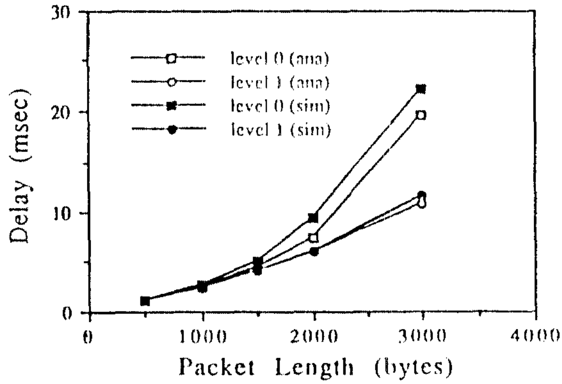

Figure 4. Tramsmission Delay vs. Paceet Lergth

5. Conclusions

In this paper, we have introduced a prioritized token ring protocol and an analytic model for evaluating its performance. The average throughput and transmission delay are obtained by means of a Markov chain model. In addition, average packet transmission delay is graphically shown as functions of inter-arrival time and packet length.

In order to validate the analytic model, the results are compared with the results obtained from simulation. From the comparison, we have observed that analytic and simulation results are consistent with each other for high priority level packet. However, there is a small discrepancy for low priority level packet but this discrepancy is acceptable.

REFERENCES

[1] A. G. Konheim and B. Meister, "Waiting lines and times in a system with polling," *J. ACM*, vol. 21, no. 3, 1974, pp.470-490.

[2] P. J. Kuehn, "Multiqueue systems with nonexhaustive cyclic service," *Bell Syst. Tech. J.*, vol. 58, no. 3, 1979, pp. 671-698.

[3] W. Bux, "Local-area Subnetworks: A performance comparison," *IEEE Trans. Commun.*, vol. COM-29, no. 10, 1981, pp. 1465-1473.

[4] W. Bux and H. L. Truong, "Mean-delay approximation

for cyclic service queueing systems," *Perform. Eval.*, vol. 13, no. 3, 1983, pp. 187-196.

[5] R. F. Berry and K. M. Chandy, "Performance models of token ring local area networks," *Perform. Eval. Rev. (Special Issue)*, 1983, pp. 266-274.

[6] A. S. Sethi and T. Saydam, "Performance analysis of token ring local area networks," *Comput. Network ISDN Syst.*, vol. 9, 1984, pp. 191-200.

[7] K. S. Watson, "Performance evaluation of cyclic service strategies - A survey," *Perform. '84*, North Holland, 1985, pp. 521-533.

[8] M. J. Ferguson and Y. J. Aminetzah, "Exact results for nonsymmetric token ring systems," *IEEE Trans. Commun.*, vol. COM-33, 1985, pp. 223-231.

[9] D. E. Everitt, "Simple approximations for token ring," *IEEE Trans. Commun.*, vol. COM-34, 1986, pp. 719-721.

[10] H. Takagi, "Analysis of polling systems," Cambridge, MA: The M.I.T. Press, 1986.

[11] O. J. Boxma and B. Meister, "Waiting-time approximations for cyclic- service systems with switchover times," *Perform. Eval.*, vol. 7, no. 4, 1987, pp. 299-308.

[12] O. J. Boxma and W. P. Groenendijk, "Pseudo-conservation laws in cyclic service systems," *J. Appl. Prob.*, vol. 24, no. 4, 1987, pp. 949-964.

[13] A. Ganz and I. Chlamtac, "Queueing analysis of finite buffer token networks," *Proc. ACM SIGMETRICS Conf. Measurement and Modeling of Comput. Syst.*, Santa Fe, New Mexico, 1988, pp. 30-36.

[14] T. Takine, Y. Takahashi and T. Hasegawa, "Exact analysis of asymmetric polling systems with single buffers," *IEEE Trans. Commun.*, vol. 36, no. 10, 1988, pp. 1119-1127.

[15] M. A. Bassiouni and A. Gupta, "Computing estimates of waiting times in ring local area networks with priority classes," *Comput. Syst. Sci. Eng.*, vol. 2, no. 4, 1987, pp. 155-160.

[16] J. Gianini and D. R. Manfield, "An analysis of symmetric polling systems with two priority classes," *Perform. Eval.*, vol. 8, 1988, pp.93-115.

[17] Z. Shen, S. Masuyama, S. Muro and T. Hasegawa, "Performance evaluation of prioritized token ring protocols," *Eleventh International Teletraffic Congress, M. Akiyama (ed.)*, Elsevier Science Publishers B.V. (North-Holland), 1985, pp. 4.2A-3-1 - 4.2A-3-7.

[18] IEEE Standard 802.5, "Local area networks: Token Ring access method and physical layer specifications," 1985.

저자소개

채기준

1982년 2월 연세대학교 수학과 학사

1984년 5월 미국 Syracuse Univ. 전산과학 석사

1990년 5월 미국 North Carolina State Univ. 전산공학 박사

1990년 8월 ~ 1992년 2월 미국 해군사관학교 전자계산학과 조교수

1992년 3월 ~ 현재 이화여자대학교 전자계산학과 조교수

관심분야: 컴퓨터 통신 및 네트워크, 성능평가