The 6th IEEE Singapore International Conference on

NETWORKS '98

IEEE SICON'98

Singapore June 30 – July 3, 1998

Sponsors

3Com

ST Mobile Data

Organizers

National University of Singapore

Computer Chapter, IEEE Singapore Section

In-corporation with

SingAREN

Editors

A. L. Ananda, Hung Keng Pung and

Department of Information Systems and Computer Science

National University of Singapore

Weiguo Wang

Kent Ridge Digital Labs, Singapore

World Scientific

Singapore • New Jersey • London • Hong Kong

CHAPTER 3: FLOW AND CONGESTION CONTROL

Worst case buffer requirements for TCP over ABR (**Invited paper**)
B. Vandalore, S. Kalyanaraman, R. Jain, R. Goyal and S. Fahmy ------------ 111

An oscillation reducing rate-based protocol based on multiple time scale for ABR flow control
Lee Chuan Hu, Yuseok Kim and Wei Kang Tsai ------------------------------ 123

ASAP: a non-per-VC accounting max-min protocol for ARB flow control with optimal convergence speed
Wei Kang Tsai, Yuseok Kim and Lee Chuan Hu ------------------------------ 139

A stability and sensitivity theory for rate-based max-min flow control for ABR service (**Invited paper**)
Wei Kang Tsai, Yuseok Kim and Chai-Keong Toh ---------------------------- 155

A congestion control protocol for ABR-like service based on the use of fuzzy controllers
Jarmo Harju and Kimmo Pulakka --- 171

CHAPTER 4: NETWORK TRAFFIC AND PERFORMANCE MODELING

Efficiency and fairness of multiple TCP connections over UBR in asymmetric cases
Peng Zhang, Jian Ma, Lei Guo and Shiduan Cheng --------------------------- 175

Throughput performance of distributed applications in switched networks in presence of end-system bottlenecks
R. Radhakrishna Pillai and Jit Biswas ------------------------------------ 187

Token holding strategy in a multichannel token-passing network
Wen-Shyang Hwang, Ling-Yang Kung and Jun-Yao Wang ------------------ 205

Analysis of Web server performance towards modeling and performance evaluation of Web systems
Yasuyuki Fujita, Masayuki Murata and Hideo Miyahara --------------------- 221

TOKEN HOLDING STRATEGY IN A MULTICHANNEL TOKEN-PASSING NETWORK

WEN-SHYANG HWANG
Department of Electrical Engineering, National Kaohsiung Institute of Technology, Kaohsiung, Taiwan,R.O.C.

LING-YANG KUNG and JUN-YAO WANG
Department of Electrical Engineering, National Cheng Kung University, Tainan, Taiwan, R.O.C

Expanding a single channel token passing network to a multichannel one does increase the network bandwidth, however problems created by the arrival of multiple tokens at a node should be addressed. This paper presents several strategies to handle these simultaneously arriving tokens. A system model and a node model are described to study the effects of these strategies. The average token cycle and the mean message delay for each strategy are analyzed and simulated to assess their relative performance. Numerical results are plotted to show the effects of these strategies and their relation with the various degrees of parallel processing in a node.

1 Introduction

Multichannel technique is specially useful for those networks which insufficient bandwidth is caused by the increase of users rather than by the rapid increase of processing rate in some nodes, and several proposals were made to study this technique. For example, a Multichannel Local Area Network (**M_LAN**) was proposed by Marsan and Roffinella in 1983 [1]. This network consisted of a set of parallel broadcast CSMA channels. Later, Wong and Benny C. Y. in 1989 [2] proposed a Multichannel Token Bus Local Area Network (**MTB_LAN**). In the same year, Sung and Kung [4] proposed a Multichannel Manufacturing Automation Protocol network (**MMAP**). In 1992, Hwang and Kung [5] proposed a Distributed Multichannel Manufacturing Automation Protocol (**DMMAP**) network. The last three networks comprise a set of parallel **Token Bus** channels. Besides these proposals of multichannel network, there have been many research interests to study their performance.

Generally, the token passing network in heavy load has a better performance than the CSMA/CD network, and its **Token Handling Policy** has a major effect on the network performance. Hence this paper concentrates on the token handling policy of a multichannel token passing network. The policy usually includes the generation, passing, holding and fault tolerance of the token. The token bus network is one of the token passing networks. Its token handling policy is described in IEEE 802.4 standard. However the IEEE 802.4 standard is designed for the single channel token bus network, if it is mapped onto a multichannel network, new problem will be created when multiple tokens arrive at a node.

The MTB_LAN, the MMAP and the DMMAP networks, all keep the token handling policy of the token bus network. The MTB_LAN assigns only one token to operate all channels in synchronization, its bandwidth grows linearly, and it has no multiple token holding problem. However, its bandwidth is not efficiently managed since the token holder always has all network bandwidth regardless of its requirement. In the MMAP network, a **headend** node is used to manage all tokens. On the other hand in DMMAP network, tokens are managed in a distributed manner. Besides, the DMMAP employs only two transmitters per node no matter how many channels the network has, and it can still retain a performance similar to a network without reducing transmitters. All these multichannel networks did not address the problem when multiple tokens arrive at a node. This paper will present some strategies to handle these multiple tokens and discusses their effect on the network performance.

In a token bus network, the network performance is strongly affected by the Token Hold Time **(THT)** setting. In the network, each node has to set its THT to prevent the token from been seized too long. The short setting of THT will reduce the quantity of message transmission of a node at each token cycle, and therefore long message will take more cycles to be transmitted. Conversely, the long THT setting will not only increase the token holding time of a node, but also increase the length of token cycle time.

In order to study the token holding strategy in the multichannel token-passing network, we introduce a new model called Multichannel Token Bus **(MTB)** network. In this network, each channel has a token, and all tokens are fully independent in operation. For the sake of compatibility, the token handling policy in the MTB network conforms to the IEEE 802.4 standard as much as possible, hence the performance affected by the original parameter setting is the same as in the token bus network. In this paper, several strategies to handle the arrival of multiple token to a node will be proposed, analyzed and simulated to study their relative performance. Since the response time of the urgent messages is usually the focus of concerns, hence other lower priority messages in MTB network will not be discussed in this paper.

Section II of this paper discusses the system and the node models of the MTB network. These models are useful for the rest of the paper. Afterward, several token holding strategies are proposed and discussed in section III. In section IV, the average token cycle time is analyzed and simulation result is presented to illustrate the effects on average response time caused by these strategies. Section V gives some observations and explanations of the simulation results on average message delay. Finally, section VI concludes this paper.

2 Model

The physical topology of the MTB network is still a bus, and its logical topology consists of many independent channels. Each channel has its own token, as shown in Figure 1.

2.1 System model

Each node in the logical topology is connected to all channels in the network, hence its message can be transmitted to the other node through any one of these channels.

In Figure 1, the MTB network is presented as M independent broadcast channels with N nodes. Every node is fully connected to all channels. Besides, each channel has its own token that not only inherits the single channel protocol, but also complies with the token holding strategy that will be discussed in the next section.

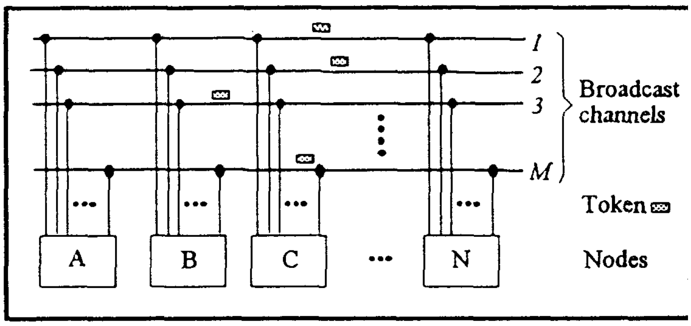


Figure 1. The logical topology of MTB network.

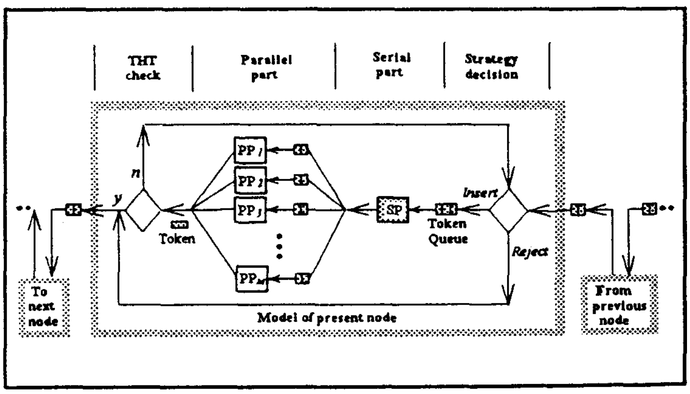


Figure 2. The node model for the multichannel token-passing network.

2.2 Node model

In the MTB network, only the token holding nodes are permitted to transmit messages. While token enters to a node, it first removes message from its transmitted queue if the queue is not empty. In a multichannel network, multiple tokens may arrive simultaneously at a node, then each token will take its turn to execute this operation. Afterward, the removed message will be immediately passed to the transmitter, and will be transmitted through a specific broadcast channel. Since each node has as many transmitters as the number of channels, the transmission operation can be processed in parallel.

In general, when tokens enter a node, some token processing operation should be done in series one token at a time. For example, the arbitration that selects a token to serve a data stream coming from the upper layer, is usually done by one token at a time. Some other operations can be done in parallel to increase throughput, provided that the required hardware for the parallel operations is available. For example, the protocol processing and the data transmission can be done in parallel. All these operations form a multistage tandem queue network that can be reduced with methods shown in [6]. This paper assumes that this tandem queue has been reduced to one SP and a set of PP's as shown in Figure 2. Because there are at most M tokens in a node, so let there be M PP's in our node model to maximize the performance. Furthermore, to simplify the hardware design, each PP processes its own token.

The node model for the token operation in the MTB network is shown in Figure 2. In this figure, the token from the previous node is examined first to see whether it can be inserted into the **Token Queue** that is located before the SP. This decision is made by a strategy that will be discussed in section III. If the result is negative then the rejected token leaves immediately and is passed to the next node. If the token is accepted, it is inserted into the token queue. Messages from the LLC (Logical Link Control) layer are stored in a transmission queue that is not shown in the figure. While the SP holds a token, a message in the transmission queue will be removed and processed. Then, the message and the token will be passed to a specific PP according to the token id. There are many PP's behind the SP, so it is possible that more than one PP operating at any time. After a token has carried out the message transmission in its PP, its THT is examined. If the THT expires, then this token is sent out, otherwise it is inserted back to token queue again.

3 Token Holding Strategy

There are many independent channels in the MTB network, so it is possible that many tokens simultaneously arrive at a node. However, only one token is permitted to enter the SP at a time as shown in the Figure 2. Therefore, a contention may happen, and is called the "**Token-Overlap**." In this section, the strategy for solving this contention will be discussed. Under this strategy, one of the tokens called **SP Token** will enter the SP and other tokens called **Queued Token** will be sequentially inserted into the token queue. Since only the SP token is permitted to serve message, the queued tokens will cause a loss in the network throughput. Therefore, it is worthy to study how many tokens should be held in the SP and in the token queue. The SP together with the token queue forms the Serial Section (SS) in our node model

3.1 Number of tokens in the SS

More queued tokens in a node mean more loss in network throughput, but less queued tokens will also create some side effects. For example, no queued token is allowed in a node and there are five messages to be transmitted in a node. Then two tokens arrive. The first token will enter the SP and serve the first message and the second token will be bypassed to the next node. After an **Extra Cycle**, the second token comes back and serves the second message. It is obvious that the message delay of the second message can be reduced if there is a token queue in the node that can accommodate more tokens. In order to study how many tokens should be held in the SS, three feasible strategies are proposed as follows:

3.1.1 Limit-1 Strategy

In this strategy, there is only one token that can stay in the SP and no token is permitted to stay in the token queue. Hence, nodes pick up the first arrived token, and bypass all other tokens to the next node to increase the network throughput. This strategy has been used for handling multiple tokens in the FDDI and DMMAP network [5].

3.1.2 Limit-η Strategy

In this strategy, every node is allowed to keep at most η tokens in the SS. The value of η can be set by each node according to its ratio of $\overline{x}_p$ to $\overline{x}_s$, where $\overline{x}_s$

and $\overline{x}_p$ are the token processing time in SP and in PP respectively. This strategy needs a scheme to select which token should be held when the number of arrived tokens is greater than η. This strategy could effectively decrease the number of extra cycle and limit the number of tokens clustering in a node.

3.1.3 No-limit Strategy

The no-limit strategy has been used by some multichannel networks [4]. It keeps all arrived tokens without any constraint. In this strategy, nodes put all incoming tokens into the token queue, hence the message delay of token holder may be reduced at the expense of the other nodes. Therefore, this strategy is suitable for the network where system load is centralized to a few nodes. In addition, since no extra cycle happens in this strategy, the system performance may be improved when the token cycle time is much greater than the token queueing time.

3.2 Service discipline of operating token

In the single token network, a token can perform the transmission service until its THT expires. In the MTB network with the limit-η or no-limit strategy, it is possible that many tokens arrive at a node, and only one token can operate in the SP. Since part of data messages is processed in SP and the rest in PP, in order to allow more concurrent operation in PP, a Round Robin Service Discipline **(RRSD)** is proposed for the MTB network. In this discipline, the SP token is permitted to serve only one message. After that, the SP will pass the token and the message to the PP, then the SP will take the next token from the head of token queue to serve another message. While the token in PP completes its service, it will return to the token queue or will depart from the node if its THT expires, as shown in Figure 2.

3.3 Schedule of tokens in a node

In limit-1 and limit-η strategies, nodes must decide which token should be kept when multiple tokens simultaneously arrive. Three practical schemes are proposed to select the SP token as follows. In addition, the limit-1 strategy is a special case of limit-η strategy, hence the following discussion will focus on the limit-η strategy.

3.3.1 Keep Next Token (KNT) scheme

All arriving tokens are put into the token queue first. After every message transmission, the token queue is checked. If its length is greater than η, the operating token will be released to next node and the token at the head of token queue becomes the next operating token. Otherwise, the operating token will continue to operate according to the RRSD. Hence, every token in the token queue will transmit at least one message if available.

3.3.2 Keep Last Token (KLT) scheme

Queued tokens are idle tokens. In order to reduce the number of idle tokens, KLT scheme is proposed. The operation of this scheme is similar to KNT except that when token queue is checked and its length is found to be greater than η, the node will release all but the last η tokens in token queue. Therefore, the queued tokens except last η tokens may not transmit any message.

3.3.3 Reject Later Token (RLT) scheme

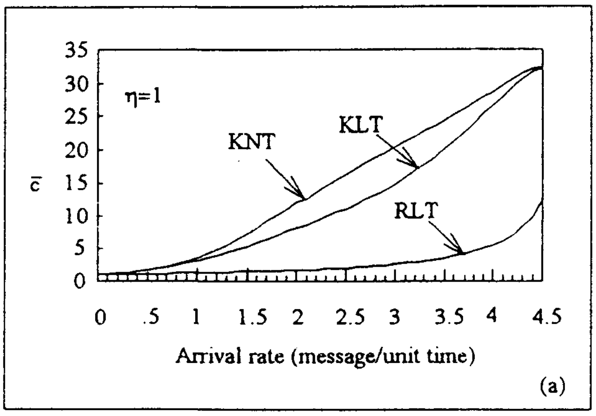
When a node already possesses η tokens, all following tokens will be rejected without putting into the token queue and will be bypassed to the next node immediately. Hence, all later tokens offer no service to a busy node.

From the node's point of view, the KNT scheme provides the most THT to a node and holds more tokens in the token queue than the others. While in the RLT scheme, the number of tokens that held in a node, is always less than η. Hence, the relation for the number of queued tokens is:

$$\#(KNT) > \#(KLT) > \#(RLT) \tag{1}$$

Where #(X) is the number of queued tokens in scheme X.

However, from the system's point of view, more tokens in the queue means fewer tokens in operation, and there will be less throughput and higher average message delay in the network. Therefore, RLT scheme offers the least message delay. In order to perform a quantitative evaluation on the effects of these token-overlap handling schemes, we simulate the system using the same parameters that are used in [3] except that we put five tokens in circulation instead of one. The simulation results are plotted in Figure 3. It validates the previous arguments, and points out that the RLT is the best scheme for handling token-overlap in the MTB network.



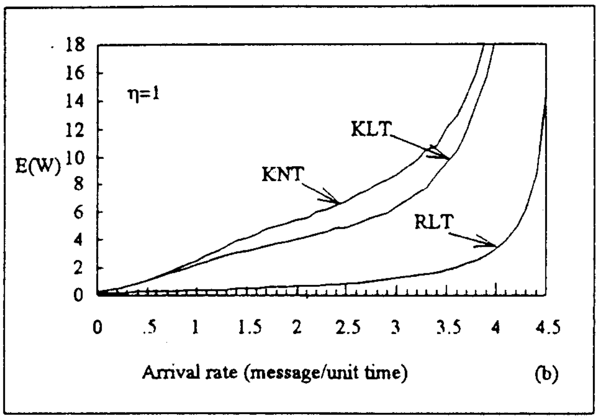


Figure 3. Average message delay E(W) and average token cycle time $\overline{c}$ versus the network arrival rate λ for three token overlap handling schemes in a network with 5 channels. (The unit time is 0.1 ms)

4 Average token cycle time

The response time of the network is directly affected by the token cycle time and is the most important performance index in a real-time system such as the multi-media network or the time-critical control system. In this section, the average token cycle time of the above strategies are analyzed and simulated for a symmetric MTB network consisting of N identical nodes.

4.1 Assumptions

1) There are M channels in the network, and each channel has its own independent token that operates according to the RLT scheme and the RRSD.
2) All channels possess the same characteristics.
3) The protocol used in each channel conforms to the IEEE 802.4 standard.
4) There are N Permanent nodes in the network, and every node use the same node model as shown in Figure 2.
5) The topology of MTB network is a fully connected network.
6) The arrival of messages to each node is an independent Poisson process with the same mean.
7) The time for serving a message or passing a token is an exponentially distributed random variable.
8) All nodes adopt the same THT settings ($200\mu s$ for a 10 Mbit/s bus).
9) $\overline{x}_p / \overline{x}_s$, the ratio of token processing time spent in PP to that spent in SP is identical for all nodes

10) The protocol management overhead is negligible.

4.2 Notations

λ $\equiv$ Message arrival rate of the network.
$\bar{c}$ $\equiv$ Average token cycle time.
$\bar{q}$ $\equiv$ Average number of tokens in the SS.
$\bar{w}$ $\equiv$ Average token-passing time.
$\bar{x}_p$ $\equiv$ Average message service time in the PP of a node.
$\bar{x}_s$ $\equiv$ Average message service time in the SP of a node.
$\bar{x}$ $\equiv$ Average message service time in a node $\equiv \bar{x}_p + \bar{x}_s$.
ρ $\equiv$ Utilization factor of each channel $\equiv (\lambda\bar{x})/M$.
$\gamma \equiv \bar{x}_p / \bar{x}_s$ $\equiv$ The ratio of service time in PP to that in SP in a node .
η $\equiv$ The maximum number of tokens permitted in a node. ($\therefore \bar{q} \le \eta$)
l $\equiv$ Average number of message arrivals to a node per channel in a token cycle.
$\bar{\zeta}$ $\equiv$ The mean of interval between a token's departure from a node and next token's arrival to the same node. If the token queue is not empty, this interval is zero.

4.3 Analysis

The token cycle time c is defined as the time interval between a token's departure from a particular node and the token's next departure from the same node. Under the previous assumption, each channel has its own token in the MTB network, therefore each channel will acquire its own $\bar{c}$ also.

In a symmetric MTB network, all channels will acquire the same $\bar{c}$ and the network saturation is reached when $THT \le l\cdot(\bar{x}+\bar{q}\cdot\bar{x}_s)$. Before the saturation, $\bar{c}$ consists of three components as follows.

$$\bar{c} = \sum_{i=1}^{N}\text{average token passing time} + \sum_{i=1}^{N}\text{average service time} + \sum_{i=1}^{N}\text{average token queueing time}$$
$$= \sum_{i=1}^{N}\bar{w} + \sum_{i=1}^{N}(l\cdot\bar{x}) + \sum_{i=1}^{N}(l\cdot\bar{q}\cdot\bar{x}_s) = N\bar{w} + N(l\cdot\bar{x}) + N\cdot(l\cdot\bar{q}\cdot\bar{x}_s). \qquad (2)$$

The first term is the sum of all average token-passing time $\bar{w}$ in a token cycle. The second term is the sum of average service time in all nodes. The average service time is made up of the average message quantity l that a token should serve in a cycle time and the average service rate $\bar{x}$. However, the $\bar{c}$ approaches saturation is not satisfied with this relation. The third component is the sum of average token queueing time in all nodes. The token queueing time is the time for a token waiting in the token queue. When a token arrives at a node, in average, it finds there are $\bar{q}$ tokens already in the SS. Hence, a token could work in the SP after waiting for $\bar{q}\,\bar{x}_s$ from its arrival. In addition, there is no contention after tokens leave the SP, because the number of PP's is equal to the number of tokens.

Because $\bar{x} = \bar{x}_s + \bar{x}_p$ and $\gamma = \bar{x}_p / \bar{x}_s$, then

$$\bar{x}_s = \bar{x} / (1+\gamma). \tag{3}$$

Substituting it to (2) yields

$$\bar{c} = N\bar{w} + N(l \cdot \bar{x})\left(1 + \frac{\bar{q}}{1+\gamma}\right). \tag{4}$$

Since the arrival rate to a node from each channel is λ / MN, the average number of arrivals to a node during a token cycle is

$$l = \frac{\lambda\bar{c}}{MN} \cdot \tag{5}$$

Substituting it to (2), we have

$$\bar{c} = N\bar{w} + N\left(\frac{\lambda\bar{c}}{MN} \cdot \bar{x}\right)\left(1 + \frac{\bar{q}}{1+\gamma}\right) = N\bar{w} + \left(\frac{\lambda\bar{c}}{M} \cdot \bar{x}\right)\left(1 + \frac{\bar{q}}{1+\gamma}\right) \cdot \tag{6}$$

In addition $\rho = (\lambda\bar{x}) / M$, the average token cycle can be expressed as

$$\begin{aligned} \bar{c} &= N\bar{w} + \rho \cdot \bar{c} \cdot \left(1 + \frac{\bar{q}}{1+\gamma}\right) \\ &= \frac{N\bar{w}}{1 - \rho \cdot \left(1 + \frac{\bar{q}}{1+\gamma}\right)}. \end{aligned} \tag{7}$$

From Eq. (7), it is apparent that $\bar{c}$ can be reduced either by reducing $\bar{q}$ or by increasing γ. γ is a system parameter which is the ratio of token processing time spent in the PP to that spent in the SP. When we change the network from single channel to multichannel, if we parallelize more operation in a node, we will have a higher γ and hence a shorter $\bar{c}$, but at a higher hardware cost. Conversely, if we keep most of the single channel hardware and process most operation serially, γ will be smaller and $\bar{c}$ becomes longer. The parameter $\bar{q}$ is affected by the processing time of SP token and token's arrival rate for the node. Its precise expression is difficult to derive, hence the numerical approximation of $\bar{q}$ is obtained by simulation and is discussed in the next section.

The service discipline of token bus network is nonpreemptive. In other words, a node is in the midst of transmitting a message when the THT expires, the message transmission is not interrupted. Meanwhile, the time for the token in service and in token queue will be $THT + \bar{x}$. Hence, the average token cycle time can be written as follows:

$$\bar{c} = N\bar{w} + N(THT + \bar{x}), \text{ where } THT \geq l \cdot (\bar{x} + \bar{q} \cdot \bar{x}_s) \tag{8}$$

4.4 Simulation

In order to verify the analytic results, a symmetric MTB network is simulated using CACI SIMSCRIPT II.5. The network parameters are the same as in [3]. They are:

Bus capacity = 10 Mbits/s
Number of network nodes N = 10
Number of network channels M = 5
Token holding time THT = 200 μs
Average message service time of node $\bar{x}$ = 0.1 ms
Average token-passing time $\bar{w}$ = 10 μs.

In order to validate the simulation, a special case with M=1 was first executed. Its results are then compared with the values published in [3]; which show good agreement. Afterward, the number of tokens is increased to five, and different values of η and γ are supplied for each simulation. Statistics are obtained from at least 20000 transmitted messages, and they fall in 95% confidence interval.

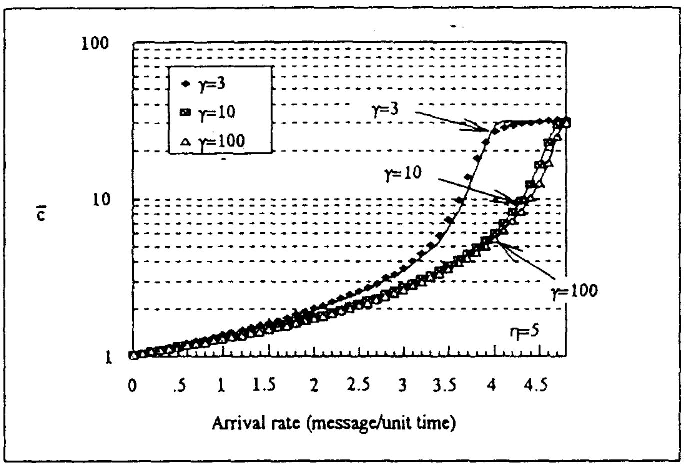


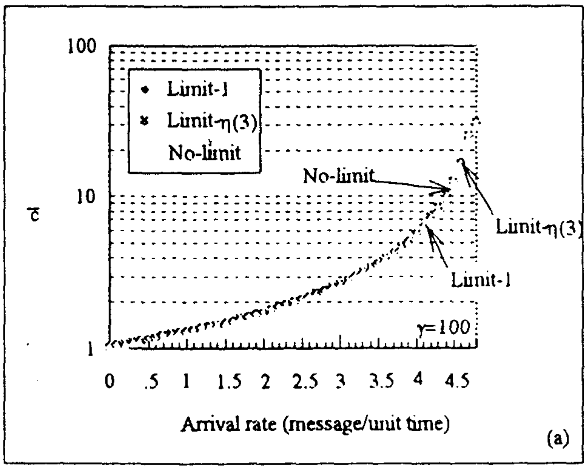
Figure 4. The average token cycle time $\bar{C}$ is under the influence of γ.

These results and the calculated values from (7) are shown in Figure 4 and Figure 5. In these figures, the simulation results are represented by the symbols, the calculated values are represented by the curves. Their differences are small.

In Figure 4, $\bar{c}$ curves in the no-limit strategy are plotted for γ equaling to 3, 10 and 100. The increasing γ does produce a decrease in $\bar{c}$ as expected, but the difference between curves is barely visible when the ratio γ is very large. The similar results are obtained for a network that uses limit-1 and limit-η strategies. By setting η to 1, 3 and 5, different $\bar{c}$ curves are obtained. These curves almost overlap together when γ=100 as shown in Figure 5-(a). This means that the token holding strategy is not an important factor for a system where a large part of the multi-token operations in a node is parallelized. With γ=3, a set of $\bar{c}$ curves for different token holding strategy is plotted in Figure 5-(b). In this figure, we can see that the $\bar{c}$ is reduced by decreasing the value of η, and the limit-1 strategy has an obvious improvement over others. In the discussion below, the γ=3 is adopted for studying $\bar{c}$ under the influence of token holding strategy, and the η=5 is adopted for studying $\bar{c}$ under the influence of γ.

The parameter $\bar{q}$ is a mean value obtained from the simulation, and its distributions are shown in Figure 6 and Figure 7. By Little's Law, $\bar{q}$ will be higher whenever the token processing time in the SP increases or the token arrival rate at a node increases. Moreover, the token processing time is affected by the variation of $l\bar{x}_s$ or $l \cdot \bar{x}/(\gamma+1)$, and the token arrival rate is affected by the variation of $\eta / \bar{c}$.

In light load situation, there are only few nodes carrying message to transmit. Hence, tokens will cluster in these few busy nodes and $\bar{q}$ will be higher. Increasing load will cause $\bar{c}$ to increase and therefore to reduce the token arrival rate at a node. It is the major factor that makes $\bar{q}$ to descend in these figures. As load grows, the token processing time in the SP grows too.



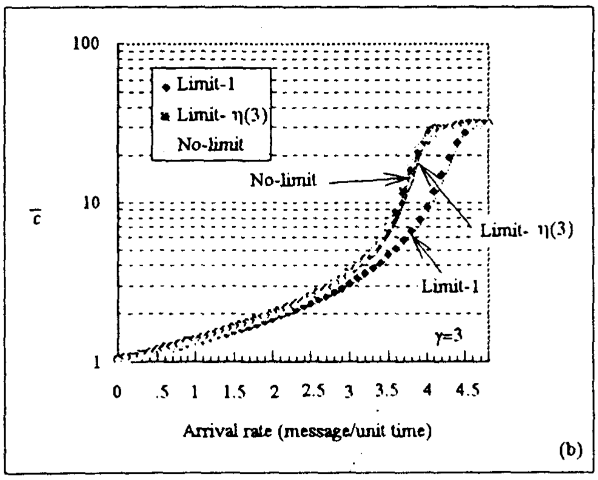


Figure 5. The average token cycle time $\bar{c}$ vs. the network arrival rate for different token holding strategies.

Since the increase in the token processing time in the SP and the decrease in the token arrival rate at a node will have conflicting effects on $\overline{q}$, therefore a peak is shown in the γ=3 curve (Figure 6) when the arrival rate is about 3.8. In neighborhood of the peak, the slop of $\overline{c}$ has the greatest value, and the token processing time in the SP has more influence over $\overline{q}$ than the token arrival rate at a node does. However, this phenomenon disappears when γ becomes larger. Figure 6 illustrates this phenomenon.

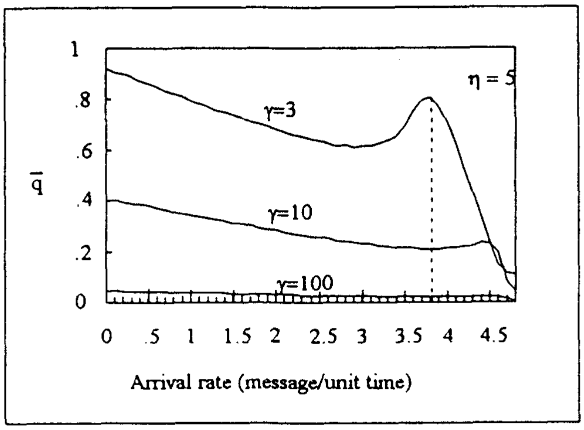


Figure 6. The average number of tokens $\overline{q}$ in a SS is under the influence of γ.

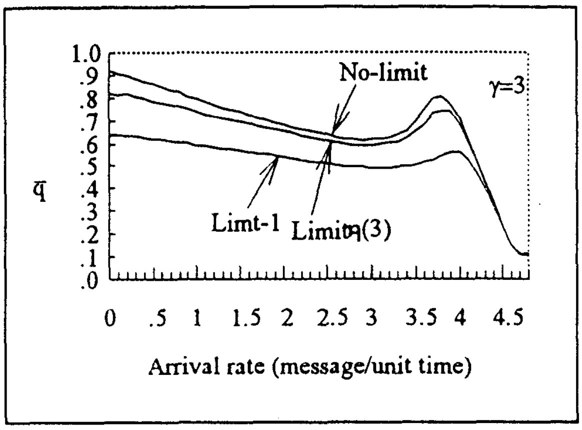


Figure 7. The average number of tokens $\overline{q}$ in a SS is under different η.

The variation of $\overline{q}$ under different η setting is shown in Figure 7. As one can see from this figure, a smaller η has a smaller token arrival rate at a node and therefore a smaller $\overline{q}$. From Figure 6 and Figure 7, it is obvious that γ has a greater impact on $\overline{q}$ than η does. In summary, the limit-1 strategy is the best strategy for the symmetric MTB network to handle the simultaneously arriving tokens.

5 Average message delay

Many research have been done to study the average message delay of the token-passing network because it directly affects the quality of service. They usually modeled the network as a M/G/1 polling system with polling overhead. Based on the same concept, the MTB network can be directly modeled as an M/G/m polling system with polling overhead. However, the formula for the average message delay in M/G/m polling system with polling overhead is too complicated and is never derived before. Here, simulation is used to study the average message delay. The simulation results are shown as in Figure 8 and Figure 9.

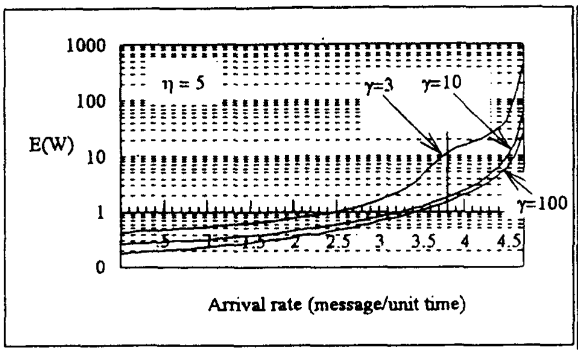


Figure 8. The average message delay E(W) under the influence of γ.

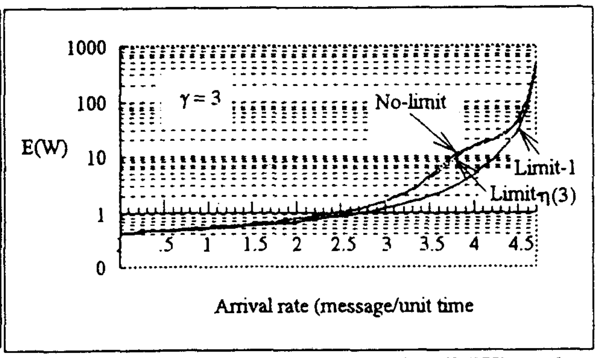


Figure 9. The average message delay E(W) under different setting of η.

In a MTB network, an average message delay could be divided into three components as follows:

$$E(W) = \bar{x} + \sum_{i=1}^{k} \overline{x_s} + \sum_{i=1}^{p} \bar{\zeta} \tag{9}$$

The first is the processing time of the message in the SP and in the PP. The second is the sum of all SP processing time of the k messages that arrived before. Since the THT limits the number of messages served by each token, a message will see p tokens and be served by the pth token. The third is the sum of $\bar{\zeta}$ of these p tokens. Generally, more tokens circulating around the network means less $\bar{\zeta}$. Using the value of $\bar{x}$ from (3), we obtain

$$E(W) = (1+\gamma)\bar{x}_s + \sum_{i=1}^{k} \bar{x}_s + \sum_{i=1}^{p} \bar{\zeta} = (k+\gamma+1)\bar{x}_s + p \cdot \bar{\zeta} \tag{10}$$

Figure 8 shows the average message delay *E(W)* versus arrival rate for different γ values. It is apparent that *E(W)* can be reduced by increasing γ. This result agrees with expression (10). However the distance between two neighboring curves of *E(W)* is small as γ becomes large.

Figure 7 shows that a larger η causes more tokens in the token queue, hence there will be fewer tokens circulating around the network. It will cause an increase in the $\bar{\zeta}$ too. By expression (10), the average message delay will consequently increase also. In Figure 9, three *E(W)* curves are plotted for three different token holding strategies. It shows that *E(W)* increases with η. In summary, the limit-1 strategy is still the best strategy for the average message delay in a symmetric MTB network.

6 Conclusion

Expanding the single channel network to the multichannel network is a practical solution to increase network bandwidth. Nevertheless, the token handling policy of multichannel network has its own problems to be studied. In this paper, several strategies are proposed to handle the token holding problem when multiple tokens simultaneously arrive at a node. In order to study these strategies, the average token cycle time is analyzed and simulated under different degree of parallel processing and different token allowance in a node. From the numerical results, we have found that the limit-1 strategy, where only one token is allowed to operate in a node, is the best of these strategies in a symmetric MTB network. In addition, the difference of these strategies is especially noticeable when the degree of parallel processing is low. However, it is expensive to have a very large γ and a compromise should be selected in design time. Hence, this investigation does give some help to design a MTB network. Besides, the result of this paper can also be applied to the multiple token ring and the multichannel FDDI network.

Reference

1. M. A. Marsan and D. Roffinella, "Multichannel Local Area Network Protocols," *IEEE J. Select. Area. Commun.*, Vol. SAC-1, NO. 5, 885 (1983).
2. Wong and Benny C. Y., "Performance evaluation of a multichannel token bus local area network architecture," *IEEE Pacific RIM conf.*, 226 (1989).
3. O. C. Yue and C. A. Brooks, "Performance of the Timed Token Scheme in MAP," *IEEE Trans. Commun.*, vol. 38, no. 7, 1006 (1990).
4. H. H. Sung and L. Y. Kung, "Multichannel Manufacturing Automation Protocol Network," *Joint Tech. Conf. comput. and commun.*, 493 (1989).
5. W. S. Hwang and L. Y. Kung, "Analyze Gain of Distributed Multichannel Manufacturing Automation Protocol Network," *IEEE Singapore ICCS/ISITA conf. proc.*, 512 (1992).
6. H. D. Friedman, "Reduction Methods for Tandom Queuing System," *Oper. Res.* 13(1), 121 (1965).