On-line Fuzzy Performance Management of Profibus Networks

Kyung Chang Lee*, Song Kang*, Dong Wook Kim**, and Suk Lee***

* Department of mechanical & intelligent systems engineering, Pusan National University, Pusan, Korea

** RAS team, DASAN Internet Co. LTD

*** School of mechanical engineering, Pusan National University, Pusan, Korea

ABSTRACT

Most networks for automation are tuned to an expected traffic condition at their design stage. During their actual operations, however, the networks experience considerable changes in traffic from time to time. These traffic changes caused by common events like machine failure and production schedule change may adversely affect the network performance and, in turn, the performance of the connected devices.

This paper presents experimental results for performance management of a Profibus-FMS network. The performance management aims to maintain a uniform level of network performance at all stations under changing network traffic. The performance management algorithm monitors the performance of individual network stations and commands the stations to change their timer values in order to have comparable performance at all stations. In order to determine the amount of timer change, the algorithm employs a set of fuzzy rules. This algorithm has been evaluated on a Profibus network.

Keywords : Networking for manufacturing automation, Profibus-FMS protocol, performance management, fuzzy logic, target rotation timer(TRT), fuzzy network performance manager(FNPM)

1. Introduction

Networking serves as communication links between islands of automation for further improvement in productivity in Computer Integrated Manufacturing (CIM). These networks are capable of interconnecting various standard-conforming devices from different vendors, which allows a system designer to be flexible on his/her initial design and future reconfigurations. This capability results in numerous advantages of networking for CIM and these include evolutionary system growth, better utilization of system resources, and improved reliability. Due to these benefits, 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 a highly integrated manufacturing system. [1,2]

According to the area of application, a computer network should be tailored at the design stage by selecting appropriate network protocols and their parameters since requirements imposed on a network may widely vary. [1,2,3] Even after design and installation of a network, several groups of functions are required to adjust the network so that the 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 manufacturing system.

The responsibility of adapting network belongs to network management that 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 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.

In order to carry out their roles successfully, the networks require an ability to adjust their protocol parameters to maintain a certain level of performance. The protocol parameters include timer values and queue capacities, and the network performance is often expressed in terms of network delay and throughput. This ability belongs to the performance management of networks, and has become increasingly important because a network interconnects numerous devices that are very diverse in their communication requirements. One can easily imagine that a network on a common factory floor connects several PLCs, robots, conveyors, and various sensors, all of which have different transmission periods and maximum allowable delays. To make matters even more complicated, the characteristics of the network traffic are random and dynamic in nature due to common events like arrival of new production orders and failure in system components. [3,4,5]

As a case in point, Figure 1 shows the response of a motor that is remotely controlled via a fieldbus network, That is, the motor and its controller don't have any direct connection between them. Instead, they communicate with each other via the network. The encoder signal and the control command are sent through the network that is shared with other devices. The figure shows the motor speed under an identical traffic condition except different values for a protocol parameter (a kind of timer). The dotted curve represents the motor's response when the motor and the controller are directly connected. As can be seen in the figure, the responses obtained with the network are generally inferior to that with direct connection. Especially, it shows that an inappropriate choice for the protocol parameter can make the system unstable even under the same traffic. This example shows how important it is to make proper choices for protocol parameters.

This paper presents the development of an on-line fuzzy performance management procedure for Profibus-FMS networks. The management procedure is based on fuzzy rules to adjust a timer setting. The algorithm is designed to maintain uniform performance of all the stations even if the individual traffic loads are different. The developed procedure has been evaluated through experiments on the network that consists of six stations. The results showed that the management procedure was successfully adjust the timer to maintain uniform performance.

This paper is organized into five sections including this introduction. Section 2 gives a brief summary of Profibus-FMS protocols. The management algorithm and the network used for the development are described in Section 3. Section 4 describes the results of the experiment with fuzzy network performance manager(FNPM). Finally, conclusions are presented in Section 5.

2. Profibus-FMS Protocols

Figure 2 shows the architecture of Profibus-FMS. It is designed to have only three layers, i.e., physical, data link, and application layers, out of seven layers of the ISO OSI model. This is because the fieldbus network should support real-time communications among the

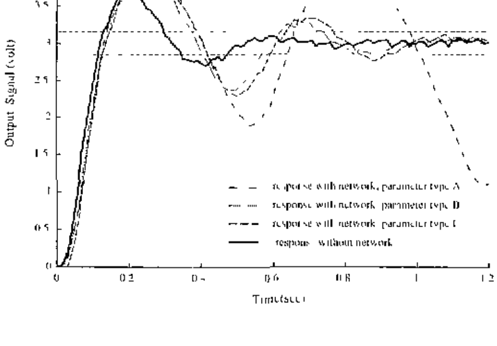

Fig. 1 Response of a Remotely Controlled Motor

devices on shop floors. Along with the three layers, it has an additional user layer and a management layer providing management services for all the layers. [6,7,8]

2.1 Physical and Data Link Layers

The physical layer is responsible for converting the data to be transmitted into transmission signals, propagating the signals to the receiver, and converting the signals back to the original data. Profibus employs the bus topology and accommodates up to 32 nodes or stations in a segment. If necessary, up to three repeaters can be used to connect four segments to interconnect maximum of 127 stations. The maximum transmission speed is 500 kbps (kilobits per second).

The major functions of data link layer are medium access control (MAC) and logical link control (LLC). The logical link control is responsible for establishing and terminating logical communication links among the stations and making the communication link reliable for the upper layer by using cyclic redundancy check (CRC). The medium access control is a set of rules to share the network among the stations. The MAC layer employs a combination of token passing and polling. The network consists of two types of stations: master and slave stations. Among the master stations, the right to use the medium is explicitly controlled by a special bit pattern called a token. The right to transmit a message is given to a master station when it receives the token. The token is passed from a master 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 of mater stations. On the other hand, a slave station is allowed to transmit only when it is commanded to do so (polled) by the master station holding the token.

As mentioned above, a master 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 and priority class of waiting messages and the several timer values. The MAC of Profibus supports two classes of message priority: high and low, by using three different time variables called Target Rotation Time (TRT), Real Rotation Time (RRT), and Token Holding Time (THT) for each master station. TRT is an expected value of the time interval for the token to circulate the whole logical ring while RRT is the actual time period between the two consecutive token reception by the station. THT is used to keep track of the time while the station is holding the token. As shown in Figure 3, when a master station receives the token, it computes the RRT and stores the difference between TRT and RRT (TRT-RRT) in THT. If RRT is greater than TRT, the value of zero is loaded into THT. Then the station is allowed to transmit one, if any, high priority message while THT is started to count down at the beginning of the transmission. After the transmission is over, more transmissions of high messages are allowed only if THT has not expired. If THT still has some time left even after all the high messages are transmitted, then the low messages can be sent until THT expires. When THT

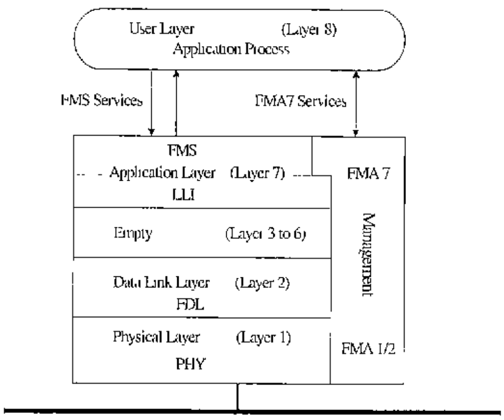

Fig. 2 Profibus-FMS Protocol Architecture

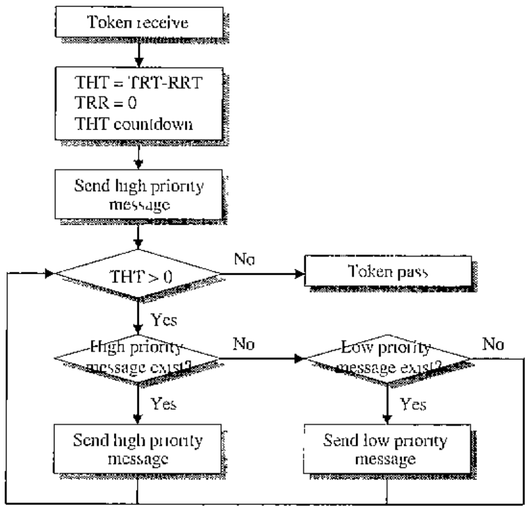

Fig. 3 Flowchart of the Priority Mechanism

expires or no messages are waiting, the token is passed to the next master station in t he logical ring sequence.

This priority mechanism allocates at least a certain fixed amount of time for high priority messages while additional time for high messages and the time for the low messages are affected by the value of TRT and the RRT. In fact, the RRT is directly influenced by the network traffic because the RRT will increase as the traffic increases. Also, with a given traffic, the longer the TRT is, the more messages can be sent at one time.

2.2 Application Layer

The application layer is divided into two sublayers: Fieldbus Message Specification (FMS) and Lower Layer Interface (LLI). FMS is similar to Manufacturing Message Specification (MMS) of Manufacturing Automation Protocol (MAP) and defines communication services along with related objects and communication models. It provides an environment in which the network user can access the communication resources by offering services like Virtual Field Device (VFD) Support, Object Dictionary (OD) Management, Context Management, Domain Management, Program Invocation Management, and Event Management. The VFD is an abstract entity that is related to some communication functions. An actual device communicating on the network should have at least one VFD in order to represent itself on the network. Each VFD has an OD that is a data structure describing the object's type, structure, and attributes. The LLI links the FMS to the data link layer and is responsible for flow control and context management. It provides the FMS and Fieldbus Management Layer 7 (FMA7) with Communication Relations (CR) to identify a logical communication link among distributed applications.

2.3 Fieldbus Management Layer 7

FMA7 plays an important role in configuring and maintaining the whole network. It is responsible for context management, fault management, and configuration management. In particular, by using services like Set-Busparameter, the configuration management provides a means to change some bus parameters while the bus is in operation.

3. Fieldbus Testbed and Fuzzy Network Performance Manager

3.1 Fieldbus Testbed

Figure 4 shows a Profibus-FMS network used to develop and evaluate the performance management algorithms. It consists of six nodes out of which stations 1 to 4 participate in normal communication. More specifically, stations 1 and 2 exchange data with each other by using Information Report service provided by the FMS while stations 3 and 4 do the same for themselves. An application process resides in each of stations 1 to 4 generating random traffic according to the user input. The manager station shown in Figure 4 hosts the management algorithm that collects performance reports, determine the amount of parameter changes, and request parameter changes to stations 1 to 4 These five stations are configured as master stations The Bus Monitor is a node that is solely for monitoring the network traffic without transmitting any frames. The hardware of each node is an IBM-compatible Pentium PC with a Profibus interface card. The Profibus interface card is responsible for transmitting and receiving frames. The card is based on a Siemens SAB-C165 microprocessor along with two communication ASICs (Application Specific Integrated Circuit): ASPC2 and SPC3. [9] The microprocessor performs the functions belonging to the application layer while the communication ASICs are responsible for the physical and data link layer functions. The Profibus interface card has 512Kbyte flash memory for its firmware and SRAM for storing the communication objects. Also, 16 Kbyte

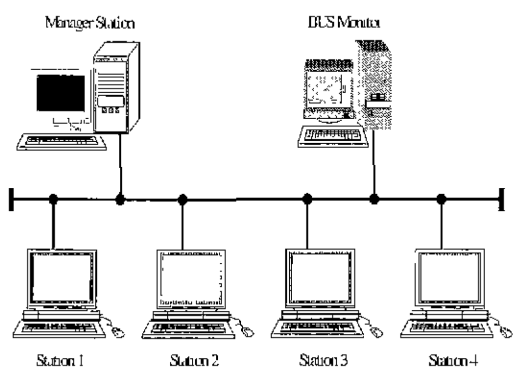

Fig. 4 Fieldbus Testbed

dual-port RAM is used to share data between the communication processor (SAB-C165) and the PC's processors (Pentium).

3.2 Structure of Communication Software

The communication software for the stations was developed using Visual C ++ 6.0 and Application Program Interface (API) provided by Softing [9] under Windows NT environment. Windows NT was used to utilize its multi-thread capability because the communication software consists of four processes as shown in Figure 5. These four processes are "Initialize parameter," "Wait user input," "Check reception buffer," and "Check message period." The following subsections describe these processes in more details.

3.2.1 "Initialize parameter" Process

It is responsible for initializing various parameters including bus parameters, Communication Relationship List (CRL), and Object Dictionary (OD) for Profibus-FMS service. The bus parameters include those parameters related to the physical and data link layers such as data rate, bit timing, and TRT. These parameters are initialized by using Set Bus Parameter service of FMA1/2. The Communication Relation (CR) is a logical link between distributed applications and CRL is a list of those links used in a station. CRL should have at lease one CR in order to set up the LLI, and additional CRs can be defined as needed. In this research, three additional CRs are defined for point-to-point data transmission, broadcast of performance report, and broadcast of timer values and synchronization signal, respectively. OD contains the structures of the data transmitted on the network. There are three objects defined for FMS-write service, TRT update, and synchronization.

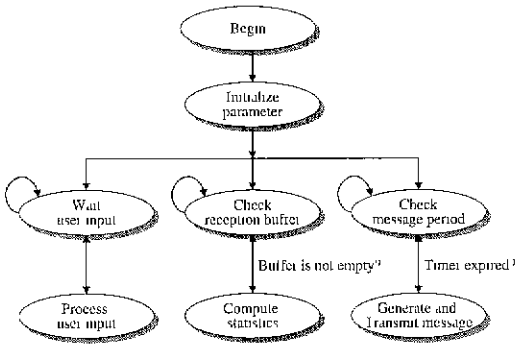

Fig. 5 Structure of Communication Software

3.2.2 "Wait User Input" Process

This process handles the mouse events caused by the user. The communication software needs the user's input to initiate the communication with others using FMS Initiate service. Also, the user can execute Information Report service and terminated the communication via this "Wait User Input" process.

3.2.3 "Check Reception Buffer" Process

"Check Reception Buffer" process is responsible for periodically checking the reception buffer of the Profibus interface card for any newly received frames. It also calls an appropriate routine to process the received frames. There are three types of frames: message frame, synchronization frame, and TRT frame. The message frames are used for artificially generating network traffic. Therefore, the receiving station is not concerned with the message frame's contents, but it simply updates its counter for the number of received messages and calculates the frame's delay to compute the network performance. The synchronization frame is used to synchronize stations' clocks that are used to calculate the message frame's delay. When the synchronization frame is received, the station immediately resets its counter to zero. Finally, the TRT frame contains the new value for TRT When received, the station uses FMA7 Set Value Local service to update its TRT.

3.2.4 "Check Message Period" Process

This process generates message frames at a given interval determined by the random distribution specified by the user input. When an internal timer expires, a new message frame is generated and put into the transmission buffer. Also, this process changes its traffic characteristics according to the user input to emulate changes in the network traffic.

3.3 Fuzzy Network Performance Manager

Many instrumentation and control applications require periodic transmission of sensor data and control commands while other applications such as file transfer generate frames at a random rate. Among the various application supported by the network, real-time applications such as instrumentation and control are more sensitive to the time period to deliver the frame to the destination node (end-to-end data latency) than other applications. [10] This end-to-end data latency is determined not only by the amount of overall traffic on the network but also by the local traffic at each node. It is necessary to adjust bus parameters so that the data latency for real-time applications is maintained below an acceptable level.

In this research, the objective of the performance management is to adjust TRT so that end-to-end data latency for high priority at each station becomes more or less the same. This goal is to spread the effect of the traffic increase at a station to others in order to avoid excessive degradation of performance at a given station. The procedure for performance management is as follows.

1. The manager station broadcasts to stations 1 to 4 a frame in order to synchronize reference clocks used to measure data latency.
2. Each station initializes its clock, and then begins to transmit its frames as necessary.
3. While sending and receiving frames, stations accumulate data to calculate average data latency and throughput (ratio of the data amount transmitted to that generated).
4. At a predefined period, each station transmits the average data latency and throughput to the manager station.
5. After collecting all the performance repots, the manager station computes the new values of TRT by using an algorithm called Fuzzy Network Performance Manager.
6. These new values are transmitted to the stations.

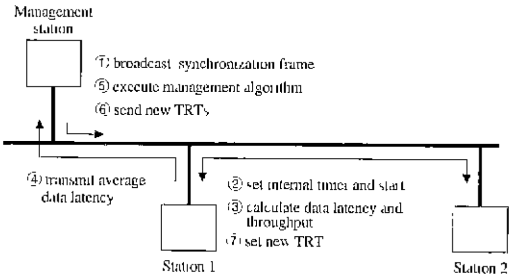

Fig. 6 Procedure for Performance Management System

7. Each station adjusts its TRT to the new value, reinitializes its data collection variables, and resumes its normal operation.

The above procedure is summarized in Figure 6.

In order to determine the new values for TRTs, the principle of fuzzy logic has been selected because the analytical relationship between data latency and TRT is known only for very limited cases. The FNPM has 49 rules with two inputs and one output. The inputs are the deviation of the average data latency of a station from that of the whole network, and the amount of change in the deviation from that of the previous observation period. That is, if we let d_{ij} be the average data latency of station i for the observation period j and n be the number of stations,

$$\bar{d}_j = \frac{1}{n}\sum_i d_{ij}$$

$$e_{ij} = \bar{d}_j - d_{ij}$$

$$e'_{ij} = e_{ij} - e_{i\,j-1}$$

where $\bar{d}_j$ is the average data latency for all stations; e_{ij} is the difference between global and local averages for data latency, and e'_{ij} is the trend of e_{ij} over two consecutive observation periods. Based on these inputs, the fuzzy rules generate the amount of change in TRT and the new value of TRT is sent to each station. Figure 7 depicts the structure of FNPM with the subscript j omitted.

For both input and output variables, seven linguistic variables have been defined. The variables are Negative Big (NB), Negative Medium (NM), Negative Small (NS), Zero (ZR), Positive Small (PS), Positive Medium (PM), and Positive Big (PB). The membership functions are shown in Figure 8 where the membership functions have

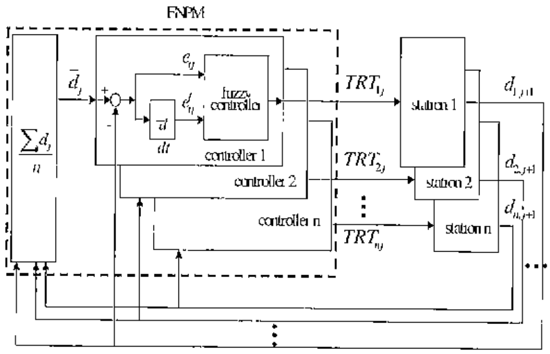

Fig. 7 Structure of FNPM

finer resolution near zero value. In particular, the membership function for ZR for ΔD_{ij} has an interval where the membership value is equal to one in order to prevent TRT from oscillating when the performance level becomes uniform over the network. Table 1 shows the fuzzy rules extracted from the simulation runs and experiments. In order for faster execution of fuzzy rules, inputs are treated as fuzzy singletons. For the inference method, Mamdani's inference method has been used along with Simplified Center of Gravity (SCG) method for defuzzification. [11]

4. Experimental Results and Discussion

The Fuzzy Network Performance Manager has been evaluated on the fieldbus testbed shown in Figure 4. The network operates at the transmission speed of 500 kbps. For the initial network traffic, each station is programmed to generate a message at a random period that is uniformly distributed from 0 to 100 millisecond

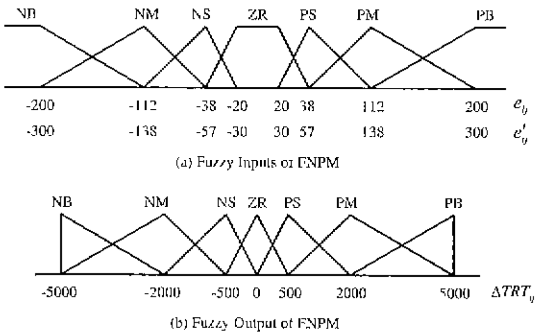

Fig. 8 Membership Functions of FNPM

Table 1 Fuzzy Rules for FNPM

e_{ij} \ e'_{ij}	NB	NM	NS	ZR	PS	PM	PB
NB	PB	PB	PM	PS	PS	PS	ZR
NM	PB	PM	PS	PS	PS	ZR	NS
NS	PM	PS	PS	PS	ZR	NS	NS
ZR	PS	PS	PS	ZR	NS	NS	NS
PS	PS	PS	ZR	NS	NS	NS	NM
PM	PS	ZR	NS	NS	NS	NM	NB
PB	ZR	NS	NS	NS	NM	NB	NB

(msec). The length of a message is 200 bytes. This type of traffic is to emulate the mixture of real-time and non-real-time applications. This traffic is equivalent to 25.6 percent of the network capacity.

The network performance is collected at every 30 seconds in order to accumulate enough data to filter out randomness and other transient behaviors. One minute (two iterations) after the experiment started, the traffic of stations 3 and 4 are programmed to increase to uniformly distributed message generation period over 0 to 40 msec. This is about 1.75 times higher than the initial traffic. This kind of traffic change can be very likely on the network for manufacturing automation because a new production order or a machine failure will require relatively large file transfers, which increases the network traffic drastically.

Figure 9 shows the average data latency of high priority messages at each station when no management action is taken. As expected, the averages of stations 3 and 4 are significantly increased. Station 3 experiences the average data latency around 800 msec while Station 4's average reaches close to 500 msec. (The reason for the difference between two stations is not exactly known, but it seems to be the hardware performance of the stations.) This level of data latency may be catastrophic to real-time applications residing on both stations. While stations 3 and 4 suffers from the huge increase in data latency, stations 1 and 2 enjoy much better and probably tolerable data latency. Furthermore, the throughput of stations 3 and 4 suffers greatly dropping from 100 to below 80 % (Figure 10). This means that two out of ten messages are deleted before being transmitted because of large delays and finite buffer capacity.

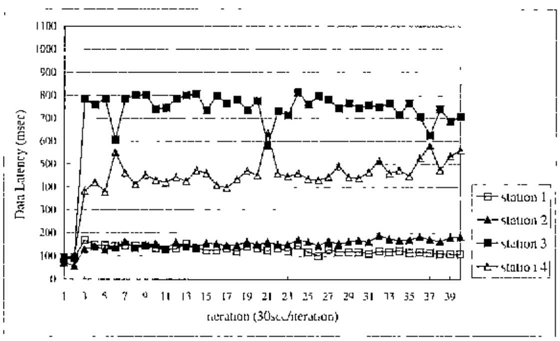

Fig. 9 Average Data Latency of High Priority Messages without Performance Management

Figure 11 shows the values of TRT at each station during the experiment with the FNPM. TRTs for stations 3 and 4 are increased rapidly from the initial value of 50000 bit time (1 bit time equals to 2 μsec). Within five iterations after the traffic increase, TRT reaches around 58000 bit time while TRT for stations 1 and 2 is reduced to 48000 bit time. This is the result of FNPM's action in order to decrease average data latency of stations 3 and 4 by increasing their TRTs and decreasing TRTs at stations 1 and 2.

Figures 12 and 13 show the average data latency of high priority messages. Due to the changes in TRT, the averages of stations 1 and 2 increase much more than the previous case. On the other hand, those of stations 3 and 4 recover quite rapidly after initial deterioration. Comparing the two figures, one can observe four curves are overlapping with each other over 300 to 400 msec range. Throughputs of stations are shown in Figure 14. As with data latency, the throughputs of stations 3 and 4 suffers initially, but recovers very well close to 95 %.

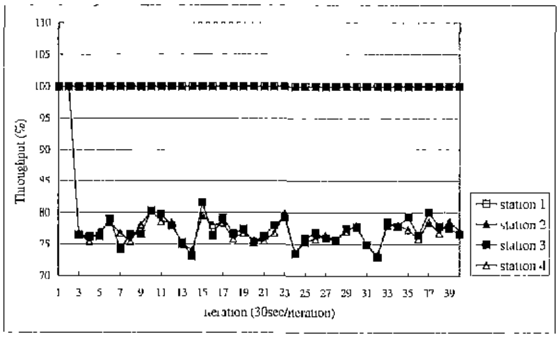

Fig. 10 Throughput of High Priority Messages without Performance Management

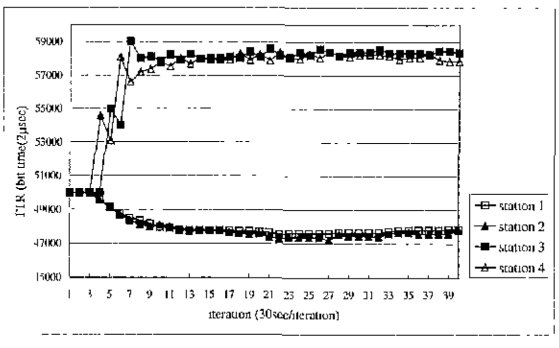

Fig. 11 TRT Valued Adjusted by FNPM

From this experiment, it can be said that the FNPM is effective in adjusting TRT values in order to recover from the unacceptable performance level caused by a local traffic increase. In fact, the FNPM make the stations with light traffic sacrifice some of their transmission time for the stations with heavy traffic by reducing TRT for the former and increasing for the latter.

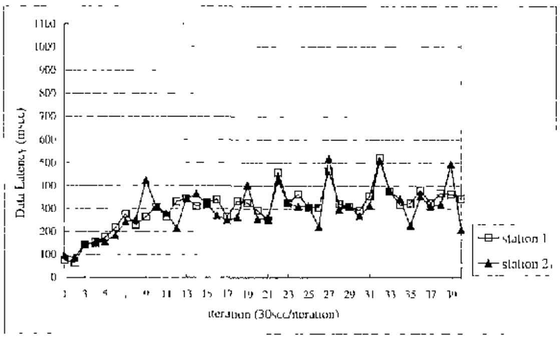

Fig. 12 Data Latency of High Priority Messages at Station 1 & 2 with FNPM

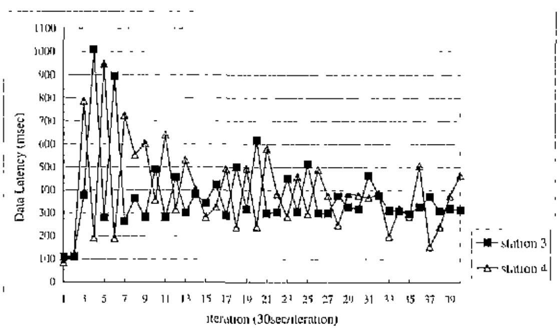

Fig. 13 Data Latency of High Priority Messages at Station 3 and 4 with FNPM

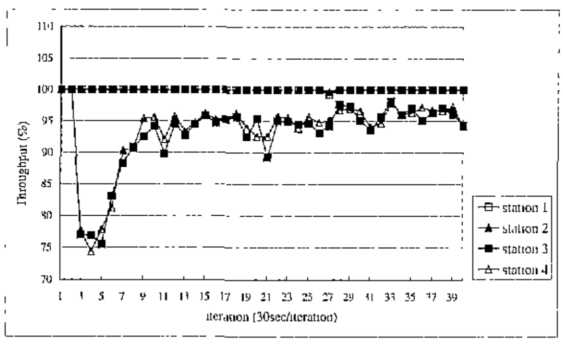

Fig. 14 Throughput of High Priority Messages with FNPM

5. Conclusions

This paper presents the development and evaluation of the on-line performance management of Profibus-FMS networks by using fuzzy logic. The major conclusions derived from this research are delineated below.

- Adjustment of TRT can be an effective method for allocating the network capacity to stations with different traffic loads.
- Fuzzy rules extracted from the simulations and experiments are proved to be successful in maintaining a uniform level performance in spite of different traffic for network stations.
- FMA7 services are the powerful tools for on-line performance management where the bus parameters can be adjusted without interrupting normal operations.

References

1. A. Ray, "Networking for computer-integrated manufacturing," IEEE Network, Vol. 2, No. 3, pp. 40-47, 1988.
2. J. P. Thomesse, "Fieldbuses and interoperability," Control Engineering Practice 7, pp. 81-94, 1999.
3. W. L. Genter and K. S. Vastola, "Performance of the token bus for time critical messages in a manufacturing environment," American Control Conference, pp. 534-539, 1989.
4. S. Lee, A. Ray, Performance Management of Manufacturing System Networks, International Journal of Flexible Manufacturing Systems, Vol.7, No.1, March, 1995.
5. S. H. Lee, J. W. Son, S. Lee, Performance Management of Token Bus Networks for Computer Integrated Manufacturing, KSME Journal, Vol. 10, No. 3, pp. 321-331, 1996.
6. Klaus Bender, *Profibus-The Fieldbus for Industrial Automation*, Prentice-Hall, 1993.
7. The Rapid Way to Profibus, 1997.
8. Profibus Specification - Normative Parts of Profibus-FMS, -DP, -PA according to the European Standard, EN 500 170, vol. 2, March, 1998.
9. Profibus Application Program Interface, user manual, version 5.2, rev.01, March, 1998.
10. H. C. Folts, "A Tutorial on the Open Systems Interconnection Reference Model," Open Systems Data Transfer, No. 1, pp. 2-21, June, 1982.
11. C. C. Lee, "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, 1990.