中華民國八十六年

全國計算機會議論文集

第三冊

Proceedings of
National Computer Symposium 1997
Republic of China

Volume 3

教育部主辦
東海大學承辦

Session F.5: 電腦網路(一)

Session F.6: 電腦網路(二)

Session F.7: 電腦網路(三)

多頻道工廠自動化協定網路

Multichannel Manufacturing Automation Protocol (M^{2}AP) Network

黃文祥* 孔令洋** 王俊堯**
Wen-Shyang Hwang, Ling-Yang Kung, Jun-Yao Wang
* 國立高雄科學技術學院電機系
Department of Electrical Engineering, National Kaohsiung Institute of Technology.
** 國立成功大學電機工程研究所
Department of Electrical Engineering, National Cheng Kung University.

摘要

本文探討了多頻道工廠自動化協定網路及其引發的問題，如相容性、多權杖傳遞、硬體複雜性及頻道擴充彈性等問題。

關鍵字: 多頻道，權杖傳遞

Abstract

This paper discusses the multichannel implementation of MAP and its related problems such as compatibility with the original MAP network, multi-token handling, hardware complexity, channel expansion capability, etc. Simulation result shows that it offers near-linear improvement with the increase of channels.

Indexing terms: Multichannel, token-passing.

1. Introduction

Manufacturing Automation Protocol (MAP) is a broadband token passing protocol. It provides a set of rules to control the token holding time in each station in order to put a bound on the maximum token rotation cycle. Thus a high priority message can be served within a time limit that is essential in real-time application. In addition, MAP is compatible to the **Cable Television** (CATV) system. Therefore multimedia capability can be added easily.

After the **National Information Infrastructure** (NII) project was announced, there is a perspective to integrate **Computer, Communication and Entertainment** (CCE) into the same backbone. Although budget can be allocated to install new high speed cables reaching every home, using the existing CATV system as backbone is likely to be the best choice, especially in those countries where more than half of the households are already wired with broadband cable. With MAP running on the CATV backbone, it offers a ready solution to the CCE integration. Besides, the controlled response time characteristic of MAP will find its application in iterative games that the current CATV companies are offering with a more expensive method.

A **Multichannel Local Area Network** protocol (M_LAN) was first published by Marsan and Roffinella in 1983 [3]. Their multichannel CSMA network consisted of a set of parallel broadcast channels. Although the delay time in the M_LAN was less than that in a single channel CSMA network, its response time could not be guaranteed for real-time application. This paper presents a **Multichannel MAP** (M^{2}AP) network that is based on the MAP protocol to provide a larger bandwidth and a faster response with controlled delay. To make the M^{2}AP network compatible with the existing MAP network, M^{2}AP should comply with all the regulations in MAP, such as distributed control protocol, deterministic token cycle time, identical frame-format and so on.

MAP used in real-time application has been studied by Jayasumana in 1989 [6] and by Park, Ahn, and Kwon in 1993 [10]. The latter proposed an algorithm for tuning the MAP parameter to satisfy the real-time requirement. However, this result is applicable only to the single channel MAP network. Here, we will give a new formula to calculate the token cycle time in M^{2}AP and use it to derive an exact expression of maximum allowable arrival rate to keep the network in stable operating region.

There were many research interests in the performance of MAP network [5][7][10]. They usually modeled the network as a M/G/1 polling system with polling overhead. The expressions in average token cycle time and message delay for the polling system were derived by Fuhrmann in 1985 [4], then by Bertsekas and Gallager in 1987 [2]. Based on these results, the performance of M^{2}AP network will be studied in this paper. Because there are many tokens (or servers) independently circulating around the M^{2}AP network, it should be modeled as an M/G/n system with vacation[2].

Section 2 discusses the M^{2}AP model. Section 3 depicts two M^{2}AP systems. The first one is called M^{3}AP (M transmitters distributed Multichannel Manufacturing Automation Protocol) where each station has as many transmitters as the number of channels in the network. The second one is called D^{2}M^{2}AP (Dual transmitters Distributed Multichannel Manufacturing Automation Protocol). A D^2M^2AP station uses only two transmitters regardless the number of channels. The functional diagram of each system and their operations are also illustrated. Section 4 describes the problem of multiple tokens coming into a station and presents three schemes to solve it. Section 5 analyzes the performance of these networks and gives some numerical results for each network. Finally, a conclusion is given in section 6.

2. M^2AP models

Although the physical topology in the M^2AP network is still a bus, the logical topology consists of many independent rings as shown in Figure 1. Each ring in the network has an exclusive token, and every station has an independent **Medium Access Control** (MAC) layer for each ring to carry out the ring maintenance. Since there is more than one data flow path between **Logical Link Control** (LLC) and MAC, a new model for M^2AP is necessary.

The lower layer services of MAP network are provided by IEEE standard 802.4, but the IEEE 802.4 model is designed for a single channel network. To make it operate for the multichannel system, the MAC layer in IEEE 802.4 model is broken into one **Upper MAC** (UMAC) sublayer and many **Lower MAC** (LMAC) sublayers as shown in Figure 2. The function of the LMAC sublayer is to implement the protocol of the original MAC layer in addition to the MAC-LLC interface. The function of UMAC sublayer is to manage all contentions produced by the LMAC sublayers, and to carry out the MAC-LLC interface protocol according to the IEEE 802.4 model.

In the original model of IEEE standard 802.4, the path between LLC and transmission medium is a single path. Here, because there are more paths in the LMAC sublayer, a one-to-many and a many-to-one **scheduling problems** will arise. They will be discussed below in detail. Besides these extra scheduling problems, M^2AP operations comply with all rules in IEEE standard 802.4. Hence, the rest of this paper will focus on how to handle the scheduling problems and on how to analyze the system response affected by these handling schemes.

A. UMAC-LMAC Interface.

- *Data frame.*

As in MAP, M^2AP divides data frames into four classes (6, 4, 2 and 0) according to their priorities in LLC layer, and stores them in the corresponding priority queue before transmission and after reception. The received data frames from different channels enter the received queue (RQ). Conversely, data frames are retrieved from the transmitting queue (TQ) only when UMAC holds a token.

- *Token frame.*

Token is passed from station to station around the logical ring, and only the token holder is permitted to transmit data frames. In a M^2AP network, although there are many tokens circulating in the network, each station is allowed to use only one token at a time, and we call it **Reserve One Operating Token Strategy** (ROOTS).

B. Physical-Transmission medium Interface.

- *Receiver.*

In a MAP network, each receiver accepts data frames from remaining N-1 stations. The ring status has to be monitored continuously by all stations for ring maintenance. Hence, the number of receivers in a M^2AP station should be equal to the number of rings it is connected to.

- *Transmitter.*

Theoretically, the minimum number of transmitters needed to operate is one. In reality this is not the case, because in the MAP network there is a special kind of frame, called the MAC frame, to be transmitted in addition to the data frame. These MAC frames are used primarily for ring maintenance, and most of them have a response time constraint. According to the MAP protocol, the maximum data frame size is 8191 octets which is much longer than the MAC frame size of 14 octets. Therefore, some MAC frames will possibly lose their validity when they queue behind data frames. Because of the above consideration, we use a dedicated transmitter to handle MAC frames and so the minimum number of transmitters in a station is two.

3. M^3AP and D^2M^2AP

The function block diagram of M^3AP and D^2M^2AP are shown in Figure 3. Except in some new modules, the operation is the same as in MAP. The operation of the new modules is described below.

A. M^3AP

For arbitrating the token contention problem, **Data Frame Manager** (DFM) is added to each station. The MAC module of MAP is replaced by a DFM and many LMACs, as shown in Figure 3-a. The DFM operations are divided into three parts as follows:

(1) Managing received frames

Data frames from different LMACs are stored in a DFM buffer. DFM passes them to the **Received Queues** (RQs) according to their frame priority. If a MAC frame arrives, it is processed according to the MAP regulation and the required response message is sent back through the same channel.

(2) Managing data frames transmission

Since each channel in M^3AP has a private transmitter, there is no transmission conflict. Once DFM obtains a token, it removes data frames from **Transmitting Queues** (TQs) according to the priority mechanism of MAP, then sends them to the channel to which the operating token belongs.

(3) **Handling token-overlap**

After each transmission, DFM will process the **token queue** (TKQ) according to the specified token-overlap handling scheme that will be discussed in next section. Tokens are released through their own channel.

B. D^2M^2AP

Each station on D^2M^2AP has only two transmitters: a **MAC Transmitter** (MTX) for transmitting MAC frames and a **Data Transmitter** (DTX) for sending data frames. Each transmitter also has a channel selection capability. This selection is carried out by using a **programmable carrier generator** which generates a specific frequency carrier to modulate the outgoing frames into the specific channel. For arbitrating simultaneous frame requests, DFM and **MAC Frame Manager** (MFM) are added to a D^2M^2AP station to schedule the transmission of data and MAC frames as shown in Figure 3-b.

The receiver's operation in D^2M^2AP is similar to the operation in M^2AP. When a frame arrives, the LMAC processes this frame first. Afterwards, the processed frame is sent to MFM for an identification check. If it is a data frame, then this frame will be passed to DFM. DFM will store this data frame in one of the received queues according to its priority. If it is a token frame the token will be added to the TKQ. Then DFM will process the TKQ as in M^2AP and send back the unwanted tokens to MFM. Later on, they will be released to their own channel.

MFM may simultaneously receive many MAC frames from different rings, but MTX will transmit only one MAC frame at a time. MFM will assign a priority to each MAC frame according to its urgency and insert them in a priority queue called **Weighted Queue** (WQ). After a MAC frame transmission the priority of each frame in WQ will be upgraded by one level. With this mechanism, all MAC frames can be sent on time even when the traffic load is heavy [8].

4. Token-overlap handling schemes

In a symmetric M^2AP network, the setting of **token holding time** (THT) for each station is identical, and each token is identical too. The acquired THT of a station is proportional to the number of tokens. However, since there are M tokens circulating in the network, it is possible that more than one token arrives simultaneously at a station. This phenomenon is called "**token-overlap**," and each of these tokens is given a name "**overlap token.**"

According to ROOTS, there is only one token that can operate in a station. If token-overlap happens, there are three possible schemes to select the operating token. They are:

A. Keep Next Token (KNT):

All arrived tokens are first put into a TKQ. These tokens are called **queued tokens**. After every message transmission, the TKQ is checked. If it is empty, the operating token will continue to operate until its THT expires; otherwise, the operating token is released and the token at the head of TKQ becomes the next operating token. Hence, every token in the TKQ can transmit at least one message if any.

B. Keep Last Token (KLT):

Queued tokens are idle tokens. The non-idle tokens are called **active tokens**. They include operating tokens and walking (passing) tokens. In order to reduce idle tokens the KLT scheme is proposed. The operation of this scheme is similar to KNT except that when TKQ is checked and is found not empty, the station will release all but the last token in TKQ. Therefore, queued tokens except for the last one may not transmit any message.

C. Reject Later Token (RLT):

When a station already possesses one token, all later tokens will be rejected and will be bypassed to the next station immediately. Hence, all later tokens offer no service to a busy station.

From the station's point of view, the KNT scheme provides the most THT to a station and holds more tokens in the TKQ than the others. On the contrary, the RLT scheme does not hold any token in the TKQ. Hence, the relations for the number of queued tokens are:

$$\#(KNT)>\#(KLT)>\#(RLT) \qquad (1)$$

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

From the system's point of view, more tokens in the queue mean fewer tokens in operation. Therefore, the relation for the number of active tokens ζ in the network is

$$\zeta(KNT)<\zeta(KLT)<\zeta(RLT). \qquad (2)$$

In order to have a quantitative evaluation on the effects of these token-overlap handling schemes, we simulate the system using the same parameters that are used in [7] except that we put five tokens in circulation instead of one. The simulation results are plotted in Figure 4. As we can see from this figure, since there is no queued token in RLT, the average message delay with this scheme should present a curve whose shape is similar to that of a single server polling system. As far as KNT and KLT are concerned, the number of queued tokens with these schemes decreases when the traffic load increases. Moreover, the data transmitting time for a station is generally much longer than the token passing time. Hence, all tokens are crowded in one station when the traffic is light; it looks as if there is only one active token moving around. With the increase of the arrival rate, the number of token holders will gradually increase, and the probability of token-overlap will decrease.

The simulation result validates equation (2), and points out that RLT is the best scheme for handling token-overlap in the M^{2}AP network.

5. Average token cycle time and Stability condition

To understand the improvement that M^{3}AP and D^{2}M^{2}AP can achieve, the average token cycle time of both networks is discussed in this section. The parameter setting for real- time requirement and the stable operating region are also presented. Here are the assumptions and notations used in our model:

A. Assumptions

1) There are M channels in both networks and each channel has its own independent token.
2) There are N permanent stations in these networks, and each station is connected to all channels.
3) The arrival of messages to each station follows an independent Poisson process.
4) All stations have the same traffic load and token holding time settings.
5) Though the token holding timer expires, the transmission of the current message will be completed, and an excess token holding time is consequently made.
6) Only average message delay of high priority message is considered.
7) The logical ring management overhead is negligible.
8) The RLT scheme is adopted in both networks to handle the token-overlap.

B. Notations

T_h ≡ Time limit that a station can hold a token per visit.
T_x ≡ Maximum value of the excess token holding time.
Tm ≡ Mean value of the excess token holding time.
W_x ≡ Maximum token passing (walking) time.
$\overline{w}$ ≡ Mean token passing (walking) time.
$\overline{v}$ ≡ Mean token visit time of a station.
l ≡ Average number of arrivals at a station per channel in a token cycle.
λ ≡ Message arrival rate of network.
$\overline{x}$ ≡ Mean service time.
ρ ≡ Utilization factor of each channel ≡ $(\lambda\overline{x})/M$.
$\overline{q}$ ≡ Average number of tokens in the WQ in a D^{2}M^{2}AP station.

C. Analysis

The token cycle time c is defined as the time interval between a token's departure from a particular station and the token's next departure from the same station. Its mean value is $\overline{c}$. The average cycle time of each token in the M^{3}AP network is:

$$\overline{c} = N\overline{w} + N\overline{v} = N\overline{w} + N(l \cdot \overline{x}) \tag{3}$$

By Little's law the number of system arrivals can be calculated as $\lambda\overline{c}$. Since the network is symmetric, the number of arrivals at each station in each channel can be obtained by dividing the total system arrival by MN. Hence the average number of arrivals l can be expressed as

$$l = \frac{\lambda\overline{c}}{MN} \tag{4}$$

Substituting for l in (3), we obtain

$$\overline{c} = \frac{N\overline{w}}{1-(\lambda\overline{x})/M} = \frac{N\overline{w}}{1-\rho} \quad (\text{M}^3\text{AP}). \tag{5}$$

Similar result was shown in [5].

In D^{2}M^{2}AP because there is only one transmitter that handles the token passing of all channels, every token will be held until all tokens that stay ahead of it are processed. Hence, the average token cycle time of D^{2}M^{2}AP is:

$$\overline{c} = N\overline{w} + N\overline{v} + N\overline{q}\,\overline{w} = N\overline{w}(1+\overline{q}) + N(l \cdot \overline{x}) \tag{6}$$

Substituting for l in (6), the average token cycle time can be rewritten as

$$\overline{c} = N\overline{w}(1+\overline{q}) + \frac{\lambda\overline{x}}{M} \cdot \overline{c} = \frac{N\overline{w}(1+\overline{q})}{1-\rho} \quad (\text{D}^2\text{M}^2\text{AP}). \tag{7}$$

The same result can be achieved by lengthening the token passing time $\overline{w}$ in M^{3}AP to $(1+\overline{q})\overline{w}$. However, these formulae are correct only before the network saturation point. While the network load approaches to saturation, the sum of average excess service time in a station is equal to $T_h + Tm$, hence these average token cycle times become:

$$\begin{aligned} \overline{c} &= N\overline{w} + N \cdot (T_h + Tm) && (\text{M}^3\text{AP}), \\ \overline{c} &= N\overline{w} \cdot (1+\overline{q}) + N \cdot (T_h + Tm) && (\text{D}^2\text{M}^2\text{AP}). \end{aligned} \tag{8}$$

D. Simulation

To verify the formulae of $\overline{c}$ before saturation, simulations were conducted for both M^{3}AP and D^{2}M^{2}AP networks. Furthermore, for validating these simulation programs, the simulation results for one token are compared with the values obtained from [2][4][7]. They show good agreement. The service time and the token passing time in these simulations were assumed to be exponentially distributed as in [7]. However, other distributions are tried, and their results converge to the same value. Then we set N=10, M=1 to 5, $\overline{x}$ =100 *us*, $\overline{w}$ =10 *us*, and T_h = 200 *us*. All simulation results were obtained over at least 20000 transmitted messages before computing the mean delay, and each sample mean was obtained from 30 different batches. The method of batch means in [11] is adopted for computing confidence interval for these results, and the outcome shows that they all fall in the 95% confidence interval.

The simulation results and the calculated values from (7) are shown in Figure 5 where the value of $\bar{c}$ is normalized with respect to $\bar{x}$. In this figure, the simulation results are represented by the dot (symbol) lines, and the calculated values are represented by solid lines. As it can be seen, their differences are small in both networks. In Figure 5-b, if we select a constant value of $\bar{c}$ we can estimate that the system arrival rate has a nearly linear growth with the number of channels. Figure 6 shows the average number of queuing tokens $\bar{q}$ in a station. They are statistically collected from the simulations. Since the number of overlap tokens decreases when traffic increases, the curve of $\bar{q}$ descends gradually as the arrival rate increases.

Although, the relative overhead in D^{2}M^{2}AP is higher than that in M^{3}AP when the network load (or λ) is light, however, the heavy load performance is more important because the main objective of this multichannel network is to provide a larger bandwidth for the crowded traffic. According to the simulation results shown in Fig. 5, the D^{2}M^{2}AP network is as good as the M^{3}AP network, though the former uses only two transmitters. Besides, D^{2}M^{2}AP makes the network expansion easier because it only requires an addition of receivers, hence the D^{2}M^{2}AP network is indeed a practical solution for the M^{2}AP network.

6. CONCLUSION

MAP is compatible to the CATV system, hence multimedia capability can be added easily. Moreover, its high priority message can be served within a time limit that is essential in real-time application. In this paper, two approaches to implement the multichannel MAP network were presented for increasing the network bandwidth. They are the M^{3}AP (M transmitters distributed Multichannel Manufacturing Automation Protocol) and the D^{2}M^{2}AP (Dual transmitters Distributed Multichannel Manufacturing Automation Protocol). Both approaches require only slight modifications of the MAC layer and produce a nearly linear grow of throughput as the number of channels increases. We believe that the number of receivers should be equal to the number of channels because the status of each channel should be continuously monitored, but the number of transmitters can be reduced to two: one for handling the data frame transmission, and the other for the MAC frames. With such an arrangement, simulation results show that the difference in their average message delay is small. Moreover, adding more channels to a D^{2}M^{2}AP station requires only the addition of receiver modules. Furthermore, it can be implemented practically by putting many receivers in an Integrated Circuit (IC) chip. Therefore the D^{2}M^{2}AP network offers greater expandability than the M^{3}AP network.

Reference:

[1] IEEE computer society, *ANSI/IEEE standards for token-passing bus access method and physical layer specifications*, 1985.

[2] D. Bertsekas and R. Gallager, *Data Networks*, Prentice-Hall, 1987.

[3] M. A. Marsan and D. Roffinella, "Multichannel Local Area Network Protocols," IEEE J. Select. Area. Commun., Nov. 1983, Vol. SAC-1, NO. 5, PP. 885-897.

[4] S. W. Fuhrmann, "Symmetric Queues Served in Cyclic Order," Oper. Res. Lett., Oct. 1985, vol. 4, no. 3, pp. 139-144.

[5] P. J. Kuehn, "Multiqueue Systems with Nonexhaustive Cyclic Service," Bell sys. Tech. J., Mar. 1979, vol. 58, no. 3, pp.671-698.

[6] A. P. Jayasumana and G. G. Jayasumana, "On the Use of the IEEE 802.4 Token Bus in Distributed Real-Time Control Systems," IEEE Trans. ind. Electron., Aug. 1989, vol.36, no.3, pp.391-397.

[7] O. C. Yue and C. A. Brooks, "Performance of the Timed Token Scheme in MAP," IEEE Trans. Commun., Jul. 1990, vol. 38, no. 7, pp. 1006-1012.

[8] W. S. Hwang and L. Y. Kung, "Distributed Multichannel Manufacturing Automation Protocol Network," IEEE Singapore ICCS/ISITA conf. proc., 1992, vol. 2, pp. 507-511.

[9] W. S. Hwang and L. Y. Kung, "Analyze Gain of Distributed Multichannel Manufacturing Automation Protocol Network," IEEE Singapore ICCS/ISITA conf. proc., 1992, vol. 2, pp. 512-516.

[10] H. S. Park, S. C. Ahn and W. H. Kwon, "Performance and Parameter Region for Real-Time Use in IEEE 802.4 Token Bus Network," IEEE Trans. Ind. Electron., Aug. 1993, vol. 40, no.4, pp. 412-419.

[11] A. Leon-Garcia, *Probability and Random Processes for Electrical Engineering*, 1989.

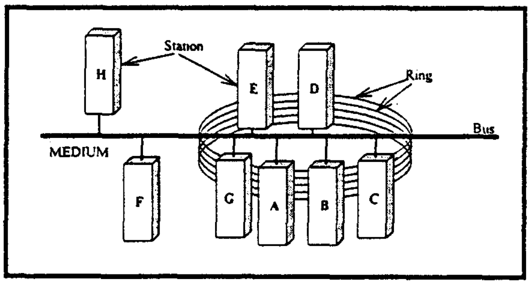

Figure 1. *Physically, M^{2}AP network is a bus topology onto which the stations are attached. Logically, all stations are organized into many rings.*

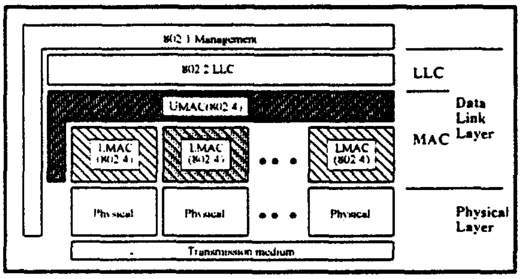

Figure 2. *The new model for* M^2AP *network.*

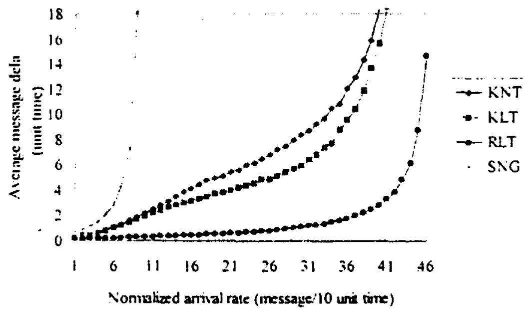

Figure 3. *Average message delays versus system load for the three token-overlap handling strategies in a network with 5 channels. (SNG presents the results of a single token in a network)*

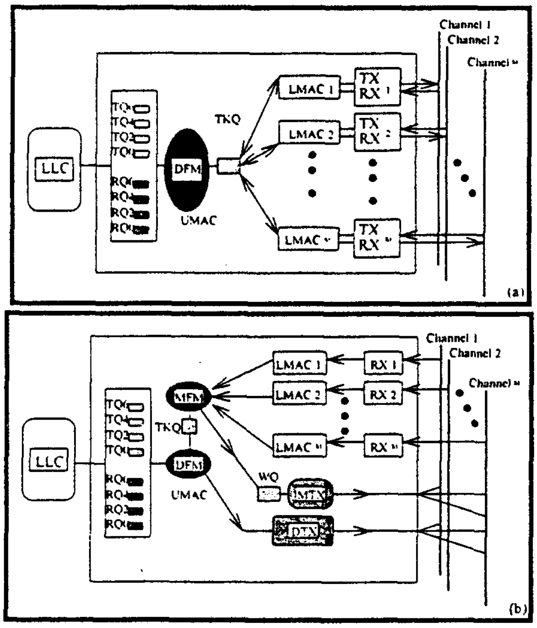

Figure 3. *The function diagram of (a)* M^3AP *and (b)* D^2M^2AP *network.*

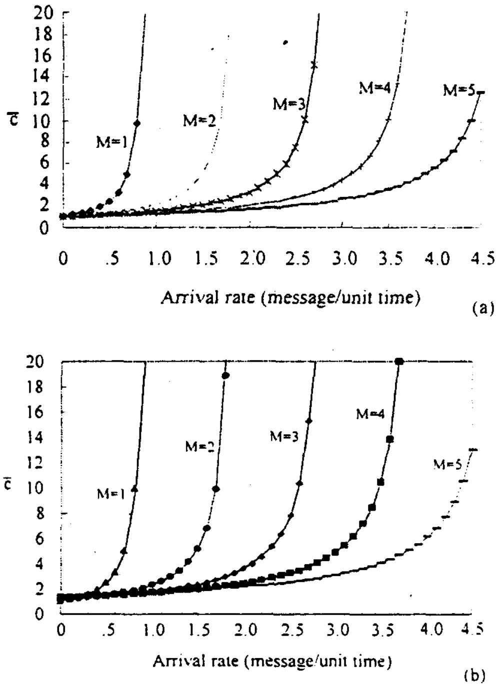

Figure 5. *The average normalized token cycle time of (a)* M^3AP *and (b)* D^2M^2AP *networks.*

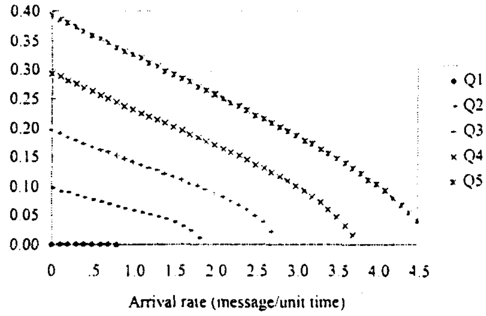

Figure 6. *Average number of tokens in token queue for* D^2M^2AP *network with M channels (QM).*