中華民國八十二年

全國計算機會議論文集

第　二　冊

Proceedings of National Computer Symposium 1993 Republic of China

Volume II

主辦單位：國 立 中 正 大 學
協辦單位：教 育 部 電算中心
中華民國 電腦學會

分散式多頻道工廠自動化協定網路的權杖迴旋目標時間
TARGET ROTATION TIMES IN DMMAP NETWORK

黃文祥　　孔令洋

Wen-Shyang Hwang and Ling-Yang Kung

國立成功大學電機工程研究所計算機組

Research Institute of Electrical & Computer Engineering

National Cheng Kung University, Tainan, Taiwan, R.O.C.

摘要

分散式多頻道工廠自動化協定網路爲高速區域網路，它的媒體存取控制方式(MAC)與IEEE 802.4的MAC是很相似。由於多頻道的緣故，它將ISO製定的OSI中MAC層再分割成一層UMAC及一層含數個LMAC，使得DMMAP實體上看起來是一條通訊線路，但邏輯上是同時存在的多個獨立環狀的網路。它的傳輸頻寬可因增加LMAC的數量隨之增加。在動態指定的環境中，頻道的數量可多於LMAC的數目，並可以讓同一工作羣的工作站專用於一個頻道上，也可讓大量且即時性的資料獨佔一條專用頻道。由於DMMAP的架構只採用兩個傳送器，因此必須增長反應窗的設定，再加上多個權杖同時抵達一個工作站所造成權杖重疊效應，使得所提高頻寬與增加的LMAC不一定成比例。在 [2]中已提出這方面的分析，並計算出在一百個頻道以內，它有近似於線性的成長。

爲了使較緊急資料的等候時間較短，DMMAP採用了四個優先等級來對應於LLC的八個優先等級。本文將對 DMMAP網路的多權杖同時抵達一工作站的處理，提出兩種可行的解決方式，並分析其對頻寬的影響。然後再對 DMMAP網路的權杖迴旋目標時間作進一步的分析，以導出一套新公式讓使用者能正確的設定權杖迴旋目標時間 。

關鍵詞：權杖，權杖重疊，時槽，反應窗

Abstract

The Distributed Multichannel Manufacturing Automation Protocol network (DMMAP) is a high speed local area network [1], which medium access control (MAC) is analogous to IEEE 802.4 token bus. Its development is based on the MAP (Manufacturing Automation Protocol) network. Physically, the topology is a linear or tree-shaped cable onto which the stations are attached, as fig 1. Logically, the stations are organized into many independent rings network as fig 2. The DMMAP expanded network bandwidth is accompanied with

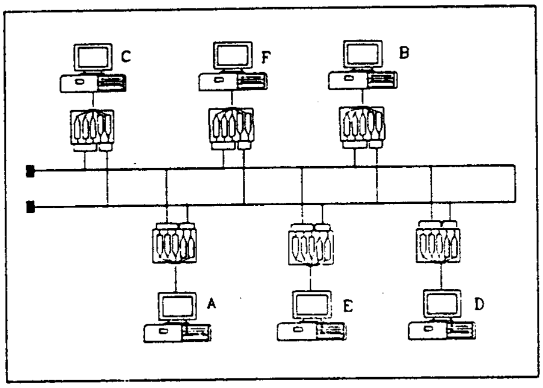

Figure 1. The physical topology of DMMAP network.

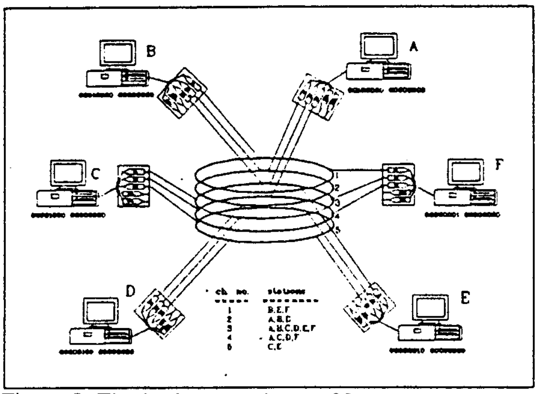

Figure 2. The logical topology of DMMAP network.

the growth channel number by increase its Lower MAC (LMAC). In the dynamic assignment environment, the channel number is not equal to the LMAC number. It allows the stations of same group can occupy a private channel, so the channel number will be great than LMAC number. Moreover, the extended bandwidth is not proportional to the channel number, since the overhead of the response window is accompanied by increased channel number. Therefore the extended bandwidth curve of service time versus channel

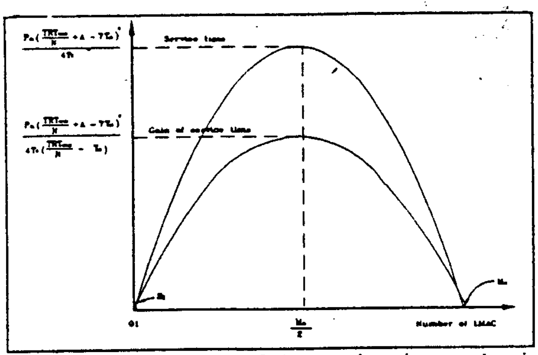

Figure 3. Comparison of the service time and gain versus the various LMAC number.

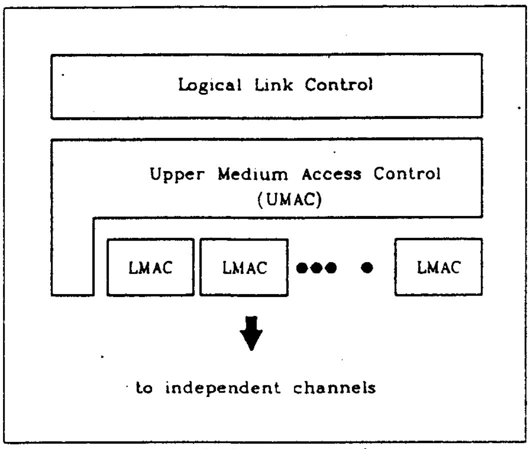

Figure 4. The DMMAP layer model.

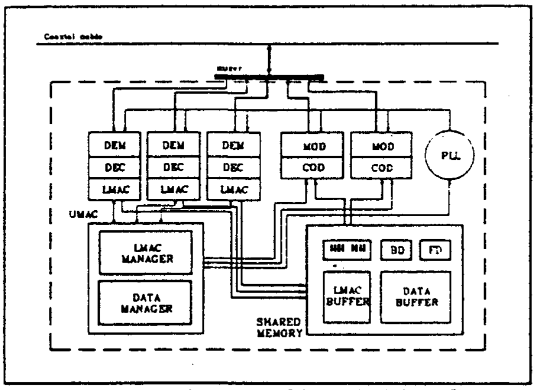

Figure 5. The architecture of DMMAP interface.

number of DMMAP is a parabola, as fig 3. In the paper [2], we found that relation is like linear growth when the channel number is little than one hundred.

In order to obtain short waiting time for urgent message, the DMMAP offers four level services called access classes (6,4,2, and 0 with 6 being the highest priority) that are equivalent to the access classes (7,6), (5,4), (3,2), and (1,0) in Logical link control (LLC). In 1988, Gorur and Weaver derive the formulae to transform the priority scheme in IEEE 802.4 token bus network into the proper Target Rotation Time (TRT) settings [3]. They call those access classes in order of descending priority, Synchronous, Urgent Asynchronous, Normal Asynchronous, and Time Available. In this paper, we will base the above scheme to derive the new formulae, that will transform the data priority scheme in DMMAP network into the proper TRT setting.

Keywords: Token, Token overlay, Slot time, Response window

Introduction

The IEEE 802.4 network uses highly reliable cable television equipment, which is available off-the-shelf from numerous vendors. It is more deterministic than 802.3, and has excellent throughput and efficiency at high load [4]. However advances in computing and other technologies have produced a significant and rapid increase in the demand for new communication services. The need of network bandwidth will enormously increase in the future. In this paper, we will introduce a high bandwidth network, DMMAP, and derive the new formulae to assist users setting the three TRT values. Section 1 surveys that network system architecture, its medium access control protocol, and its data frame control protocol. In addition, we built an analytic model and derive the TRT formulae of the lower three priority access classes in section 2.

I. System architecture.

The IEEE 802.4 standards does not directly correspond to the ISO/OSI model. IEEE 802.4 breaks the data link layer into the media access control (MAC) and the logical link control (LLC) sublayers [5]. In a multichannel network, each channel needs a private MAC device to service its medium access control, so that standard is not enough to model this kind network. Hence we break the MAC sublayer of IEEE 802.4 standard into an upper MAC (UMAC) and many Lower MAC (LMAC) sublayers, as fig 4. The former provides the scheduling data frames and MAC frames services according to those frame's priorities. The latter provides the MAC services that are similar to IEEE 802.4 MAC service. Each LMAC has to continuously monitor the channel to which it is attached, so the number of receiver should equalize with the number of LMAC. However the LMAC is not need to always occupy the transmitter, and the consideration bases on reducing data-flow, power consumption, magnetoelectric disturbance, heat, and cost point of view. We adopt only two transmitters that is called data frame transmitter (DTX) and MAC frame transmitter (CTX).

The network station architecture on the DMMAP with 3 channels, for example, should consist of an UMAC, 3 LMACs, 3 receivers, a shared memory, a CTX, and a DTX, as shown in fig 5. The channel of DMMAP is fully independent of each other, and everyone has a private token. Therefore all channels can operate simultaneously in this system. However there are many LMACs in a network station, they may need to send its MAC control frame to the UMAC at a same time, so the DMMAP should pay some bandwidth to resolve that contention overhead [2]. It will retard the total bandwidth growth. The UMAC sublayer is consist of data manager (DATAM) and LMAC manager (LMACM). The former is used to manage the data packets from or to LLC, those packets are assigned to transmitting or received queues according to their data priorities. The latter is used to arrange the simultaneous arrival of MAC control frames which from LMACs to the CTX. It will make the urgent MAC control frame to transmit first.

A. Medium access control protocol.

The LMACM can schedule the simultaneous arrival MAC frames that from different LMACs, then sequentially send them to the CTX according to their frame's priority. The schedule policy is to increase the priority of all waiting frames by one after the highest priority MAC frame has been transmitted. It can avoid the lower priority MAC frame waiting too long and losing the mission. The MAP network only has a token around its ring, so there is a medium access operation happening at one time. Hence its MAC frames are no priority. However the DMMAP network has many simultaneous tokens around its logical rings; consequently many medium access operations in a station may need to send at a time. We analyzed the characteristic of those MAC frames, and assigned priorities to them. There are brief descriptions as below:

1) A token frame is bounded too long will lose the real time characteristic of network, and cause a channel reinitialization mistake.
2) A long delayed set_successor frame may lose (1) to response the who_follows frame in time that makes an unnecessary reinitialization, (2) to response the solicit_successor _2 frame that will cause to give up maintain the logical channel, and (3) an entering the channel chance.
3) The other frames delayed too long, that may take down a little system performance.

B. Data frame control protocol

The priority scheme of data frame in DMMAP network is similar as that of token bus network. Halsall [6] mentions the application of token bus. It is primarily in the domains manufacturing automation and process control, and typical usage of the four access classes may be:

1) Synchronous: Urgent messages such as those relating to critical alarm conditions and associated control functions.
2) Urgent asynchronous: messages relating to ring management functions and normal control actions.
3) Normal asynchronous: messages relating to routine data gathering for data logging purposes
4) Time available: messages relating to program downloading and general file transfers.

The DATAM in a DMMAP station needs four priorities received queues and transmitting queues to buffer its data frames, as well as many timers to keep up the mechanism of medium access control. It uses a set of four timers, one for each of the lower three access classes and one for ring maintenance. When the DATAM begins processing the token at a given access class, the associated timer is reloaded with the value of the target rotation time (TRT) for that level. When the DATAM again receives the token, it may send data of that access class until the residual time in the associated token rotation timer has expired. The residual time from the current token rotation timer is loaded into the token hold timer just before the token rotation timer is reloaded. The DATAM may send data frames of the corresponding access class as long as the token hold timer has not expired. The high priority token hold time (HPTHT) is the maximum amount of time in which the DATAM may transmit data from the queue of synchronous access class to the channel that is holding a token. When the station is sending synchronous access class message the value of HPTHT is loaded into the token hold timer. Thus Synchronous access class messages are limited to only a fixed number of bytes regardless of current network loading [7]. In view of real time characteristic, the frames of lower three priority access classes must be limited to send, under associated throughput limit value.

II. Analysis

The DMMAP defines the operation of the protocol and its operational parameters such as the three TRT's, but it does not specify default values for the TRT's. The users may not be properly able to associate priority with token cycle time. We present an analytic model that calculates the values of the TRT's that will implement a user-defined priority scheme. It will provide optimum service to that access class until network throughput exceeds the user-defined limit.

A. Definitions

An active access class at a station on DMMAP network is termed a server. The following notation is used:

X_T ≡ duration of a token transmission in a healthy condition (seconds/token transmission)

s ≡ the Synchronous access class

ua ≡ the Urgent Asynchronous access class

na ≡ the Normal Asynchronous access class

ta ≡ the Time Available access class

λ_{ac} ≡ the mean message arrival rate at each server of an access class, where ac is s, ua, na, or ta

S ≡ the set of all Synchronous access class servers

UA ≡ the set of all Urgent Asynchronous access class servers

NA ≡ the set of all Normal Asynchronous access class servers

TA ≡ the set of all Time Available access class servers

R ≡ set of all distinct servers on all logical rings (S, UA, NA, TA)

λ_x ≡ the mean message arrival rate at server $x \in R$ (in messages / second)

$HPTHT$ ≡ the High Priority Token Hold Time (in seconds)

TRT_x ≡ the Target Rotation Time at server $x \in (UA, NA, TA)$ (in seconds)

TRT_{asy} ≡ the Target Rotation Time at each server of an access class, where asy is ua, na, or ta (in seconds)

$TCT_{x,i,j}$ ≡ the time form the end of the (i-1)st service period until the beginning of the ith service period (in seconds) on the jth logical ring
≡ token circulation time as seen by the jth logical ring of server x on the ith token cycle (the time the token is away from x)

$\overline{TCT}_{x,j}$ ≡ token circulation time as seen by server x on the jth logical ring of server x

$TC_{x,i,j}$ ≡ time from the end of the (i-1)st service period until the end of the ith service period (in seconds) on the jth logical ring
≡ token cycle time as seen by the jth logical ring of server x on the ith token cycle

$\overline{TC}_{x,i}$ ≡ token cycle time as seen by server x on the ith token cycle

$A_{x,i}$ ≡ the number of message arrivals at x between the (i-1)st and ith service periods

$Q_{x,i}$ ≡ the number of messages enqueued at server x after the (i-1)st service period

$f(n)$ ≡ time required to transmit n messages (in octet times)

t ≡ residual time in the token hold timer (in octet times)

$eff(t)$ ≡ the effective amount of time which a server can transmit message
≡ max(0, $t + f(1)$ - one octet time) (in octet times)

$TS_{x,i,j}$ ≡ duration of the ith service period (in seconds) on jth logical ring of sever x

$\overline{TS}_{ac}$ ≡ average service time availabel to every server of an access class, where ac is s, ua, na, or ta (in seconds)

C ≡ channel capacity in bits/second

B. Assumptions

In the following section, an analytic expression for the token cycle time of a DMMAP network is derived. The derivation is based on the following assumptions about the characteristics of the network:

1) The protocol management overhead is assumed to be negligible.
2) Stations are permanent members of the logical rings.
3) Each station has traffic at all four access class.
4) Message arrivals at each station follow a Poisson process.
5) Messages from all the servers are of constant length lM bits and take XM seconds for transmission.
6) The token is of length 112 bits for a 10 Mbit/s bus and takes XT seconds for transmission.
7) All Synchronous access-class servers have the same $HPTHT$ setting.
8) Each station has $HPTHT$ set to the maximum allowed value (52.43 ms for a 10 Mbit/s bus). so

messages from the Synchronous access class normally are transmitted on the same token cycle as they arrive.

9) All servers of the same priority have identical traffic.
10) All servers of the same priority have the same *TRT* settings.
11) The TRT's conform to the relation $TRTua > TRTna > TRTta$.
12) Each active access class has N servers.
13) Each server has M channels.
14) Token overlay probability P_i is a geometric random variable, it is the probability at the frequency of token overlay occurrences is i, and sum of all probability is one, i.e., $P_0 + P_1 + ... + P_{M-1} = 1$.
15) DATAM adopts the second mechanism when token overlay occurrence.

C. Service time

There are many tokens simultaneously around those rings in the DMMAP network. The DATAM in a DMMAP station may arrive many tokens, but it can take one token at a time. There are two problem methods to handle the overlapped token. The first method is that the token has less service time will be sent back to its channel for the performance reason [2]. This method makes a phenomenon that the first coming token will be firstly sent back, therefore it is called *FCFB* as fig. 6. The *token overlay* phenomenon makes a non-finished token to shorten its service time. For an example, A DMMAP network has M channels which are attached by N stations, and the token holder (station) can send L frames (or $\Gamma = L \cdot l_M \cdot X_M$ seconds) at most. Furthermore, A probability factor P_{ijy} is defined as the probability of token overlay occurence that the station i sends y frames in its received j token period. In the figure 6 (a) occurence, the station i sends y (or $2L-d$) frames when two tokens arrived at a period of service time Γ. The DATAM sends the token of channel 1 back at the time t_1 of the token of channel 2 is arrived and loses a service time $d \cdot l_M \cdot X_M$. seconds.

For station 1 in a DMMAP network that using the FCFB method and L=3.

- In no token overlay occurence as fig. 6 (a): y = L = 3, i=1, j=1.

  The truncated service time factor of no token overlay in station 1 is

  $$\Delta_{11} = P_{113} \cdot 1 .$$

- In two token overlay occurence,

  as fig. 6 (b): y = 2L-1 = 5, i=1, j=2.

  as fig. 6 (c): y = 2L-2 = 4, i=1, j=2.

  as fig. 6 (d): y = 2L-3 = L= 3, i=1, j=2.

The truncated service time factor of two token overlay in station 1 is

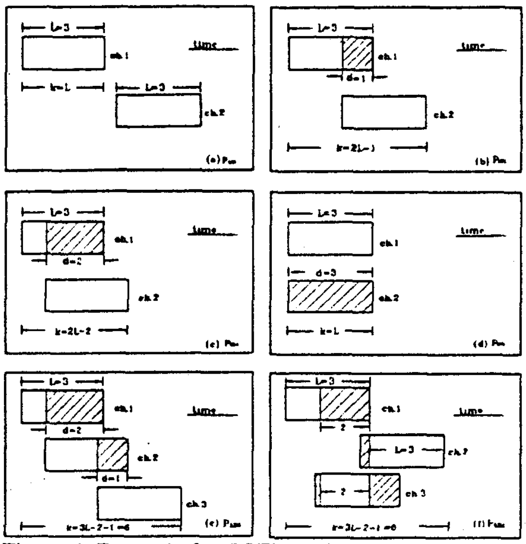

Figure 6. Example for FCFB policy.

$$\Delta_{12} = P_{123} \cdot \frac{1}{2} + P_{124} \cdot \frac{4}{6} + P_{125} \cdot \frac{5}{6}$$

$$= \sum_{y=L}^{\eta} P_{12y} \frac{y}{2 \cdot L}, \quad \eta = (L-1) \cdot 2 + 1$$

- In three token overlay occurence,

  as fig. 6 (e) and (f): y = 3L-2-1 = 6, i=1, j=3.

  ••••••••••••••••••

  The truncated service time factor of three token overlay in station 1 is

$$\Delta_{13} = P_{133} \cdot \frac{1}{3} + P_{134} \cdot \frac{4}{9} + \cdots + P_{137} \cdot \frac{7}{9} = \sum_{y=L}^{\eta} P_{13y} \frac{y}{3 \cdot L}, \quad \eta = (L-1) \cdot 3 + 1$$

, and so on.

Therefore the token overlay statistic factor of station 1 can represent as

$$\Delta_1 = \sum_{j=1}^{M} \Delta_{1j} = \sum_{j=1}^{M} \sum_{y=L}^{\eta} P_{1jy} \cdot \frac{y}{j \cdot L}, \quad \eta = (L-1)j + 1.$$

The token overlay statistic factor of the DMMAP network is

$$\Delta = \sum_{i=1}^{N} \Delta_i = \sum_{i=1}^{N} \sum_{j=1}^{M} \sum_{y=L}^{\eta} P_{ijy} \cdot \frac{y}{j \cdot L}, \quad 0 \le P_{ijy} \le 1, \quad \eta = (L-1) \cdot j + 1$$

If x is a synchronous server then the average service time of x can be expressed as

$$\overline{TS}_x = \min\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\sum_{y=L}^{\eta} P_{ijy}\frac{y}{j\cdot L}\cdot M\cdot eff(HPTRT), f(\overline{Q}_x+\overline{A}_x)\right),$$
$$0 \le P_{ijy} \le 1,\ \eta = (L-1)\cdot j+1$$

where $\overline{Q}_x$ is the average queue length at x and $\overline{A}_x$ is the average number of arrivals at x during a token cycle. The sum of all probability factor P_{ijy} is always one in accordance with to probability law $P[Sample\ space] = 1$. If x is an asynchronous server then the average service time of server x can be express as

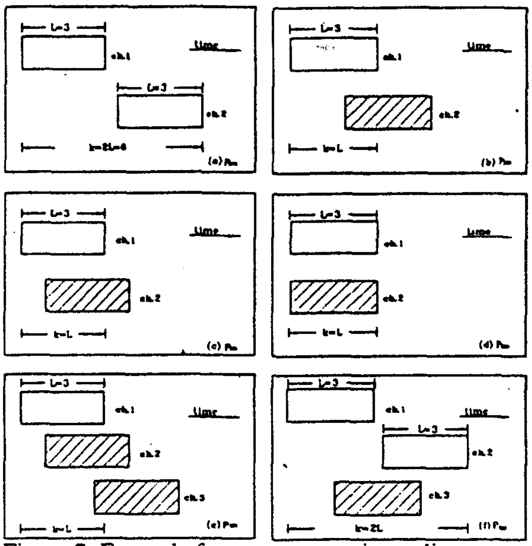

Figure 7. Example for non-preemptive policy.

$$\overline{TS}_x = \begin{cases} \min\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\sum_{y=L}^{\eta} P_{ijy}\frac{y}{j\cdot L}\cdot M\cdot eff(TRT_x-\overline{TCT}_{x,j}), f(\overline{Q}_x+\overline{A}_x)\right), \\ \quad 0 \le P_{ijy} \le 1,\ \eta = (L-1)\cdot j+1, \quad for\ \overline{TCT}_{x,j} < TRT_x \\ 0, \quad for\ \overline{TCT}_{x,j} \ge TRT_x \end{cases}$$

where $\overline{TCT}_{x,j}$, is the average token circulation time at every server x, $\overline{Q}_x$ is the average queue length at x and $\overline{A}_x$ is the average number of arrivals at x during a token cycle.

The second method is a simple method. The DATAM will rejects any arrived token when it is holding a token, hence every accepted token can finish its work without any preemptive interference as fig.7. This policy is called *non-preemptive*. A probability factor P'_{ijk} is defined as the probability of token overlay occurence that the station i sends $(k\cdot L)$ frames in its received j token period. In the figure 7, the token of channel 2 was arrived at station i at the time t_1 and it is sent back at once. The sending message length of station i is L frames (or $\Gamma = L\cdot l_M\cdot X_M$ seconds) when two tokens arrived at a period of service time Γ. In non-preemptive, the sending message length or the loss length is a multiple of Γ.

For station 1 in a DMMAP network that using the non-preemptive method.

- In no token overlay occurence as fig. 7(a): $k=\Gamma$, $i=1$, $j=1$.
  The truncated service time factor of no token overlay in station 1 is
  $$\Psi_{11} = P'_{111}\cdot 1\ .$$
- In two token overlay occurence,
  as fig. 7 (b): $k = \Gamma$, $i=1$, $j=2$.
  as fig. 7 (c): $k = \Gamma$, $i=1$, $j=2$.
  as fig. 7 (d): $k = \Gamma$, $i=1$, $j=2$.
  The truncated service time factor of two token overlay in station 1 is
  $$\Psi_{12} = P'_{121}\cdot\frac{1}{2}\ .$$
- In three token overlay occurence,
  as fig. 7 (e): $k = \Gamma$, $i=1$, $j=3$.
  as fig. 7 (f): $k=2\Gamma$, $i=1$, $j=3$.
  ....
  The truncated service time factor of three token overlay in station 1 is
  $$\Psi_{13} = P'_{131}\cdot\frac{1}{3} + P'_{132}\cdot\frac{2}{3}\ ,\ \text{and so on.}$$

Therefore the token overlay statistic factor of station 1 can represent as

$$\Psi_1 = \sum_{j=1}^{M}\Psi_{1j} = \sum_{j=1}^{M}\sum_{k=1}^{\omega} P'_{1jk}\cdot\frac{k}{j},\quad \begin{cases}\omega = 1, & \text{for } j = 1\\ \omega = j-1, & \text{for } j \ge 2\end{cases}$$

The token overlay statistic factor of the DMMAP network is

$$\Psi = \sum_{i=1}^{N}\Psi_i = \sum_{i=1}^{N}\sum_{j=1}^{M}\sum_{k=1}^{\omega} P'_{ijk}\cdot\frac{k}{j},\quad 0 \le P'_{ijk} \le 1,\ \begin{cases}\omega = 1, & \text{for } j = 1\\ \omega = j-1, & \text{for } j \ge 2\end{cases}$$

(If x is a synchronous server then the average service time of x can be expressed as

$$\overline{TS}_x = \min\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\sum_{k=1}^{\omega} P'_{ijk}\frac{k}{j}\cdot M\cdot eff(HPTRT), f(\overline{Q}_x+\overline{A}_x)\right),$$
$$0 \le P'_{ijk} \le 1,\ \begin{cases}\omega = 1, & \text{for } j = 1\\ \omega = j-1, & \text{for } j \ge 2\end{cases}$$

where $\overline{Q}_x$ is the average queue length at x and $\overline{A}_x$ is the average number of arrivals at x during a token cycle. The sum of all probability factor P'_{ijk} is always one in accordance with to the probability law. If x is an asynchronous server then the average service time of server x can be express as

$$\overline{TS}_x = \begin{cases} \min\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\sum_{k=1}^{\omega} P'_{ijk}\frac{k}{j}\cdot M\cdot eff(TRT_x-\overline{TCT}_{x,j}),\ f(\overline{Q}_x+\overline{A}_x)\right), & \\ 0 \le P'_{ijk} \le 1, \begin{cases}\omega = 1, & \text{for } j=1\\ \omega = j-1, & \text{for } j \ge 2\end{cases} & \text{for } \overline{TCT}_{x,j} < TRT_x \\ 0, & \text{for } \overline{TCT}_{x,j} \ge TRT_x \end{cases}$$

(where $\overline{TCT}_{x,j}$, is the average token circulation time at every server , $\overline{Q}_x$ is the average queue length at x and $\overline{A}_x$ is the average number of arrivals at x during a token cycle.

Assuming an average message arrival rate of λ_x at jth logical ring server x, the number of messages that have arrived while the x is being served on the ith token cycle is

$$A_{x,i} = \sum_{j=1}^{M}\lambda_x TCT_{x,i,j} + \lambda_x TS_{x,i,j} = \lambda_x\overline{TC}_{x,i}$$

Hence, the average number of messages that have arrived at any server x, $x \in (S, UA, NA, TA)$, during a token cycle can be expressed as $\overline{A}_x = \lambda_x\overline{TC}_x$.

Assuming all the messages from the Sychronous access class get transmitted during the same token cycle in which they arrive, and all the messages from that access class and from access classes of higher priority that have arrived during a particular token cycle get transmitted during the same token cycle in which they arrive. Hence, $\overline{Q}_x$ can be considered to be negligible. The average service time can be expressed as

$$\overline{TS}_x = f(\overline{Q}_x + \overline{A}_x) = f(\overline{A}_x) = X_M\,\lambda_x\overline{TC}_x \qquad (1)$$

D. Determining TRT's

If the designer determines that service at the Time Available access class should reach a maximum when the network throughput is a certain fraction α of network capacity, $0 \le \alpha \le 1$, then

$$\alpha = \frac{N\,\Lambda_\alpha\, l_M}{M\,C\,\overline{TC}_\alpha} \qquad (2)$$

where Λ_α is the mean number of messages transmitted per station per token cycle at a throughput of α, and $\overline{TC}_\alpha$ is the average token cycle time at a throughput of α. Hence, the average number of messages that arrive during the time interval $\overline{TC}_\alpha$ should equal the mean number of messages transmitted in this time interval.

$$\Lambda_\alpha = (\lambda_s + \lambda_{ua} + \lambda_{na} + \lambda_{ta})\overline{TC}_\alpha$$

substituting for Λ_α in (2) we obtain the sum of λ to be

$$\lambda_s + \lambda_{ua} + \lambda_{na} + \lambda_{ta} = \frac{\alpha\,C\,M}{N\,l_M}$$

Since token cycle time can be expressed as

$$\overline{TC}_\alpha = P_0(NX_T + \frac{N}{M}X_M\Lambda_\alpha) + P_1(NX_T + \frac{N-1}{M}X_M\Lambda_\alpha) + \ldots + P_{M-1}(NX_T + \frac{N-M+1}{M}X_M\Lambda_\alpha)$$

$$= (P_0 + P_1 + \ldots + P_{M-1})NX_T + \frac{P_0N + P_1(N-1) + \ldots + P_{M-1}}{M}X_M\Lambda_\alpha$$

$$= NX_T + \sum_{i=0}^{M-1} P_i\frac{N-i}{M}X_M\Lambda_\alpha \qquad (3)$$

substituting for Λ_α from (3) yields

$$\overline{TC}_\alpha = NX_T + X_M\sum_{i=0}^{M-1} P_i\frac{N-i}{M}(\lambda_s + \lambda_{ua} + \lambda_{na} + \lambda_{ta})\overline{TC}_\alpha$$

$$eff(TRT_{ta}) = \overline{TC}_\alpha = \frac{NX_T}{1 - X_M(\lambda_s + \lambda_{ua} + \lambda_{na} + \lambda_{ta})\sum_{i=0}^{M-1} P_i\frac{N-i}{M}}$$

If the designer determines that service at the Normal Asynchronous access class should reach a maximum when the network throughput reaches β , $0 \le \alpha < \beta \le 1$, then

$$\beta = \frac{N\,\Lambda_\beta\, l_M}{M\,C\,\overline{TC}_\beta} \qquad (4)$$

where Λ_β is the mean number of messages transmitted per station per token cycle at a throughput of β, and $\overline{TC}_\beta$ is the average token cycle time at a throughput of β. Hence, the average number of messages that arrive during the time interval $\overline{TC}_\beta$ should equal the mean number of messages transmitted in this time interval.

$$\Lambda_\beta = (\lambda_s + \lambda_{ua} + \lambda_{na})\overline{TC}_\beta$$

substituting for Λ_β in (4) we obtain the sum of λ to be

$$\lambda_s + \lambda_{ua} + \lambda_{na} = \frac{\beta C M}{N l_M}$$

Since token cycle time can be expressed as

$$\overline{TC_\beta} = NX_T + \sum_{i=0}^{M-1} P_i \frac{N-i}{M} X_M \Lambda_\beta \quad (5)$$

substituting for Λ_β from (5) yields

$$\overline{TC_\beta} = NX_T + X_M \sum_{i=0}^{M-1} P_i \frac{N-i}{M} (\lambda_s + \lambda_{ua} + \lambda_{na})\overline{TC_\beta}$$

$$eff(TRT_{na}) = \overline{TC_\beta} = \frac{NX_T}{1 - X_M(\lambda_s + \lambda_{ua} + \lambda_{na}) \sum_{i=0}^{M-1} P_i \frac{N-i}{M}}$$

If the designer determines that service at the Urgent Asynchronous access class should reach a maximum when the network throughput reaches γ $0 \le \alpha < \beta < \gamma \le 1$, then

$$\gamma = \frac{N \Lambda_\gamma l_M}{M C \overline{TC_\gamma}} \quad (6)$$

Using logic similar to the previous discussion, the number of message transmitted during the time interval $\overline{TC_\gamma}$ is

$$\Lambda_\gamma = (\lambda_s + \lambda_{ua})\overline{TC_\gamma}$$

substituting for Λ_γ in (6) we obtain the sum of λ to be

$$\lambda_s + \lambda_{ua} = \frac{\gamma C M}{N l_M}$$

Since token cycle time can be expressed as

$$\overline{TC_\gamma} = NX_T + \sum_{i=0}^{M-1} P_i \frac{N-i}{M} X_M \Lambda_\gamma \quad (7)$$

substituting for Λ_γ from (7) yields

$$\overline{TC_\gamma} = NX_T + X_M \sum_{i=0}^{M-1} P_i \frac{N-i}{M} (\lambda_s + \lambda_{ua})\overline{TC_\gamma}$$

$$eff(TRT_{ua}) = \overline{TC_\gamma} = \frac{NX_T}{1 - X_M(\lambda_s + \lambda_{ua}) \sum_{i=0}^{M-1} P_i \frac{N-i}{M}}$$

III. Conclusion.

The high speed network DMMAP provides a multichannel environment to increase the network bandwidth. However the multiple independent token makes a simultaneous problem. In this paper, we propose two mechanisms to solve that problem, and analyze the token overlay probability. The operation of the priority scheme in the DMMAP is totally dependent upon the choice of the three TRT for the lower three priority classes. We have reformulated that problem in term of network throughput. In the formulation, the user defines three throughput limits α, β, and γ, with $0 \le \alpha < \beta < \gamma \le 1$, such that Time available traffic will be serviced when network throughput is less than α, Normal asynchronous will be serviced when throughput is less β, and Urgent asynchronous will be serviced when throughput is less than γ. Each access class formula that computes the setting of the TRT will provide optimum service to that access class, until network throughput exceeds the user-defined limit (α, β, or γ).

Reference

[1] W. S. Hwang and L. Y. Kung, "Distributed Multichannel Manufacturing Automation Portocol Network", IEEE Singapore ICCS/ISITA '92 Conference Proceedinds, pp 507-511, 1992.

[2] W. S. Hwang and L. Y. Kung, "Analyze Gain of Distributed Multichannel Manufacturing Automation Portocol Network", IEEE Singapore ICCS/ISITA '92 Conference Proceedinds, pp 512-516, 1992.

[3] R. MANGALA GORUR and ALFRED C. WEAVER, "Setting Target Rotation Time In An IEEE Token Bus Network," IEEE Transaction on Industrial Electronic, vol.35, no.3, pp.366-371, 1988.

[4] Andrew S. Tanenbaum, "Computer Networks" Second Edit, Prentice-Hall International, pp 163, 1988.

[5] Motorola, " MC68824 Token Bus Controller User's Manual", pp 1-3, 1987.

[6] Fred Halsall, "Data Communications, Computer Networks and OSI" Second, Addison Wesley, pp355, 1988.

[7] IEEE Computer Society, "Token-Passing Bus Access Method and Physical Layer Specifications ", pp 88, December 1984.