Development of a Real-time Communication Service over Random Medium Access Scheme Networks

Young-yeol Choo*, and Jang-woo Kwon**

* Department of Computer Engineering, Dongmyong Univ. of Information Technology, Busan, Korea
(Tel : +82-51-610-8398; E-mail: yychoo@tit.ac.kr)

**Department of Computer Engineering, Dongmyong Univ. of Information Technology, Busan, Korea
(Tel : +81-51-610-8394; E-mail: jwkwon@tit.ac.kr)

Abstract: The increasing use of Ethernet-TCP/IP network in industry has led to the growing interest in its reliability in real-time applications such as automated manufacturing systems and process control systems. However, stochastic behavior of its medium access scheme makes it inadequate for time-critical applications. In order to guarantee hard real-time communication service in Ethernet-TCP/IP network, we proposed an algorithm running over TCP/IP protocol stack without modification of protocols. In this paper, we consider communication services guaranteeing deadlines of periodic real-time messages over MAC protocols that have unbounded medium access time. We propose a centralized token scheduling scheme for multiple access networks. The token is used to allow a station to transmit its message during the time amount that is appended to the token. The real-time performance of the proposed algorithm has been described.

Keywords: Real-time communication, Ethernet, periodic service, TCP/IP

1. INTRODUCTION

Increase in the use of real-time systems is widespread in such applications as multimedia, computer-integrated manufacturing, aircraft, process control, and so on. These applications require much higher bandwidth than the bandwidth provided by many traditional local area networks. In addition, they are often time-critical in the sense that a failure leads to a seriously abnormal state or a great degradation in performance.

Real-time systems are defined as those systems in which the correctness of the system is determined not only by the logical result of the computation/communication process, but also by the time at which the results are generated. Except for timing constraints, the important characteristics of real-time systems include predictability, schedulability, stability and dependability[1].

Because most real-time systems have a distributed implementation, real-time communication is crucial for the successful operation of real-time systems. In real-time communication, periodic messages constitute a large portion of real-time traffic. The message length is relatively short and does not exceed several hundred bytes. However, the arrival rate is very high. In real-time systems, it is well-known that time delays have a bad effect on system performance, making it difficult to design a stable system. In distributed real-time systems, there are many delay elements caused by the stochastic nature of computer systems, communication networks, and errors. The communication network is mainly responsible for delays.

Besides delay, delay variation can also result in a negative effect on real-time systems. It is known that time-varying delays can lower control performance and cause instability [2]. In distributed real-time systems, time-varying delays occur at the feedback path and the data sampling period. Generally, time-varying delays are difficult to treat theoretically and decrease the predictability of real-time systems behavior. Therefore, periodic communication service is necessary to facilitate the design and analysis and improve performance.

To cope with time delay and delay jitter in a multiple access network, it is very important to provide periodic communication services deterministically. First of all, in order to provide real-time communication services on multiple access networks, media access time to transmit packets should be bounded definitely. Obviously, it is the data link layer's responsibility to determine the transmission time since the MAC (Medium Access Control) sub-layer of data link layer controls access order of the nodes participating in network and carries out flow control and error control of packets. Next, the periodic service should be defined in service definition and supported by the algorithm in or above the protocol layer.

Due to the nature of automated factory or process, real-time communication is likely to take place among the entities on the same link. Ethernet may be one of the most well-known LAN standardized by IEEE 802 committee. The protocol of Ethernet is very simple and cost-effective. It has been largely employed in recent years as an industrial network connecting various types of plants. Previously, its applications were limited to the office environments due to the random medium access scheme. Recently Ethernet is adopted as several industrial networks for factory or process automation[3][4]. Ethernet interfaces working at transmission speeds of 100 Mbps are commonly available for simple process devices such as sensors/actuators. Nevertheless, non-deterministic behavior of its medium access scheme makes it inadequate for time-critical applications[5]. To guarantee real-time communication, many industrial networks, often called as fieldbus, have been proposed by various standard organizations. Examples of fieldbus include Profibus, Foundation fieldbus, WorldFIP, ControlNet, LonWork, Controller Area Network, Interbus, and so on[6-9]. Although the fieldbuses can satisfy the real-time requirements for industrial networks, they are not able to be interoperable with each other. The cost for hardware and software of fieldbus is relatively high. These are obstacles of wide-spread adoption of fieldbus.

In this paper, we consider communication services guaranteeing deadlines of real-time messages over MAC protocols that have unbounded medium access time. We propose a centralized token scheduling scheme for multiple access networks. The token is used to allow a station to transmit its message during the time amount that is appended to the token. The rest of the paper is organized as follows. In section 2, a description of message characteristics and network model are given. In section 3, we describe a centralized token scheduling scheme for real-time messages in multiple access networks where bounded media access time is not guaranteed. Section 4 contains the test results of performance of proposed scheme. Finally, conclusions are presented in section 5.

2. NETWORK AND MESSAGE MODELS

2.1 Network model

In this paper, we assume that the multiple access networks consist of n nodes connected via common multi-access link. Data Link layer protocol of each node is Ethernet or CSMA/CD (Carrier Sense Multiple Access with Collision Detection) with LLC (Logical Link Control). TCP and UDP over IP (Internet Protocol) are considered as Transport protocol. Real-time scheduler operates on TCP/UDP layer. The communication profile of each node is shown in Fig. 1. This profile allows us to use conventional application services such as FTP service, e-mail service and etc. without modification along with real-time communication service.

Real-time service	Conventional service (FTP, e-mail, etc.)
Real-time scheduler	
UDP	TCP
IP	
Ethernet	

Figure 1. Communication profile of a node

Each node constructing a network is classified as Link Control Node (LCN) and general node. LCN is a special node that allocates network bandwidth to general nodes by delegation token. Only the node that has delegation token can exploit communication link. LCN gives each general node access right to communication medium through passing delegation token. LCN also works as general node when it transmits messages. General node can be LCN according to predefined order if current LCN breaks down. Since access right to link is controlled by central LCN, the network is different from Token Bus, Token Ring or FDDI networks.

2.2 Message model

Messages may be classified as either real-time or non-real-time messages. Real-time messages fall into two classes according to their arrival patterns: *periodic /synchronous* and *aperiodic/asynchronous*[5]. Periodic messages are generated at periodic intervals and must satisfy their deadlines. Otherwise, they are regarded as lost. In many cases, their periods are considered to be equal to their deadlines. Because periodic messages are deterministic, the most scheduling policies follow the guaranteed strategy. Thus, the performance metrics for periodic messages are worst case utilization and worst case response time. At design time, these parameters should be considered for safe operation of real-time application. The periodic message, for example, includes process values in feedback control systems for rolling mill process.

In contrast to periodic messages, the arrival periods of aperiodic messages are not defined. Aperiodic messages are generated aperiodically and they should be transmitted by their predefined deadlines. In most cases, the system has little information of their arrival pattern or timing characteristics. Thus deadlines are important to aperiodic messages.

Except real-time messages, there are non-real-time messages arriving periodically or aperiodically. Voice data or process monitoring data for display belong to this class. Multiple access networks in real-time communication should be able to provide periodic, real-time aperiodic, and non-real-time communication services.

A real-time message at node i, M_i, is characterized as (C_i, D_i, P_i), where C_i is message size, D_i is deadline which is defined as a time interval from the message creation to the completion of its transmission, and P_i is the inter-creation time of messages. If there are m message streams in node i, we divide this node into m subnodes, where token passing time between subnodes equals to zero[10]. By doing this transformation, a network with multiple message streams in a node can be treated equivalently as one with one message stream per node . For periodic messages, we assume that $D_i = P_i$ and $P_{min} = \min\{P_1, P_2, \dots, P_n\}$. P_{min} is named as *basic period.*

3. PROPOSED REAL-TIME SCHEDULING

We now describe the operation of real-time scheduler and the scheduling procedure of real-time and non-real-time messages in detail. Message scheduling is accomplished at initialization phase.

3.1 Message scheduling table

At initialization phase, real-time messages are scheduled according to their periods. Period of real-time message M_i, P_i, is adjusted to integer multiple of *basic period* P_{min} as follows.

Let $a = P_{min}$. Then, the value of P_i is replaced as

$$P_i' = ka \le P_i < (k+1)a . \quad (1)$$

Hence, if real-time message at node i, M_i, is allocated bandwidth C_i within P_i', it can always meet its deadline. *Complete token cycle* time is given as $P'_{max} = \max\{ P_1', P_2', \dots, P_n'\}$. After re-adjusting all the periods, scheduling table for real-time message is obtained. At normal operation state, LCN dispatches delegation token according to the table. Non-real-time messages are served at residual time after all the real-time messages have been transmitted. An example of scheduling table is shown at Fig. 2. As shown in Fig. 2, *complete token cycle* is repeated at normal operation.

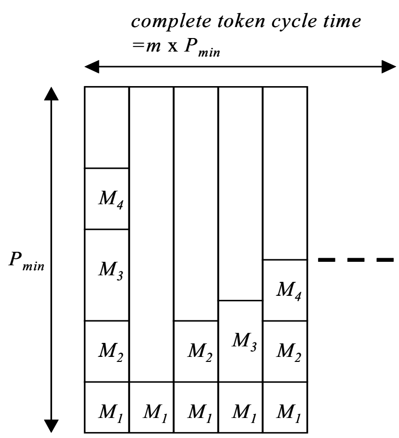

Figure 2. An illustrative example of scheduling table.

3.2 Operation of real-time scheduler

After successful initialization, LCN allocates each node bandwidth for real-time and non-real-time messages using token. The centralized real-time scheduler of token prevents data collisions during Ethernet communication. Token is short message frame which contains destination node id. and the allocation time for destined node. The format of token is shown at Fig. 3, where $T_{P(i)}$ and T_{NRT} stand for amount of time for real-time message M_i and non-real-time message, respectively. T_{NRT} is given as $P_{\min} - \sum_{i=1}^{m} T_{P(i)}$, where m is the number of messages to be sent in this *basic period*.

Frame type	Destination ID	$T_{P(i)}$	T_{NRT}

Figure 3. Format of token frame

As the token visits node *i* according to the scheduling table, LCN operates in each *basic period* as follows.

Op 1) Dispatch token carrying $T_{P(i)}$ and T_{NRT} to node *i*. and wait return token. Repeat until all the real-time messages are served.

Op 2) If time is remained after all the real-time messages are sent and this basic period is the last one during complete token cycle, update network node list. Otherwise, token is sent to each node for non-real-time message (sync. state).

A new node may want to join into the network during normal operation. Addition of the new active node can be managed by LCN in Op 2).

During the residual time, LCN node sends delegation token repeatedly to each node participating in the network. At this time, the value of $T_{P(i)}$ in token is zero and the bandwidth allocation is represented in T_{NRT}. The token visits node *i* according to the ascending order of *i*; hence, first node 1, then node 2, 3, … until node n, then again node 1. Although there are no messages to be transmitted in any node, LCN doesn't begin new periodic cycle. This guarantees that the token arrives at any node exactly at every basic period or integer multiple of basic period. The operation of LCN is presented as state transition diagram in Fig. 4.

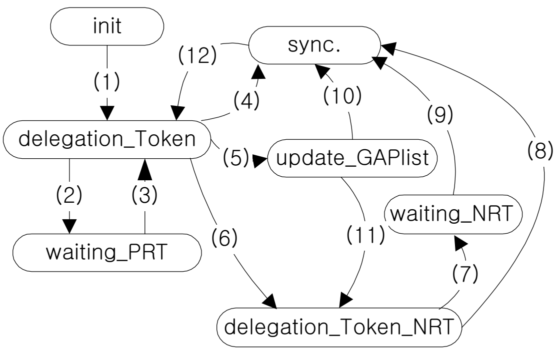

Figure 4. State transition diagram of LCN node

When a general node *i* receives *token*, it sets sets transmission timer TX_i to the sum of $T_{P(i)}$ and T_{NRT} in token frame. Transmission timer TX_i is defined to control how long node *i* may transmit messages. Node *i* is allowed to transmit messages until TX_i expires. TX_i always counts down. Real-time messages are served at first. Non-real-time messages can be transmitted using the residual time. If node *i* does not consume all the bandwidth allocation, instead of holding the token for the residual time r_i, r_i is set to residual time field in *return token*. Node *i* sends this return token to LCN to prevent the waste of the bandwidth.

The operation of general node I can be summarized as follows.

Op' 1) Upon receiving token, node *i* passes return token to LCN node immediately if there are no message to be transmitted. Otherwise, node *i* sets TX_i to the sum of $T_{P(i)}$ and T_{NRT}.

Op' 2) Node *i* may transmit messages until TX_i expires. TX_i continues to count down during message transmission.

Op' 3) If TX_j = 0 or node *i* runs out of messages to transmit, it sends return token to LCN with setting $T_{residual}$ in return token to TX_j.

State transition diagram for a general node is shown in Fig. 5.

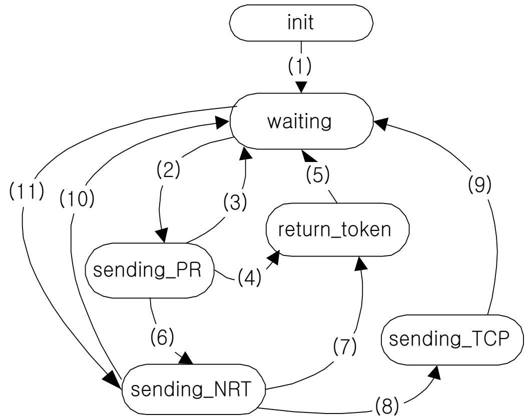

Figure 5. State transition diagram of general node.

4. PERFORMANCE TEST

To test working of proposed scheme and evaluate real-time performance, a testbed was implemented as shown in Fig. 6. Specifications of hardware and software used at the testbed are presented at Table 1.

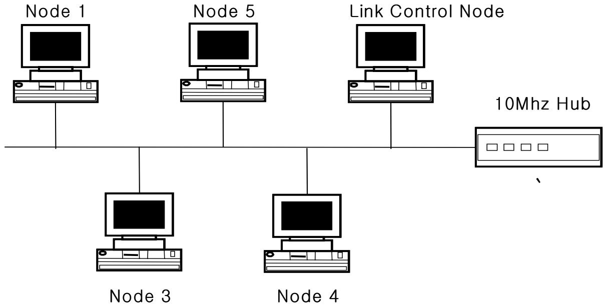

Fig. 6 Configuration of testbed

Table 1 Hardware and software specification of testbed

	OS	CPU	RAM	LAN
A	Windows 2000	PIII 500Mhz	256M	Realtek 8139
B	Windows 98 SE	Celeron 466Mhz	128M	3Com 509 ISA
C	Windows 98 SE	PII 333Mhz	128M	Realtek 8139
D	Windows 98 SE	PII 333Mhz	128M	Realtek 8139
E	Windows 98 SE	PIII 500Mhz	128M	Realtek 8209

At Fig. 6, five nodes including LCN have real-time messages to be transmitted. For the performance test, characteristics of messages are as follows.

- Periods of real-time messages at each node are equal to basic period.
- Bandwidth for message transmission is evenly allocated to five nodes.
- Non-real-time messages are generated exhaustively.

Unless a real-time message in any node was transmitted within the time interval between arrival of token at a node and departure of return to LCN, it is considered as failure and we quit the test. We increased basic period gradually from 80 to 140 milliseconds, as shown in Table 2, and check the communication failure. Test also has been carried out on various number of data packets. We assumed that each data consists of two bytes. Periodic messages are generally short but are exchanged frequently between communication entities. Moreover, owing to multiple access network environment, processing time for messages is more critical than transmission or propagation time. Hence, our test may be realistic.

Test results represent that proposed real-time scheduling scheme guarantees real-time communication services if the periods of real-time messages are larger that 110 millisecond. Because processing time for a message is dominant factor in communication delay, this value may be variable depending on the computational capability of each node.

Table 2. Test results according to length of basic period and the number of data packets.

No. of Data \ Basic period (ms)	140	110	105	100	95	90	85	80
1	O	O	O	O	O	X	X	X
2	O	O	O	O	X	X	X	X
4	O	O	O	O	X	X	X	X
5	O	O	O	O	X	X	X	X
6	O	O	O	X	X	X	X	X
7	O	O	O	X	X	X	X	X
10	O	O	O	X	X	X	X	X
13	O	O	O	X	X	X	X	X
16	O	O	O	X	X	X	X	X
19	O	O	O	X	X	X	X	X
22	O	O	X	X	X	X	X	X
25	O	O	X	X	X	X	X	X
28	O	O	X	X	X	X	X	X

O : success, X : failure

5. CONCLUSION

Ethernet and TCP/IP protocol suite are supported with fluent hardware and software solutions for industrial applications. Recently Ethernet is adopted in several industrial automation fields. Nevertheless, non-deterministic behavior of its medium access scheme makes it inadequate for time-critical applications. In order to guarantee periodic real-time communication, we proposed centralized real-time scheduler running over the protocol suite without modification of the protocols. Hence, it allows Ethernet and TCP/IP protocols to be used in real-time shop floor along with other non-real-time application. We tested the performance of proposed scheme using the testbed consisting of Personal Computers and dummy hub. However, because of limitation of Windows Operating System, proposed scheme doesn't guarantee the messages periods of which are less than 110 milliseconds, as revealed in the results of the performance test. To serve the message having shorter period, Real-time Operating System is recommended.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the contribution of Dr. Smith.

REFERENCES

[1] Herman Kopetz, "Real-time systems: Design principles for distributed embedded applications," Kluwer academic publishers, Massachusetts, 1997.

[2] Torngren, M. "Fundamentals of implementing real-time control applications in distributed computer systems," *Real-Time Systems*, 14, pp 219-250. 1998

[3] Daniele Miorandi and Stefano Vitturi, "Hybrid wired/wireless implementations of Probibus DP: a feasibility study based on Ethrnet and Bluetooth," *Computer Communications*, 27, pp. 946-960, 2004.

[4] G. Ye, H. Deng, L. Chen, L. Liu, and X. Wang, "A prototype switched Ethrnet data acquisition system," *Fusion Eng*. Des. 43, pp. 413-416, 1999.

[5] Kyung Chang Lee and Suk Lee, "Performance evaluation of switched Ethernet for real-time industrial communications," *Computer Standard & Interfaces*, 24, pp.411-423, 2002

[6] Profibus committee, Profibus technical Description, www.profibus.com

[7] IEC 61158-4, Digital data communications for measurement and control-Fieldbus for use in industrial control systems-Part 4: Data link protocol specification, 1999.

[8] FIP committee, FIP technical Description, www.worldfip.org.

[9] Phoenix Contact, Interbus basic, www.ibsclub.com/library/ibsinfo , 2004

[10] Young-yeol Choo and Cheeha Kim, "Periodic communication support in multiple access networks exploiting token with timer," *Lecture Notes in Computer Science*, Vol. 2662, pp. 141-150, 2003.