KSME Journal, Vol. 10, No. 3, pp. 321~331, 1996

Performance Management of Token Bus Networks for Computer Integrated Manufacturing

Sang-ho Lee*, Joon-woo Son* and Suk Lee*

(Received November 27, 1995)

This paper focuses on performance management of computer communication networks serving computer integrated manufacturing systems. The performance management aims to improve the network performance in handling various types of messages by on-line adjustment of protocol parameters. The principle of fuzzy logic has been used in formulating the performance management rules and in deriving management decisions. Three types of the fuzzy performance management have been developed for IEEE 802.4 protocol standard and have been evaluated via discrete event simulation.

Key Words: Computer Integrated Manufacturing, Production Management, Local Area Network, Communication Equipment, Manufacturing Automatiőn Protocol, Priority Mechanism, Fuzzy Network Performance Manager, Data Latency, Token Circulation Time

1. Introduction

Computer networking serves as communication links between islands of automation for productivity improvement in computer integrated manufacturing (CIM). Computer networks are capable of interconnecting various devices from different vendors, which allow a system designer to be flexible on his/her initial design and future reconfigurations. Therefore, the major advantages of computer networking include evolutionary system growth, better utilization of system resources and improved reliability based on the availability of additional resources. Due to these benefits, computer networking is ideal for the role of the nerve system of advanced manufacturing systems where various and spatially distributed components and subsystems are integrated to realize computer integrated manufacturing.

According to the area of application, a computer nutwork should be tailored at the design stage by selecting appropriate network protocols and their parameters since requirements imposed on a network may widely vary (Ray, 1988 ; MAP ; and TOP). Even after design and installation of a network, several groups of functions are required to adjust the network so that initial design objectives are satisfied. This is because the condition under which a network operates may be different from that considered at the design stage. For example, in a manufacturing system network, the number of devices on the network will change continuously due to addition and deletion of devices for maintenance and repair. The network should be able to allow these changes without disrupting other devices on the network. Therefore, the network must always adapt to the dynamic environment, which is extremely essential because the network serves many crucial functions of the munufacturing system.

The responsibility of adapting network belongs to network management which aims to maintain reliable, flexible and efficient operations. The major components of network management are fault management, configuration management and performance management. As its name implies, fault management is responsible for detection, isolation and recovery from component

* Department of Mechanical and Production Engineering and Research Institute of Mechanical Technology, Pusan National University

failure. Configuration management is related to network initialization and accommodation of any network configuration changes. It is also responsible for reconfiguration requested by fault and performance management. Performance management is related to improvement of the network performance via adjustment of protocol parameters. The discipline of network management is a relatively young field, and its importance is being realized as the number and diversity of the subscribers increase. Basic results have been reported on the architecture, service definition, protocol standardization and distribution of network management functions (Thompson, 1986 ; Saydam and Sethi, 1987 ; and Klerer, 1988). Several standards for network management have been already published or under preparation as an important part of the integrated protocol suite (MAP ; TOP ; IEEE, 1992 ; and SAE, 1987). However, analytical research has been largely limited to the classical areas of routing and flow control (Bertsekas and Gallager, 1987). Apparently, researchers have not addressed the key issue of how to accomplish the network management tasks in real time as problems or disruptions arise either in the network operations or in the process that is served by the network. This has happened partly because the development of a methodology for network management is beyond the scope of standardization. Nevertheless, from the users' viewpoint, network management is very crucial for maintaining uninterrupted operations because many important functions are dependent on the services provided by the network.

Among the major functions of network management, performance management, which is responsible for improving the network performance by adjusting protocol parameters, has become important because characteristics of network traffic are essentially dynamic in large-scale integrated systems such as manufacturing systems. Network operations with default settings of protocol parameters may not prove to be efficient under diverse operating conditions for a prolonged period of time. For example, a network for computer integrated manufacturing (CIM) should be able to deliver various messages such as CAD files, interpersonnel electronic mails, control signals and sensor data within their time limits of delivery under changing characteristics of network traffic (Ray, 1988). The dynamic characteristics are caused by common events like arrival of new production orders and/or failure in system components. Even though individual protocols offer mechanisms to handle various types of messages efficiently, the key parameters of the protocol suite, which determine the network performance, are at the network operator's disposal. The operator has to adjust these parameters based on certain heuristics and his/her individual experience since the relationship between protocol parameters and network performance is not well known in general and no systematic approach for parameter adjustment exists.

This paper presents development of three types of performance management procedure for token bus networks using fuzzy reasoning. These procedures aim to improve the network performance by on-line adjustment of protocol parameters. The developed procedures have been evaluated through a series of simulation experiments of an IEEE 802.4-based network. The procedure presented in this paper offers a step forward to bridging the gap between management standards and users' demands for efficient network operations since most standards such as ISO and IEEE address only the architecture, services and interfaces for network management. Furthermore, it demonstrates the simplicity and effectiveness of fuzzy reasoning for performance management.

This paper is organized into five sections including the introduction. Section 2 gives a brief summary of IEEE 802.4 token bus protocol standard. The development of the performance management is presented in Sec. 3. The evaluation results are presented and discussed in Sec. 4. Finally, conclusions are presented in Sec. 5 along with recommendations for future research.

2. Token Bus Protocol

A token bus protocol is a distributed controlled-access protocol for the medium access control (MAC) layer. The right to use the medium is explicitly controlled by a special bit pattern called a token and the responsibility of controlling the use of the medium lies with every station. This section describes the major features of IEEE 802.4 token bus protocol (ANSI/IEEE, 1990) for which the performance procedure has been developed.

A token bus network consists of a number of stations connected via a broadcast medium on which any transmission from a station can be heard by all stations. The right to transmit a message is given to a station when it receives a special bit pattern called a token. The token is passed from a station to another following a sequence of station addresses. The last station in the sequence sends the token back to the first station to form a logical ring. A station may transmit its messages before it passes the token to the next station in the logical ring sequence. A station with the token has complete control of the medium for a finite period of time. The length of this period depends on the number of waiting messages and the status of several timers as explained later on priority mechanism. A station can transmit a number of messages or can pass the token to its successor (i. e., the next station in the logical ring sequence) without any transmission.

The IEEE token bus protocol has a priority mechanism of four classes, namely 6, 4, 2 and 0, among which the priority level 6 has the highest privilege of medium access. A token holding timer (THT) and three token rotation timers, i. e., TRT4, TRT2 and TRT0, regulate message transmissions for the priority class 6, 4, 2 and 0, respectively. The priority class 6 messages are allowed to start transmission within a period equal to the length of THT. For lower priority class messages, the initiation of a transmittion must not occur beyond the residual period (i. e., the time left until its expiration) of the corresponding timer. If a timer expires while the corresponding priority message is being transmitted, the transmission will be continued to completely finish the current message and no further transmission is allowed until the instant of next token reception.

If a station receives the token, it performs the token processing and self-diagnostics during the period of response time before any transmission. At the end of the response time, the station resets THT to its full value and checks whether any message is waiting in the priority 6 queue. If the queue is empty, the chance of transmission is given to the priority class 4 ; otherwise, the station start its THT and begins to transmit the oldest one among the priority 6 messages. At the completion of a message transmission, the station checks whether THT has expired and whether there are more priority 6 messages waiting. If THT is not expired and there is a message waiting, the station starts another message transmission. This procedure continues either until there is no waiting message or until THT expires.

After the station finishes the procedure for the priority class 6 messages, it checks the status of TRT4 that was started at the previous token circulation. If it is expired or no priority 4 message is waiting, then TRT4 is reset to its full value and restarted and the chance of transmission is passed to the priority class 2 messages without any transmission of the priority 4 message. If TRT4 is not expired and a priority 4 message is waiting, then THT is restarted after being reset to the remaining value of TRT4 and TRT4 is reset to its full value and restarted. The priority class 4 messages can be transmitted consecutively either until THT expires or until there is no priority 4 message waiting to be transmitted.

When one of two conditions, namely, THT expiration and absence of priority class 4 message, is satisfied, the station begins the same procedure for TRT2 and the priority 2 messages and continues for TRT0 and the priority 0 messages. After the station completes the procedure for the priority class 0, the token is passed to the successor station. This priority mechanism is summarized by a flowchart in Fig. 1.

The priority mechanism allocates a fixed amount of time for the priority class 6 while a variable length of time for the lower 3 classes depending on the time elapsed between two adjacent transmission opportunities. The values of timers and the amount of network traffic have a direct influence on the available time periods for

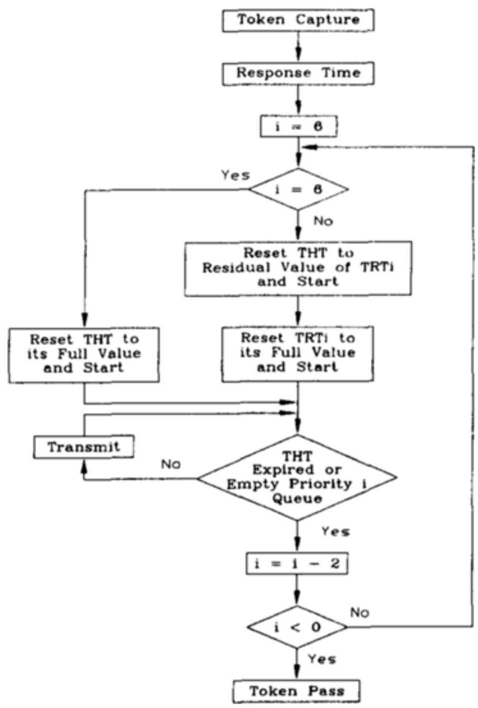


Fig. 1 Flowchart of the IEEE 802.4 priority mechanism

message transmission. The available time periods are, in turn, closely related to data latency (time spent by a message inside the network system, i. e., time interval from the instant when a message enters a queue for transmission to the instant when it is received by the destination station) and throughput that are two major factors of network performance. Therefore, for a given traffic condition, it is crucial to select an appropriate set of timer values in order to obtain an acceptable level of performance.

3. Fuzzy Performance Manager

The role of performance management is to manipulate the adjustable protocol parameters in real time so that the network can adapt itself to a given environment. In the previous research (Lee and Ray, 1993), this was divided into two tasks: (1) performance evaluation to find how changes in protocol parameters affect the network performance measure, and (2) decision making on how to adjust protocol parameters. The first task is essentially equivalent to finding a relationship between the network performance and the protocol parameters. In the worst situation, this task is required to estimate the network performance at some points in the neighborhood of the current parameter settings. The second task of performance management is to decide direction and magnitude of the parameter adjustment vector, i. e., what are the next settings for improved network performance, utilizing pieces of information provided by the first task and the history of performance. In order to accomplish these tasks, the principles of perturbation analysis (PA) of discrete event dynamic systems (DEDS), stochastic approximation (SA) and learning automata (LA) have been combined and successfully applied to a token bus network.

This work was able to avoid many difficulties from which the conventional techniques such as queueing theory and discrete event simulation suffer. The difficulties include requirement of satisfying some unrealistic assumptions and excessive computational burden. From the viewpoint of implementing the performance management, however, it requires every station to maintain an algorithm for PA and to communicate with its predecessor and successor in order to exchange necessary information for PA. This may result in unjustifiable complexity, cost and some changes in the protocol itself.

The difficulties in developing a practical performance management procedure comes from the fact that the relationship between the four timer values and a specified network performance is not well known under general conditions and another fact that the information available for performance management always contains uncertainties due to the network observation over a finite period of time. For this kind of systems where the quantitative input/output relation is not known, fuzzy logic (Zadeh, 1973) has been successful in many practical applications. This is mainly because fuzzy logic is much closer to human thought process and natural language than traditional mathematical logic. Since it lends itself to represent approximate and inexact human knowledge accumulated from experience (Lee, 1990), it is very suitable for this problem and therefore selected as an alternative for performance management. The performance management procedure developed in this research utilizes the knowledge base of a human expert in the form of fuzzy rules instead of constructing the history of the network operations by using PA, determining timer values by using SA and selecting appropriate SA parameters by LA. The replacement of PA, SA and LA with the fuzzy rules and their inference mechanism reduces the complexity of the performance management procedure without any necessity to modify the protocol.

3.1 Formulation of knowledge base

In order to formulate the required knowledge base, a token bus network was simulated using SIMAN (Pegden, Shanon and Sadowski, 1990) over a wide range of timer settings. From the series of simulation experiments, the following observations were made.

- In general, the longer a timer is, the shorter the corresponding data latency is.
- In order to reduce the data latency for priority class 6, it may be more effective to reduce TRT's than to increase THT.
- When all TRT's are quite long compared to token circulation time (TCT, time spent for the token to make a complete circulation around the logical ring), the waiting time inside a transmission queue is almost identical regardless of priority class.
- When a TRT is not long enough compared to TCT, it is possible for the corresponding priority class not to be allowed to transmit. In this case, throughput of this class will be less than the offered network traffic and its data latency will be large.
- Decreased data latency for a priority class may adversely affect data latencies of other classes.
- If the queue capacities are large enough, throughputs for all priority classes are equal to the offered network traffic, i. e., all the messages are transmitted without getting deleted except with some choices of extreme timer values.

The objective of the fuzzy performance manager is to maintain the average data latencies of all priority classes below the user-specified corresponding levels. Therefore, the performance manager has to know the average data latency for each priority class (D_i, $i=6, 4, 2, 0$) as a minimum input set. The outputs of the fuzzy performance manager are changes in four timer values (ΔTHT, ΔTRT4, ΔTRT2, ΔTRT0).

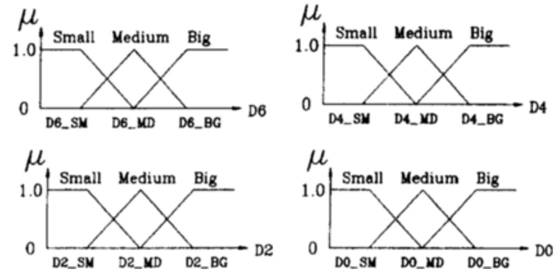


Fig. 2 Membership functions for fuzzy inputs

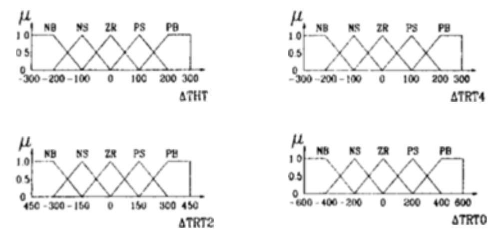


Fig. 3 Membership functions for fuzzy outputs

For each priority class, three primary fuzzy sets are defined over a continuous universe of discourse for average data latency. These sets are named as Small, Medium and Big as shown in Fig. 2. The three numerical values to define the membership functions are provided by the network user and the smallest value represents the satisfactory level of data latency for a given priority class. For the outputs, the fuzzy performance manager has five sets : Negative Big (NB), Negative Small (NS), ZeRo (ZR), Positive Small (PS), and Positive Big (PB) as shown in Fig. 3.

Based on the above knowledge base, three types of fuzzy network performance manager (FNPM) have been developed. They are named as basic, TCT-based and switch-type FNPM's.

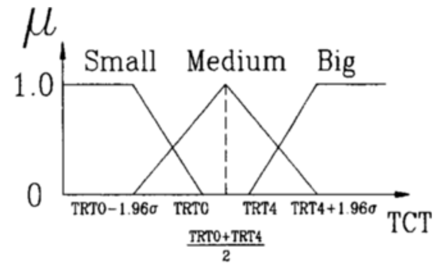
3.2 Basic fuzzy network performance manager

The basic FNPM utilizes only the average data latencies to determine the timer changes. It is designed to adjust timers by always considering data latencies for all priority classes. Its rule base contains all the possible combinations of average data latencies (total of 81 rules).

3.3 TCT-based fuzzy network performance manager

In addition to average data latencies, the TCT-based FNPM accepts average token circulation time(TCT) as an additional input. The rationale for this inclusion is that the timer lengths has to be determined in relation to TCT which indicates the amount of traffic on the network. For TCT, there are three primary sets, i. e., Short, Medium, and Long whose membership functions are defined with the current timer values and the standard deviation of TCT as shown in Fig. 4.

Using five input and four output variables, twelve IF-THEN rules are developed for the fuzzy performance manager. These rules are summarized in Table 1. The first rule is intended to reduce all TRT's when TCT is very short since this situation results in less distinction among the average data latencies of the three lower priority classes. The second rule is the opposite case; when TCT is too long compared to TRT's, this



* σ is standard deviation of TCT

Fig. 4 Membership functions for the TCT of the TCT-based FNPM

rule will increase TRT's as long as D_6 is Small. This rule is expected to increase chances of message transmissions for lower three classes. When TCT is Long and D_6 is Medium, the third rule applies in order to reduce D_6 first. The rest of rules apply when TCT is Medium compared to TRT's and their intentions are self-evident.

3.4 Switch-type fuzzy network performance manager

The switch-type FNPM consists of two parts, FNPM1 and FNPM2 as shown in Fig. 5. This FNPM is designed to improve the average data latency of class 6 only by using FNPM1 either until D_6 becomes acceptable or until the transmis-

Table 1 Fuzzy rules of TCT-based FNPM

	TCT	D6	D4	D2	D0	ΔTHT	ΔTRT4	ΔTRT2	ΔTRT0
1	S					ZR	NB	NB	NB
2	L	S				ZR	PB	PB	PB
3	L	M				PB	ZR	ZR	ZR
4	M	B				PB	NB	NB	NB
5	M	M				PS	NS	NS	NS
6	M	S	B			ZR	PB	NS	NS
7	M	S	M			ZR	PS	ZR	ZR
8	M	S	S	B		ZR	ZR	PB	NS
9	M	S	S	M		ZR	ZR	PS	ZR
10	M	S	S	S	B	ZR	ZR	ZR	PB
11	M	S	S	S	M	ZR	ZR	ZR	PS
12	M	S	S	S	S	ZR	ZR	ZR	ZR

sion of the significant portion of the lower class messages is not allowed. FNPM1 has only two rules shown in Table 2 which essentially increase THT while decreasing TRT's. When one of the conditions is met, the switch activates FNPM2 which is identical to the TCT-based FNPM to improve lower priority classes. In order to detect how much the lower priority messages suffer, an additional input called relative throughput (RT) has been defined. RT is the sum of ratios of the amount of message transmission (Ti) to the amount of messages entering the network (Gi) for lower three priority classes. That is,

$$RT=\frac{T_4}{G_4}+\frac{T_2}{G_2}+\frac{T_0}{G_0}$$

where T_i and G_i denote the number of bits transmitted and the number of bits entering the network system, respectively, for the priority class i during a given period of time. When all the lower priority messages are allowed to be transmitted, RT equals to 3. The switch operates on the crisp

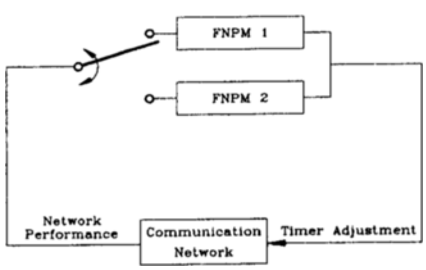


Fig. 5 Configuration of the switch-type FNPM

Table 2 Fuzzy rules of FNPM1 for the switch-type FNPM

	D6	ΔTHT	ΔTRT4	ΔTRT2	ΔTRT0
1	B	PB	NB	NB	NB
2	M	PS	NS	NS	NS

Table 3 Switch operating condition

D6	RT	FNPM
Not satisfied	>2	1
Not satisfied	≤2	2
Satisfied	≤2	2
Satisfied	>2	2

value of D_6 and RT in order to have more decisive action to satisfy the threshold for D_6. Table 3 shows which rule base is used depending on the value of D_6 and RT.

All three types of FNPM's are implemented by using Mamdani's minimum operation rule and the results are inferred by Zadeh's sup-min operation. The inputs to the fuzzy performance manager are entered as fuzzy singletons where the membership function has zero value everywhere except at a single point. The inferred results are converted into crisp values using the center of area method.

4. Evaluation of Fuzzy Network Performance Managers

This section presents the evaluation results of the fuzzy performance management for a 10 megabit per second (Mbps) IEEE 802.4 token bus network with 10 stations. A total of fifty one independent simulation runs are performed for each type of FNPM. At the end of each iteration, the necessary statistics are passed to the fuzzy performance management that will generate new timer values for the next iteration.

The simulation experiment is performed with the total network traffic equivalent to 80% of the network capacity. Priority classes 6, 4, 2 and 0 carry 10%, 20%, 20% and 30% of the total network capacity, respectively. The related simulation parameters are summarized in Table 4. It is assumed that the network is required to maintain the average data latency under 2300, 6000, 10000 and 20000 μsec for priority classes 6, 4, 2 and 0, respectively. Table 5 shows the values for the membership functions of average data latencies along with the initial timer settings. In order to maintain the order of priority class, TRT's are kept to satisfy $TRT4 \geq TRT2 \geq TRT0 > TCT_{min}$ where TCT_{min} is the minimum token circulation time which equals to the time for a token pass multiplied by the number of stations (120 μsec in our simulation). Also, THT is kept under a certain limit (1000 μsec) in order to avoid unnecessarily large values.

Table 4 Simulation parameters

	Priority6	Priority4	Priority2	Priority0
Number of stations	10			
Simulation time(μsec)	3.0×10^7			
Number of queues per station	4			
Queue capacity of each priority	10	10	10	10
Assigned traffic(%)	10	20	20	30
Average message generation interval(μsec)	50000	50000	100000	200000
Type of probability distribution for message length	Exponential			
Type of probability distribution for message generation interval	Exponential			

Table 5 Initial conditions for simulation experiments

		Priority6	Priority4	Priority2	Priority0
Initial timer(μsec)		150	3500	3500	3500
Fuzzy input region(μsec)	D_i_SM	2300	6000	10000	20000
	D_i_MD	2800	7000	11500	22000
	D_i_BG	3300	8000	13000	24000

4.1 Evaluation of the basic FNPM

Figure 6 shows the average data latencies for four priority classes during the simulation experiment. In the figure, the required upper limit for the average is represented by a solid horizontal line for each priority. At the beginning of the experiment, no priority class can satisfy its limit. As the experiment proceeds, it can be observed that D_4 improves and D_6 deteriorates while the rest remains more or less the same. The timer values during the experiment are shown in Fig. 7 where THT increases up to its limit and TRT's

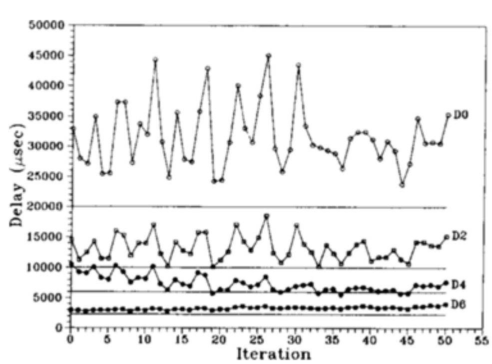


Fig. 6 Average data latencies from the simulation experiment with the basic FNPM

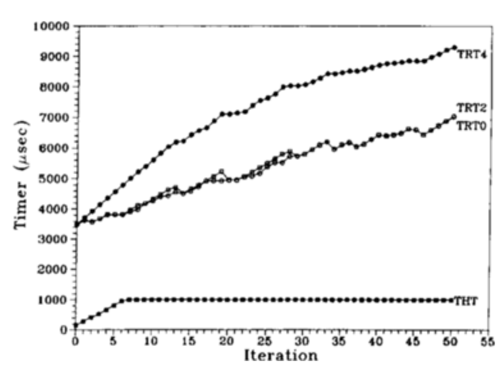


Fig. 7 Timer settings from the simulation experiment with the basic FNPM

keep on increasing. Even while THT is increasing, there is no improvement in D_6 because other TRT's are increasing at the same time. After THT reaches its limit, D_6 worsens as TRT's increase. This is because the rules tend to have less emphasis on the class 6 and to adjust TRT's relatively to each other.

4.2 Evaluation of the TCT-based FNPM

From Fig. 8, it can be seen that D_6 and D_4 steadily improve while D_2 and D_0 remains almost unchanged. Compared with Fig. 6, D_6 is quite better than that with the basic FNPM even though D_6 still stays above the threshold. Figure 9 shows the changes in the timer settings. At the beginning, THT is increased while other timers are reduced because D_6 is close to Big and Medium (rules 4 and 5 in Table 1). When D_6 become closer to Small (iteration 8), TRT4 is the first timer to get a chance to increase itself. From that point on, D_4 shows a decreasing trend while the network does not have any remaining capacity to improve D_2 and D_0. It clearly demonstrates that it is more effective to reduce TRT's than to increase THT in order to improve D_6. Overall, the TCT-based FNPM tends to give the priority class 6 more privilege than the basic FNPM in adjusting timers.

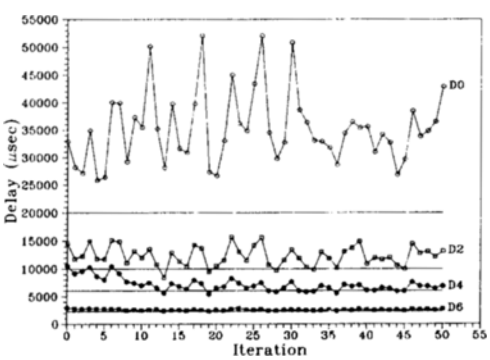


Fig. 8 Average data latencies from the simulation experiment with the TCT-based FNPM

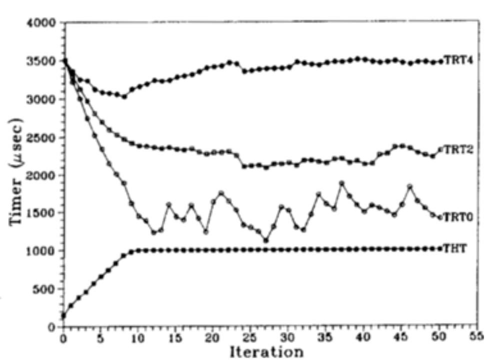


Fig. 9 Timer settings from the simulation experiment with the TCT-based FNPM

4.3 Evaluation of the switch-type FNPM

The simulation experiment with the switch-type FNPM shows that D_6 is reduced very close to or sometimes under the required level. Figures 10 and 11 show the trend of average data latencies and timer settings, respectively. The experiments start with D_6 over the required level, which triggers FNPM1 to increase THT while reducing TRT's. This situation continues until the 13th iteration where D_6 falls below the threshold for the first time. At this iteration, FNPM2 takes over to increase TRT0 while holding two other TRT's steady. After this, one of two rule bases is selected depending on the value of D_6. The relative throughput (RT) does not influence the choice of the rule base because it stays at the value of 3, which indicates no rejected messages. The reason for this phenomenon is that TRT's are forced to be greater than its minimum and that the queue capacity is set to 10 which is quite large enough to accommodate messages. As the experiment continues, D_6 and D_4 decrease remarkably while D_2 shows a seemingly decreasing trend. From Fig. 11, it can be observed that the values

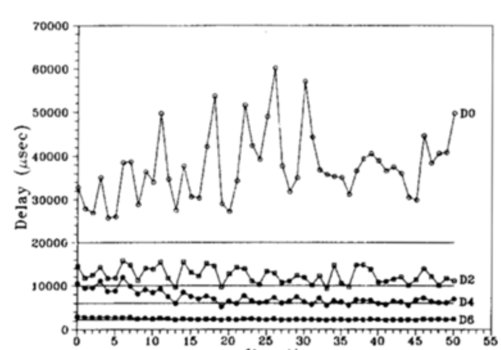


Fig. 10 Average data latencies from the simulation experiment with the switch-type FNPM

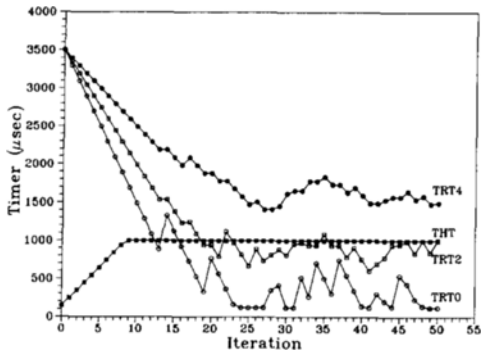


Fig. 11 Timer settings from the simulation experiment with the switch-type FNPM

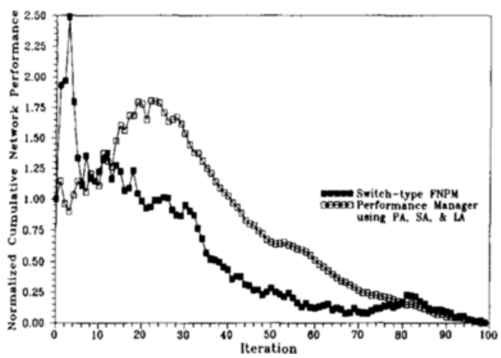


Fig. 12 Comparison of the cumulative network performance indexes with the switch-type FNPM and the performance manager using PA, SA and LA

for TRT's are much lower than those obtained with the TCT-based FNPM. These lower values help the priority class 6 satisfy the required data latency threshold.

Figure 12 shows the trends of the normalized network performance index with the switch-type FNPM and the performance management procedure developed in the previous research. The performance index is formulated as an average penalty per message where the penalty for the j-th message of the priority class i, $F_i(\delta_i(j))$, is defined by the function of its data latency, $\delta_i(j)$. In the equation below, θ_i denotes the threshold for the zero penalty and b_i represents the saturation band.

$$F_i(\delta_i(j)) = \begin{cases} 0 & \text{if } \delta_i(j) \leq \theta_i \\ (\delta_i(j) - \theta_i)^2 & \text{if } \theta_i \leq \delta_i(j) \leq \theta_i + b_i \\ b_i^2 & \text{if } \delta_i(j) \geq \theta_i + b_i \end{cases}$$

This performance index penalizes only the messages whose data latencies exceed the threshold by the square of the exceeding time amount until it reaches the saturation band. From Fig. 12 where the cumulative performance indexes are shown, it can be observed that the performance obtained by the switch-type FNPM is quite comparable to the one obtained in the previous research. It should be noted that the comparable performance is obtained with the much simpler and more intuitive fuzzy rules.

5. Conclusions and Future Research

This paper presents the development and evaluation of three Fuzzy Network Performance Managers for IEEE 802.4 token bus networks. The major conclusions derived from this research are delineated below.

- The basic FNPM is found to be the least biased over the priority classes when compared to the other two types. This may be suitable for less demanding data latency requirements under light to medium traffic. On the other hand, it may defeat the purpose of the priority mechanism for demanding requirements. In addition, the rule base contains too many rules for practical implementation.
- The TCT-based FNPM is found to be biased for the priority class 6 in adjusting timers. The primary objective of this type is to satisfy the priority class 6 requirement. It is also capable of adjusting TRT's relative to the current token circulation time. The simulation experiment shows that the TCT-based FNPM performs well only with 12 rules.
- The switch-type FNPM consists of two rule bases to place more emphasis on the priority class 6 in adjusting timers. It takes more decisive action to reduce TRT's until D_6 is under the required level. The switch selects one of the rule bases based on the crisp values of D_6 and RT. From the simulation experiments, it is found to be the most biased toward the priority class 6 and it is the only type that is able to reduce D_6 below the required level.
- These FNPM's can be selectively used

according to the network traffic and the policy for performance management. For example, the TCT-based FNPM can be employed for the traffic without messages under stringent real-time requirements while the switch-type FNPM may be suitable with the traffic with real-time messages.

- For all types of FNPM, it is not necessary to identify the statistical characteristics of the network traffic for performance management. In addition, any revision on the protocol is not needed.
- These FNPM's are able to improve the network performance without having very complex techniques such as perturbation analysis, stochastic approximation and learning automata.

In order to further improve this fuzzy performance management, research on the following issues is needed.

- It may be required to consider the amount of the network traffic. By doing so, the fuzzy performance management may be able to apply more appropriate rules to the current network traffic.
- The fuzzy performance management needs to check if the user's demand on the data latency is achievable. If not, it should be able to pursue some second best goals.
- It is needed to have some systematic way to tune the membership functions. The promising techniques include neural network and genetic algorithm.

References

ANSI/IEEE, 1990, *ANSI/IEEE Standard 802.4 : Token-Passing Bus Access Method and Physical Layer Specifications.*

Available through MAP/TOP Users Group, 1988, *Technical and Office Protocols(TOP) 3.0 Implementation Release,* One SME Drive, P. O. Box 930, Dearborn, MI 48121.

Available through MAP/TOP Users Group, 1993. *Manufacturing Automation Protocol(MAP) 3.0 Implementation Release,* One SME Drive. P. O. Box 930, Dearborn, MI 48121.

Bertsekas, D. and Gallager, R., 1987, *Data Networks*, Prentice Hall.

IEEE, 1992, *IEEE Standard 802.1 B : LAN/MAN Management.*

Klerer, S. M., 1988, "The OSI management Architecture: An Overview," *IEEE Network*, Vol. 2, No. 2, pp. 20~29.

Lee, C. C., 1990, "Fuzzy Logic in Control Systems: Fuzzy Logic Controller-Part I and II," *IEEE Transactions on Systems, Man and Cybernetics*, Vol. SMC-20, No. 2, pp. 404~435.

Lee, S. and Ray, A., 1993, "Performance Management of Multiple Access Communication Networks," *IEEE Journal on Selected Areas in Communication*, Vol. 11, No. 9, pp. 1426~1437.

Pegden, C. D., Shannon, R. E. and Sadowski, R. P., 1990, *Introduction to Simulation Using SIMAN*, McGraw-Hill.

Ray, A., 1988, "Networking for Computer-Integrated Manufacturing," *IEEE Network*, Vol. 2, No. 3, pp. 40~47.

SAE, 1987, *Linear Token Passing Multiplexed Data Bus Standard, Version 3.0*

Saydam, T. and Sethi, A. S., 1987, "Token Bus/Ring Local Area Network Management Concepts and Architecture," *IEEE INFOCOM*, pp. 988~993.

Thompson, D. M., 1986, "LAN Management Standards-Architecture and Protocols," *IEEE INFOCOM*, pp. 355~363.

Zadeh, L., 1973, "Outline of a New Approach to the Analysis of Complex Systems and Decision Process," *IEEE Transactions on Systems, Man and Cybernetics*, Vol. SMC-3, No. 1, pp. 28~44.