Transmission of a Scheduled Message Using a Foundation Fieldbus Protocol

Seung Ho Hong, *Member, IEEE*, and Sung Min Song

***Abstract*—In the last decade, process instrumentation field devices such as sensors and actuators that have a direct contact with industrial processes have become increasingly intelligent. An important property of such intelligent field devices is their ability to be networked. The Foundation fieldbus (FF) is one of the most widely used fieldbus protocols for process instrumentation and control systems applications. This paper presents a study that validates the effectiveness of a new scheduling method that satisfies the performance requirements for a scheduled message using an FF protocol. Our scheduling method generates a schedule list that records the scheduled starting time and the scheduled period of each scheduled message. Then, it allocates scheduled message traffic to a bandwidth-limited fieldbus medium. As part of this study, an experimental model of an FF network was developed. The experimental results from our model confirmed that our scheduling method restricts the delay of a scheduled message to a prespecified bound.**

***Index Terms*—Foundation fieldbus (FF), network management, scheduled message, system management, time constrained.**

NOMENCLATURE

N_s Number of nodes that generate a scheduled message on the medium.

L_s Transmission time of a scheduled message (in milliseconds).

ϕ_i Maximum allowable delay of a message generated by the function block (FB) at node i ($i = 1$ to N_s). A user of a Foundation fieldbus (FF)-based control system determines ϕ_i according to the requirements of the application systems.

T_i Message-generation period (FB macrocycle) at node i that generates a scheduled message ($i = 1$ to N_s).

t_i Message-generation start time (the start time of the FB schedule) at node i that generates a scheduled message ($i = 1$ to N_s).

S_i Link active scheduler (LAS) schedule period for node i that generates a scheduled message ($i = 1$ to N_s).

s_i LAS schedule start time for node i that generates a scheduled message ($i = 1$ to N_s).

γ Maximum number of scheduled messages that can be generated during T_1.

σ Transaction time for the LAS to execute a scheduled or an unscheduled message transmission.

A_l Time point at which the lth T_1 slot begins in T_{N_s} (see Fig. 3).

Manuscript received August 23, 2005; revised September 21, 2007. This work was supported by the Gyeonggi Regional Research Center under a grant from the Basic Research Program of the Ubiquitous Sensor Networks Research Center.

S. H. Hong is with the Department of Electrical Engineering and Computer Sciences, Hanyang University, Ansan 425-791, Korea (e-mail: shhong@hanyang.ac.kr).

S. M. Song is with the System Software Department, LG-Nortel Company, Ltd., Anyang 135-985, Korea.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIM.2007.910100

I. INTRODUCTION

DIGITAL data acquisition and control systems have been in use for more than 20 years. All the instruments and actuators that are necessary for controlling any industrial process today are intelligent (digital). Until a few years ago, the control center was intelligent, but the field devices were not. Today, however, both the control center and the field devices are intelligent. Communication with the field devices was once based on analog signals (e.g., 4–20 mA or 0–10 V), but this has changed with the progress of digital technology. Data in the control center are digitally created, and the same happens in intelligent field devices (IFDs), measuring instruments, or actuators built with digital technology. A modern measurement and control system fully integrates IFDs and control devices. The fieldbus is the information pipe that carries data among and between IFDs and control devices [1]–[3].

The FF [4] is one of the most widely used fieldbus protocols in process instrumentation and control systems applications. The user layer of the FF protocol stack is specifically designed to support the functional requirements of process instrumentation and control systems. The lower layers of the protocol stack are designed to support both periodic and aperiodic data transmission through scheduled and unscheduled messages, respectively.

Hong and Choi [5] reported a bandwidth-allocation scheme that can be applied to the data link layer (DLL) of an FF. This scheme satisfies the requirements for real-time communication of both periodic and aperiodic data while fully utilizing the bandwidth of a fieldbus medium. However, previous research considered only the data transfer properties of the FF DLL, and the practical issue of coordinating these properties with FB scheduling at the system management level was left as future work. In a practical process instrumentation and control system, the user must be able to implement bandwidth allocation at the system management level. To support our study, we developed an experimental model of an FF communication system that consists of IFDs that are typical of those used in actual process instrumentation and control industries. Using this experimental model, this study demonstrated that our bandwidth-allocation scheme for a scheduling service could be implemented at the

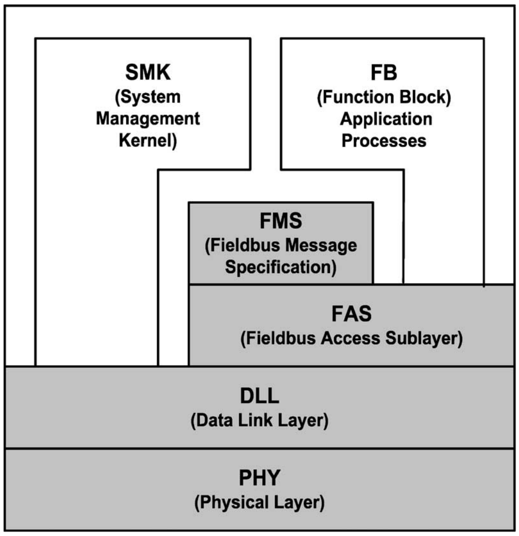

Fig. 1. Structure of the FF protocol stack.

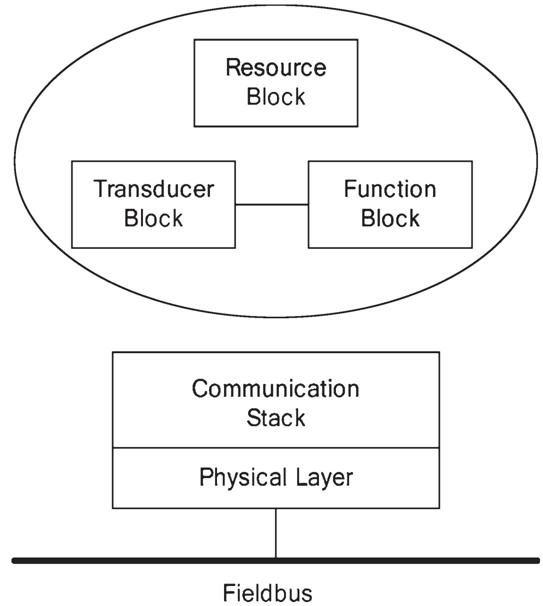

Fig. 2. Blocks in the user application of an FF.

system management level of an FF. This study also validated the effectiveness of our proposed scheduling method. The experimental results confirmed that our scheduling method for bandwidth allocation satisfies the delay requirements for scheduled messages.

This paper consists of six sections. Section II provides a brief description of the structure of the FF protocol stack. Section III discusses the bandwidth-allocation scheme that was previously introduced [5]. Section IV describes our experimental model and its implementation, and Section V presents our experimental results. Finally, Section VI presents our conclusions.

II. Structure of the FF Protocol Stack

An FF consists of a physical layer, a communication stack, and a user application. The communication stack is composed of a DLL and an application layer in accordance with the Open Systems Interconnection (OSI)-layered communication model [6]. The user application is not defined by the OSI model, although the FF has a specified user application model. Fig. 1 shows the outline structure of an FF protocol stack.

The physical layer [7] receives messages from the communication stack and converts them into physical signals that use the fieldbus transmission medium and vice versa. The physical layer supports the following three different data rates: 1) 31.25 kb/s; 2) 1 Mb/s; and 3) 2.5 Mb/s.

The DLL controls the transmission of messages using the fieldbus. It manages access to the fieldbus through a deterministic centralized bus scheduler called the LAS. The LAS has a cyclical list of transmit times for all the data buffers in all the IFDs that need to transmit data. When it is time for an IFD to send the data from its buffer, the LAS issues a compel data (CD) message to the IFD. When it receives the CD, the IFD broadcasts or publishes the data from its buffer to all the other IFDs on the fieldbus.

All the IFDs on the fieldbus are given a chance to send unscheduled messages between the scheduled messages. The LAS grants permission to an IFD to use the fieldbus by issuing it a pass token (PT) message. The list of all of the IFDs subject to responding to a PT is called the "live list." The LAS sends a PT message to each of the IFDs in the live list in a predefined order. When an IFD receives the PT, it is allowed to send messages until it has finished or until the maximum token hold time has expired, whichever comes first.

The fieldbus access sublayer (FAS) uses the scheduled and unscheduled features of the DLL to provide a service for the fieldbus message specification (FMS). The different types of FAS services are described by virtual communication relationships (VCRs). The are three different types of VCRs, which are listed as follows: 1) client/server; 2) report distribution; and 3) publisher/subscriber.

FMS services allow IFDs to send messages to each other across the fieldbus using a standard set of message formats. The FMS describes the communication services, message formats, and protocol behavior required to build the messages for the user application.

The FF defines a standard user application based on "blocks," which represent the different types of allocation functions. The types of blocks used in a user application include resource blocks, transducer blocks, and FBs (Fig. 2).

The resource block describes IFD characteristics such as device name, manufacturer, and serial number. FBs provide the control system behavior. The FF has defined sets of standard FBs, which include analog input (AI), analog output (AO), discrete input (DI), discrete output (DO), proportional-integral derivative, etc. The input and output parameters of FBs can be linked over the fieldbus. The execution of each FB is precisely scheduled. Transducer blocks decouple FBs from the local input/output functions that are required to read sensors and control output hardware.

FBs must execute at precisely defined intervals and in the proper sequence for correct instrumentation and control operation. The FF includes a system management kernel (SMK) to synchronize the execution of the FBs and the communication of the FB parameters on the fieldbus.

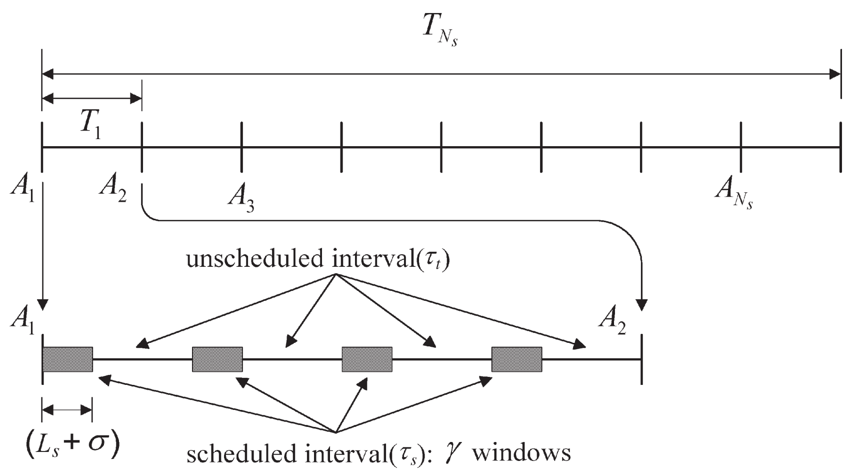

Fig. 3. Bandwidth-allocation scheme in the DLL of an FF.

III. Scheduling Method for an FF

The system manager of an FF network needs to specify two different types of scheduling: 1) FB and 2) LAS. FB scheduling is executed in the SMK of the user layer in each application process, whereas LAS scheduling is executed in the DLL of the LAS node that grants permission to a field device to use the fieldbus medium.

A. Relationship Between FB Scheduling and LAS Scheduling

FB scheduling in the SMK supports the cooperative execution of FBs located on the same link. A message generated by an FB must be delivered before the next scheduled message is generated by that same FB. The execution of an FB is governed by the SMK of each IFD connected to the link, whereas the data transfer of messages generated by the FB is governed by the LAS node on the link. Thus, the scheduling of an FB, performed in the SMK of each IFD, must be synchronized with the data transfer that is separately scheduled and controlled by the LAS. One repetition of either schedule is called a "macrocycle." Data transfers and FB execution times are synchronized by specifying their time offsets from the start of their respective macrocycles. The appropriate portions of the FB schedule and its macrocycle duration must be downloaded to the IFDs that are actually executing the FBs. The timing of the data transfers is defined in the LAS schedule, and this must be downloaded to the LAS. The LAS schedule contains a list of starting times and cycle periods for the data transfer for each FB. At precisely the scheduled time, the LAS sends a CD message to a specific IFD, specifying one of its data buffers. This IFD immediately broadcasts a message that contains the specified buffer data to all the IFDs on the fieldbus.

The specifications described in [4] and [8] do not specify how an LAS schedule list should be constructed. Rather, it is left up to the network designer to prepare an LAS schedule according to the characteristics of the FBs in the IFDs. This section presents our method of constructing the LAS schedule list based on the requirements for real-time communications of application systems, i.e., the scheduling of FBs.

B. Construction of the Scheduling Algorithm

The LAS schedule list is constructed using the bandwidth-allocation algorithm previously introduced [5] based on time-division multiple access. The fieldbus bandwidth is divided into time-division multiplexing intervals T_1, each of which is further divided into scheduled (τ_s) and unscheduled (τ_t) intervals, as shown in Fig. 3. During a scheduled interval, only one scheduled message packet is transmitted on the CD schedule. The length of τ_s is the time required to transmit one packet of periodic data (L_s) plus the LAS transaction time to execute a scheduled service (σ). On the other hand, unscheduled messages are transmitted during an unscheduled interval through the PT service. To reduce the delay of an unscheduled message that is transmitted between scheduled intervals, our bandwidth-allocation scheme uniformly distributes scheduled intervals across each T_1.

T_i and ϕ_i are ordered according to increasing values (i.e., $T_i \leq T_{i+1}$, $\phi_i \leq \phi_{i+1}$). In Fig. 3, T_1 and T_{N_s} are the minimum and maximum message-generation periods among N_s nodes, respectively. The basic concept of using the bandwidth-allocation scheme for our scheduling service is to schedule the message-generation time such that the number of scheduled messages generated during T_1 does not exceed γ. That is, N_s buffers in the medium dynamically share the γ $(\gamma \leq N_s)$ windows in T_1 such that the number of scheduled messages transmitted during T_1 does not exceed γ. Our scheduling method determines both t_i ($i = 1$ to N_s) and T_i ($i = 1$ to N_s) of the scheduled messages for FB scheduling. t_i and T_i are mapped to the schedule starting time s_i and the schedule period S_i in the LAS scheduling list maintained by the LAS.

From the bandwidth-allocation scheme presented in [5], T_i is determined as follows:

$$T_1 = \phi_1 \tag{1}$$

$$T_i = k_i T_1, \qquad k_i = 2^{\left\lfloor \log_2 \frac{\phi_i}{T_1} \right\rfloor}, \qquad \forall i = 2 \text{ to } N_s. \tag{2}$$

When T_i is determined by (1) and (2), it does not exceed the predetermined maximum allowable delay ϕ_i. In our scheduling

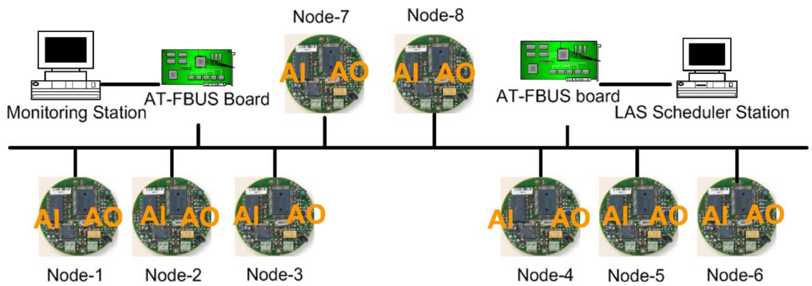

Fig. 4. Configuration of our experimental model.

method, the values for T_i ($i = 1$ to N_s) are selected to be integer multiples of each other

$$\mathrm{Rem}\left[\frac{T_j}{T_i}\right] = 0, \qquad \forall i, j\ (j \geq i). \tag{3}$$

Thus, as shown in Fig. 3, our scheduling method repeats at intervals of T_{N_s}. The number of windows γ in T_1 is determined by

$$\gamma = \lceil \alpha_K \rceil, \qquad \alpha_K = \sum_{i=1}^{N_s} \frac{T_1}{T_i} \tag{4}$$

where α_K is the average number of scheduled messages generated during T_1.

To restrict the number of scheduled messages generated during T_1 within γ, an additional constraint is required on the time offset t_i. Let $u^n(A_l)$ be the number of scheduled messages generated at A_l, counting from buffer 1 to n. Starting from $i = 1$, the message-generation starting time t_i $(t_i \leq t_{i+1})$ at node i is determined from [5] to be

$$t_i = \inf\left[A_l \geq A_{l-1} : u^i(A_l) \leq \gamma\right]$$
$$i = 1 \text{ to } N_s; \quad l = 1 \text{ to } T_{N_s}/T_1. \tag{5}$$

The stability condition for a scheduled message is

$$\gamma(\tau_s + \tau_t) \leq T_1. \tag{6}$$

If this condition is not satisfied, the network designer must reduce N_s in the local link. The scheduling list maintained by the LAS can be constructed by mapping the FB schedules t_i and T_i to the starting time s_i and the period S_i of the LAS schedule service as follows:

$$s_i = t_i + \frac{T_1}{\gamma}\left[u^i(A_l) - 1\right] \tag{7}$$
$$S_i = T_i. \tag{8}$$

The starting time s_i in (7) uniformly distributes the scheduled intervals across T_1 (see Fig. 3). Applying our scheduling method, all of the scheduled messages are transmitted within T_1 ($= \phi_1$, which is the minimum of the maximum allowable delay for FB messages). Thus, the requirement of a maximum allowable delay of ϕ_i $(\geq \phi_1)$ is satisfied for all nodes ($i = 1$ to N_s).

Fig. 5. Our experimental model.

C. Implementation of the Scheduling Algorithm in an FF

Scheduling is the most common way of coordinating FB execution in an FF. There exists only one schedule that includes both the FB and LAS schedules for each FF. The FF network system manager can generate a schedule and download it to the IFDs using a configuration tool. The schedule is stored in the system management/network management virtual field device in the IFD. This schedule tells the IFD when to execute its FBs (FB schedule) and tells the LAS when to issue a CD to the FB buffer (LAS schedule). The data in the FB buffer of the IFD will be published when it receives a CD from the LAS.

The FB schedule contains the start time offset and the macrocycle. The start time offset is measured from the beginning of the absolute link schedule start time, which is known by all IFDs on the FF. As previously mentioned, a macrocycle is a single iteration of a schedule in an IFD. The system manager of the FF network can construct an FB schedule by mapping t_i and T_i in (5) and (6) to the start time offset and macrocycle of the corresponding IFD. The LAS schedule list can be constructed by mapping s_i and S_i in (7) and (8) to the start time offset and macrocycle of the LAS schedule.

IV. Description of Our Experimental Model

In this paper, we demonstrated that the scheduling method presented in Section III can be implemented at the system management level of an FF. We developed an experimental model using an FF process instrumentation and control starter kit that contained all of the required components that are necessary for a small fieldbus control system [9]–[13]. As shown in Fig. 4, the experimental model consists of two PC and eight round cards. Each PC interface card contains embedded FF communication-stack software and a high-level FF interface. The round card

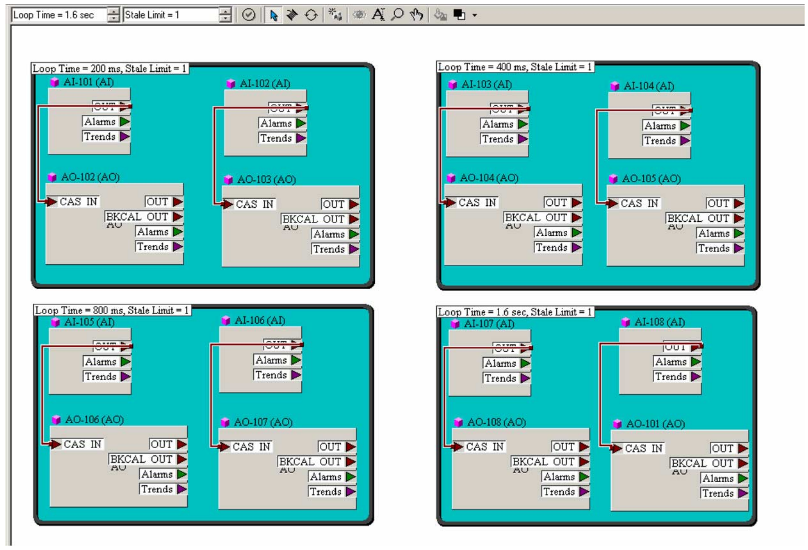

Fig. 6. Screen capture of the execution of the FBs in the experimental model.

is a standalone device that provides an FF interface and can be connected to a daughter card to implement the DI, DO, analog-to-digital, or digital-to-analog functions for sensors or actuators. In this experimental model, one PC plays the role of an LAS station, and the other PC monitors and analyzes the FF data packets. The monitor station captures the time-stamped data packets that are passed between nodes and displays them in real time on the monitor. Fig. 5 shows a photograph of our experimental model. Each round card IFD node has a set of functions that it can perform. These functions are represented as the FBs in the IFD node. In our experimental model, each round card has both AI and AO blocks. Fig. 6 is a screen capture of the FB execution in our experimental model, where AI-101(AI), AO-102(AO), etc. are the designations of the individual FBs. We used the NI-FBUS Configurator software [9] to generate the FBs for each round card. FBs can be synchronized simply by connecting an arrow from the output port of one FB to the input port of another. Fig. 6 also shows that the loop times for each FB are assigned the value of macrocycle T_i that is determined from the algorithm.

The schedule of data traffic allows for communication between a given set of IFD nodes. The schedule can be divided into two parts: 1) an FB schedule that determines when a block executes and 2) an LAS schedule that determines when data parameters are published over the fieldbus. The FB schedule is downloaded to the round card IFD nodes that contain the block, and the LAS schedule is downloaded to the LAS station. Using the schedule configuration tool provided with the starter kit, the system manager of the FF network determines the execution order and timing of the schedule. Fig. 7 shows the schedule window of our experimental model. A list of the schedule objects appears to the left of the schedule window. The timing diagram on the right side displays the execution times of the FBs and indicates when the data are transmitted on the bus. The start time offset for each FB schedule [indicated on the timing diagram as AI-101(AI), AI-102(AI), etc.] can be changed by sliding the corresponding bar to the desired execution time. The start time offset of the LAS schedule (indicated on the timing diagram by “OUT”) can also be changed by sliding the corresponding bar to the exact execution time.

The starter kit used in this study does not provide a configuration tool for generating unscheduled messages in the IFD nodes. In our experimental model, we generated unscheduled messages using an FB monitor program in the LAS station. This program allowed the LAS station to generate unscheduled messages for the monitoring of the input and output data of the FBs. The unscheduled messages were transmitted using the PT service.

V. Experimental Results

In this paper, we made several measurements under different experimental conditions to determine whether the algorithm is valid. We varied the transmission time of the scheduled messages to change the traffic load on the network and adjusted the maximum allowable delay in the FBs at each node to change the real-time requirements of the application system. In this paper, we present one representative experimental result to demonstrate that the proposed scheduling algorithm is indeed valid.

In our experimental model, the data transmission rate was set to 31.25 kb/s. The length of a scheduled message was fixed at 39 B, and its transmission time L_s was set to 9.984 ms. The transaction time for the LAS to execute a message transmission σ was determined to be 5 ms for our model. The maximum allowable delay for the scheduled messages ϕ_i generated by the FBs in IFD nodes 1 to 8 was set at 200, 200, 500, 500, 1000, 1000, 2000, and 2000 ms, respectively. Using the algorithm described in Section III, the maximum number of scheduled messages that could be generated during T_1 was determined to be $\gamma = 4$. The FB schedule period T_i of the scheduled messages from IFD nodes 1 to 8 was set to 200, 200, 400, 400, 800, 800, 1600, and 1600 ms, respectively. The FB schedule start time t_i from IFD nodes 1 to 8 was set to 0, 0, 0, 0, 200, 250, 600, and 600 ms, respectively. The LAS schedule period S_i

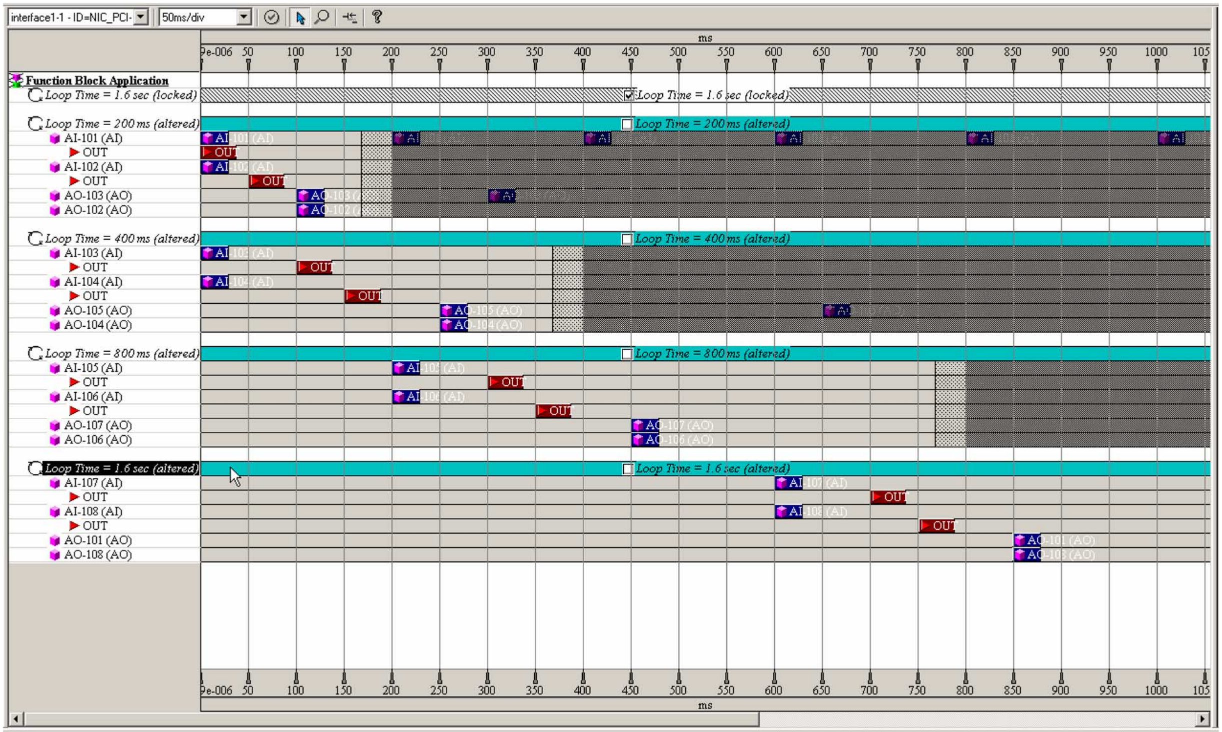

Fig. 7. Schedule window of our experimental model.

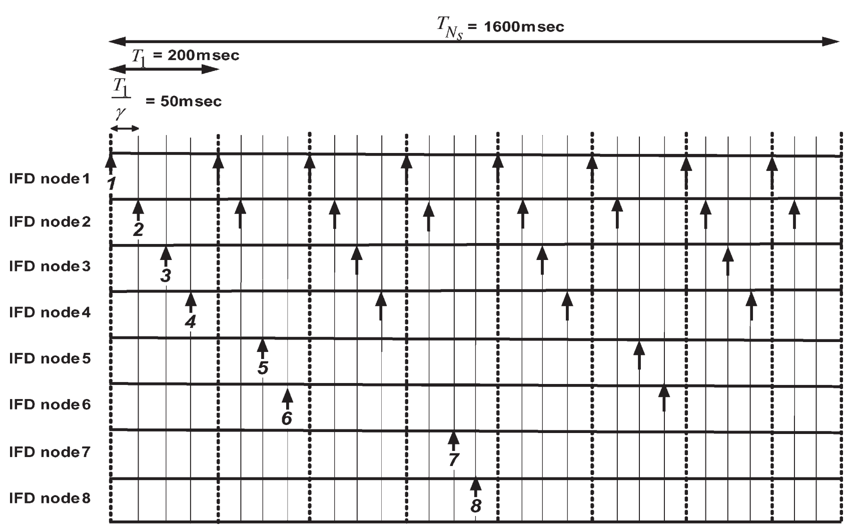

Fig. 8. LAS schedule pattern for our experimental model.

of the scheduled messages from IFD nodes 1 to 8 was set to the same values as for T_i, and the LAS schedule start time s_i from IFD nodes 1 to 8 was 0, 50, 100, 150, 300, 350, 700, and 750 ms, respectively. Fig. 8 shows the LAS schedule pattern with respect to the given S_i and s_i. As shown in Fig. 8, the $\gamma = 4$ windows are uniformly distributed across the interval T_1. This pattern repeats at intervals of T_{N_s}.

As previously mentioned, the starter kit that was used did not provide a configuration tool for unscheduled messages. In our experimental model, unscheduled messages of varying lengths were randomly generated by the monitoring program. For this experiment, the maximum length of an unscheduled message was restricted to 110 B, and its transmission time was set to 28.16 ms. For any of the given experimental processes, the stability condition for a scheduled message, which is calculated as $\gamma(\tau_s + \tau_t) \leq T_1$ $[4 \times (14.984 + 33.16) = 192.576 \text{ ms} \leq 200 \text{ ms}]$, was satisfied.

Fig. 9 shows the delay distribution of the scheduled and unscheduled messages measured using the experimental model. The delay of none of the scheduled messages exceeded $\phi_1 = 200$ ms; thus, the delay requirements were satisfied. Fig. 9 also shows that the delay of the unscheduled messages did not exceed $T_1 = 200$ ms. The average delay of all the unscheduled messages was 73.2 ms.

To examine the validity of our scheduling method, we performed a second experiment, choosing arbitrary values for the

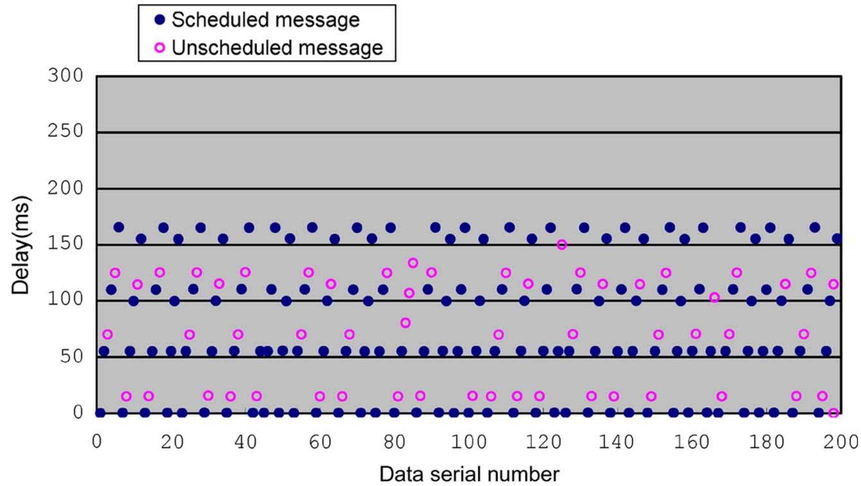

Fig. 9. Delay distribution of the scheduled and unscheduled messages when our scheduling method was applied.

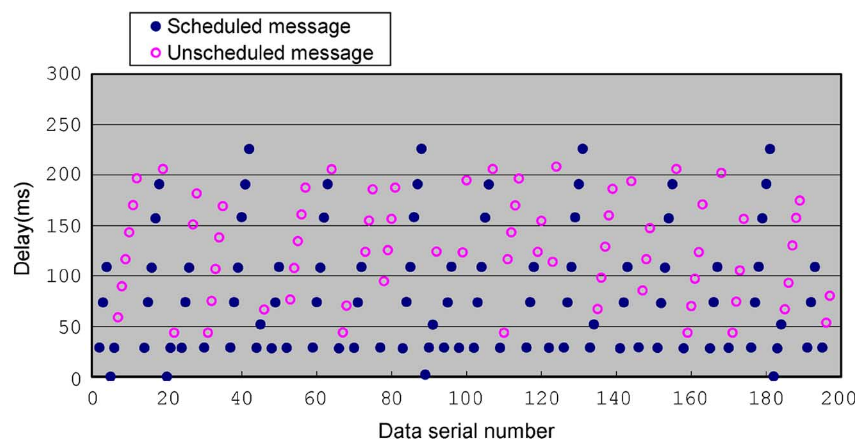

Fig. 10. Delay distribution of scheduled and unscheduled messages without our scheduling method.

TABLE I
COMPARISON OF THE MESSAGE DELAYS IN THE TWO EXPERIMENTS (IN MILLISECONDS)

	Scheduling model			Arbitrary setting		
	maximum	minimum	average	maximum	minimum	average
Scheduled message	165	0.2	77.5	226	0.2	63.2
Unscheduled message	150.2	15	73.2	208	44	128.4

LAS schedule period S_i and the LAS schedule start time s_i. The maximum allowable delay of the scheduled messages ϕ_i generated from IFD nodes 1 to 8 was the same as in the previous experiment. The LAS schedule period of a scheduled message S_i from IFD nodes 1 to 8 was determined to have the same values of ϕ_i, i.e., 200, 200, 500, 500, 1000, 1000, 2000, and 2000 ms, respectively. The LAS schedule start time s_i from IFD nodes 1 to 8 was set arbitrarily to 0, 25, 50, 72, 100, 125, 148, and 170 ms, respectively.

Fig. 10 shows that the transmission pattern of the scheduled messages in the second experiment was not uniform. It also shows that the delay of some scheduled messages exceeded $\phi_1 = 200$ ms. If this happens when IFD node 1 transmits its scheduled message, the delay requirement for IFD node 1 $(\phi_1 = 200 \text{ ms})$ cannot be satisfied. The delay of unscheduled messages was also longer than in our scheduling method. This is because unscheduled intervals using the token-passing service are randomly distributed in the time-division multiplexing bandwidth.

Table I shows the minimum, maximum, and average delays of the scheduled and unscheduled messages for the first and second experiments. The delay performance of both the scheduled and unscheduled messages was better when our scheduling method was applied. For a scheduled message, the maximum delay is an important design factor, although the average delay is not.

VI. Conclusion and Recommendations for Future Work

This paper has demonstrated that our scheduling algorithm using the previously introduced bandwidth-allocation scheme [5] is applicable to a practical FF system that consists of IFDs that are typical of those used in process instrumentation and control industries. Our scheduling algorithm can be easily implemented at the system management level of an FF by setting the appropriate scheduling parameters in the FB and LAS schedules. Our experimental results have shown that our scheduling method satisfies the specified delay requirements for scheduled messages.

References

[1] P. Pinceti, "Fieldbus: More than a communication link," *IEEE Instrum. Meas. Mag.*, vol. 7, no. 1, pp. 17–23, Mar. 2004.
[2] M. Sveda, P. Benes, R. Vrba, and F. Zezulka, "Introduction to industrial sensor networking," in *Handbook of Sensor Networks: Compact Wireless and Wired Sensing Systems*. Boca Raton, FL: CRC, 2004, pp. 10/1–10/25.
[3] J. Berge, *Fieldbuses for Process Control: Engineering, Operation, and Maintenance*. Research Triangle Park, NC: ISA, 2004.
[4] *Foundation Specification*, Fieldbus Found., Austin, TX, 1996.
[5] S. H. Hong and I.-H. Choi, "Experimental evaluation of a bandwidth allocation scheme for Foundation fieldbus," *IEEE Trans. Instrum. Meas.*, vol. 52, no. 6, pp. 1787–1791, Dec. 2003.
[6] *Information Processing Systems—Open Systems Interconnection—Basic Reference Model*, ISO 7498, 1989.
[7] *Fieldbus Standard for Use in Industrial Control Systems: Physical Layer Specification and Service Definition*, IEC 61158-2, 2003.
[8] *Fieldbus Standard for Use in Industrial Control Systems: Data Link Protocol Specification*, IEC 61158-4, 2003.
[9] *NI-FBUS Configurator User Manual*, Nat. Instrum., Aug. 1998. Part Number 321423B-01.
[10] *MC68331-Based Fieldbus Round Card User Manual*, Nat. Instrum., Jan. 1998. Part Number 321 866A-01.
[11] *NI-FBUS Communications Manager User Manual for Windows 95 and Windows NT*, Nat. Instrum., Jul. 1997. Part Number 321287B-01.
[12] *NI-FBUS Monitor User Manual*, Nat. Instrum., Mar. 1997. Part Number 321018B-01.
[13] *NI-FBUS Function Block Shell Reference Manual*, Nat. Instrum., Jan. 1998. Part Number 326016C-01.

Seung Ho Hong (M'92) received the B.S. degree in mechanical engineering from Yonsei University, Seoul, Korea, in 1982, the M.S. degree in mechanical engineering from Texas Tech University, Lubbock, in 1985, and the Ph.D. degree in mechanical engineering from the Pennsylvania State University, University Park, in 1989.

He is a Professor with the Department of Electrical Engineering and Computer Science, Hanyang University, Ansan, Korea. He is the Director of the Ubiquitous Sensor Networks Research Center, Gyeonggi, Korea: a subsidiary of the Gyeonggi Regional Research Center. His research interests are in the areas of ubiquitous sensor networks, fieldbuses and industrial communication networks, building and home network systems, and networked automation and control systems.

Sung Min Song received the B.S. degree in electrical engineering and the M.S. degree in computer science from Hanyang University, Ansan, Korea, in 2001 and 2003, respectively.

He was with the System Research Laboratory, LG Electronics Inc., Anyang, Korea, from 2003 to 2005. Since 2005, he has been with LG-Nortel Company, Ltd., Anyang, where he is currently a Senior Research Engineer, developing a UMTS transceiver station software for wireless networks. His research interests include fieldbuses, resource management, and UMTS transceiver station software design.