Communication Network Protocol for Real-Time Distributed Control and Its LSI Implementation

著者	亀山　充隆
journal or publication title	IEEE Transactions on Industrial Electronics
volume	44
number	3
page range	418-426
year	1997
URL	http://hdl.handle.net/10097/46844

doi: 10.1109/41.585841

Communication Network Protocol for Real-Time Distributed Control and Its LSI Implementation

Koji Kobayashi, Michitaka Kameyama, *Senior Member, IEEE,* and Tatsuo Higuchi, *Fellow, IEEE*

***Abstract*— A new token-passing mechanism, priority token passing, which features real-time access and fast detection and recovery of transmission errors, is discussed in detail in comparison with standard token-passing protocols, and its large-scale integration (LSI)-oriented design concept is described. Priority token passing includes only a small performance overhead, due to its switching functions, which can change network topology from ring to broadcast medium. A token-holding node passes the token to another node after determining the successor through priority comparison. Errors occurring during token passing can, thus, be detected and corrected simply and promptly. Priority token passing has a simple hardware implementation, requiring only small additions to the frame control circuitry, and has a small implementation overhead. The priority token-passing protocol and two other important network communication functions, dual ring network reconfiguration and high-level data link control (HDLC) normal response mode-based message transmission, are designed as a single finite-state machine, and implemented into a compact LSI chip. This integrated instrument network (IINET) chip provides complete network communication services and requires only three additional external electronic components for operation.**

***Index Terms*— Large-scale integration, local area networks, network fault tolerance, network reliability, protocols, real-time systems, reconfigurable architectures, token networks.**

I. INTRODUCTION

THE standardization of local area network (LAN) protocols has been successful in networks of personal computers and workstations. However, establishing standard protocols has not been popular in industrial and commercial control applications. One of the reasons for this is the fact that the cost and performance of control devices vary widely and, so, the options offered by standard protocols are insufficient for most small-scale controllers.

Recent advances in microelectronics have driven the growth in the application of small microprocessor-based controllers for industrial and commercial control. If these small controllers are interconnected via a communications network, a distributed control system can be constructed which is more versatile and offers higher reliability than a group of stand-alone controllers. The authors have investigated communication protocols for real-time control applications [1], [2].

Such networks require deterministic network access characteristics and must add only a small overhead to network performance. It is also necessary to have priority control mechanisms to expedite emergency messaging. The protocol has to have robust transmission error and temporary disturbance recovery mechanisms. Such problems must be detected and recovered by retries, with minimum performance degradation. Network communication must be maintainable, even during persistent and unrecoverable disturbances. This can be achieved either by media redundancy or by reconfiguration.

Most importantly, the communication protocol must be able to be implemented at a modest cost compared to that of the small controllers, which are typically used in control applications. The lack of inexpensive communications solutions has been one of the major factors preventing popularization of distributed real-time control.

Token passing is a well-known network access mechanism with deterministic access characteristics. Of the three LAN protocols originally standardized by the IEEE 802 Committee, two are token passing protocols [3], [4]. In this paper, a new token-passing protocol, priority token passing, is discussed in contrast to the standard token-passing protocols. It is shown that priority token passing exhibits both the token ring's low overhead and high throughput characteristics and the token bus's reliable token relay. The scale and complexity of implementation are considerably reduced for priority token passing, because of its simple mechanism.

Network integrity and message transmission are important factors of network communication. Priority token passing is implemented in a ring network, where failure of a ring segment may cause total network failure. The integrated instrument network (IINET) features a dual-ring reconfiguration mechanism with autonomous cooperation between nodes and isolates the failed segment regardless of failure mode, whether it is open, short circuit, or node failure. The message transmission protocol is based on the ISO standard high-level data link control (HDLC) normal response mode [5], which provides segmented message transmission and eliminates the possibility of message loss or duplication.

The ISO seven-layer hierarchical network architecture entails a large implementation overhead for small controllers and, so, a collapsed architecture is often used, where multiple ISO layers are combined into a single layer. Typical network

---

Manuscript received April 12, 1995; revised May 26, 1996.

K. Kobayashi is with the Second Engineering Department, Building Systems Division, Yamatake-Honeywell Co. Ltd., Isehara, 259-11 Japan.

M. Kameyama is with the Department of Computer and Mathematical Science, Graduate School of Information Science, Tohoku University, Aoba, Aramaki, Aoba-ku, Sendai, 980 Japan.

T. Higuchi is with the Department of System Information Science, Graduate School of Information Science, Tohoku University, Aoba, Aramaki, Aoba-ku, Sendai, 980 Japan.

Publisher Item Identifier S 0278-0046(97)04142-7.

TABLE I
COMPARISON OF TOKEN-PASSING ACCESS METHODS

NAME	TOPOLOGY	TOKEN MANAGEMENT		PRIORITY
		TOKEN FRAME ERROR DETECTION	TOKEN MONITORING	
TOKEN BUS	BUS	FRAME CHECK SEQUENCE	TOKEN HOLDING NODE	ROTATION TIMER
TOKEN RING	RING (STAR)	MANCHESTER CODE VIOLATION	ACTIVE MONITOR	RESERVATION
PRIORITY TOKEN PASS	DUAL RING	FRAME CHECK SEQUENCE	TOKEN HOLDING NODE	DIRECT REFERENCE

communication protocols which implement collapsed architecture are Real-Time MAP and FieldBus. The IINET also uses a collapsed architecture, i.e., protocols for priority token passing, loopback reconfiguration, and HDLC normal response mode message transmission are implemented in a single-state machine.

The ISO defined each layer as an independent collection of functions. However, in actual implementations, functional selections of layers may become dependent on each other. For example, simple connectionless services are most commonly used in logical link sublayer services, provided that the transport layer error detection and recovery function compensates for the lack of the function in the datalink layer. The IINET is designed to provide a complete package of communication functions in the shape of an LSI and not to be dependent on microprocessor-based upper layer functions, as long as communication is restricted to a single network.

These techniques achieve a more compact communication protocol processor, compared to those which implement standard token-passing protocols. A network communication protocol which can be embedded into small controllers with minimal hardware and software overhead and its LSI implementation is, thus, achieved.

II. COMPARISON OF TOKEN-PASSING PROTOCOLS

Differences between token-passing protocols discussed here have been broken down into three categories: token passing, token maintenance, and prioritization. The major criteria of comparison are:

1) small overhead for real-time access, including fast error detection and recovery;
2) implementation simplicity.

It is to be noted that a mechanism which needs to identify a single coordinating node almost always produces overhead, because the network in consideration does not have a centralized control node. The designation of a node as the coordinator usually necessitates a selection mechanism for choosing one node from its peers, when the network is initialized or when the current coordinator node ceases to function. A resolution mechanism is also required, for when accidental duplication occurs.

Each token-passing protocol is discussed in detail. Table I summarizes the characteristics of token bus, token ring, and priority token passing. Structures of token frames are shown in Fig. 1.

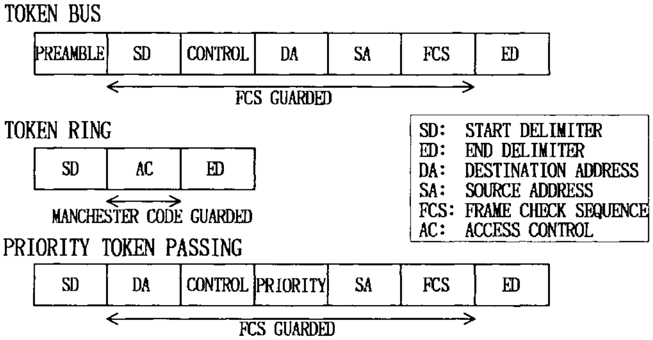

Fig. 1. Token frame structures.

A. Token Bus

Token bus is defined in a bus topology network. Therefore, a logical sequencing of connections has to be established to define the order of token rotation. At network initialization, the token rotation definition table has to be automatically created. Change of membership occurs when a node either leaves or joins the network and requires dynamic update of this table. Such update apparently requires a complicated mechanism.

As the number of nodes in a network increases, the overhead required to rotate the token increases linearly. Because the time needed to pass the token is relatively large, most bandwidth may be consumed just by token rotation, if the number of nodes is sufficiently large.

The detection of, and recovery from, transmission errors at the time of token passing is relatively easy, because the responsibility for reliable token passing is taken by the token-holding node and because the node which is going to be the successor is specified prior to a token-passing sequence.

Four priority levels are defined. Each node measures the interval of transmission availability per priority. The message transmission of a particular level is allowed only when the rotation time margin is less than the predefined maximum. In this priority control mechanism, throughput may be limited in cases when a large load of low -level priority messages is scheduled, because a certain amount of bandwidth for higher priority messages must be reserved at all times.

B. Token Ring

Token ring's access method is simple to implement, because it refers to the order of the physical connection. The status field of a token frame is tested at each node and modified if its status is free and the node has a request to transmit. The testing and modification of the token status are executed in 2.5-b time [6]. Bit time here is defined as the time needed to transmit a single bit of information at the transmission data rate.

The reliability of token passing is, however, sacrificed. If a token frame is distorted by a transmission error, the validity of the frame may not be detected immediately. The responsibility for reliable token passing is not taken by the node which issues the token, nor by the node which takes over the token, because the successor node is not identified prior to the token passing

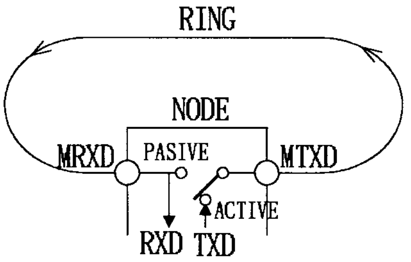

Fig. 2. Switch.

and because no direct acknowledgment mechanism is provided between the token-holding node and the successor. Also, the frame which represents the token is not guarded by a frame check sequence but by the less stringent Manchester code violation check. For these reasons, ultimate responsibility for error detection and recovery is assigned to a designated node, which is called the active monitor. The initial designation, possible replacement, or detection of duplications of the active monitor also require a rather complicated mechanism.

Token ring has 127 levels of access priority. Because the token is captured by the first node requesting transmission encountered on ring rotation, a mechanism is employed in which priority is reserved at the preceding message transmission cycle. The active monitor has to provide LIFO capabilities to restore overridden priorities when a higher priority transmission request is completed.

C. Priority Token Passing

Priority token passing can switch from a sequential ring topology to a broadcast bus topology by a switch attached to each node. A node is in passive mode if it is receive-only mode and passes on all received signals. It is in active mode if it receives and transmits simultaneously. The mode is determined by the position of the switch, as shown in Fig. 2. When all the nodes except the token-holding node are in the passive mode, the ring is equivalent to a broadcast bus. The per-node latency is basically the time needed to resynchronize the received signal by a digital phase-locked loop.

A third mode, echo mode, is defined as a subclass of the active mode. In echo mode, the reception of a token frame is anticipated, and the received token frame is either bypassed or replaced by a new token frame, depending on the result of priority comparison. Priority comparison and frame modification are executed sequentially, and take 3.6-b time together with digital phase-locked loop synchronization. Compared to token ring, priority token passing uses the complete HDLC frame format [7] for the token. Priority token passing designates its successor at the time of token passing, so the responsibility for error detection and recovery is clearly defined. All information in a token frame is guarded by the frame check sequence, so that transmission errors are easily detected and recovered by retries, as with other HDLC-based frames.

The number of priority levels defined is 127. The priority level is referred straightforwardly to identify the successor, which claims the highest priority at the time of token passing.

Priority token passing consists of four phases, as shown in Fig. 3. Prior to token passing, the token-holding node is in active mode and all the rest are in passive mode.

The first phase is soliciting. The token-holding node broadcasts a pass frame, which solicits the nodes which have transmission requests to switch their connection to echo mode.

The second phase is priority comparison. The token-holding node transmits a null token, which has a lower priority level than any message transmission request, with the address of the token-holding node as the sender field value. Each node which switches to echo mode compares the priority field of the received token, and replaces it with its own token frame only if the level of priority exceeds the level of the received token. The received token is otherwise transmitted intact. The token-holding node recognizes the existence of the token request and the highest priority level value and node address of the requester, if it receives a modified token frame. When there are a plural number of nodes requesting the same highest priority, a node closest to the token-holding node on the ring first modifies the token and will be the successor.

When there is no request, the token-holding node receives the original null token. Phase three is skipped in this case, and the token-holding node proceeds to phase four.

The third phase is successor designation. If there is a request, the token-holding node transmits a busy token, which has a priority level of busy, a higher priority than any of the message transmission requests, and the successor's node address.

The fourth phase is acknowledgment. The node designated by the busy token recognizes that it is now the successor. It transmits an OK frame in response, and becomes the new token-holding node. When there is no request, the original token-holding node transmits an OK frame, which shows no token movement.

The performance characteristics of token-passing protocols has been studied extensively [8], [9]. The performance of priority token passing in a typical configuration is examined together with those of token ring and token Bus by computer simulation. Only the results of throughput versus acquisition delay characteristics are shown in Fig. 4, because this comparison most effectively demonstrates the differences among the three token-passing protocols. The message arrivals to a node are assumed to follow a Poisson distribution. The size of messages is fixed to 800 bytes. Transmission delay is assumed to be zero. Transmission speed is normalized to bit time and mapped to 10 Mb/s. The number of nodes on a network is set at 50.

It is shown that the characteristics of priority token passing, shown under the IINET in the figure, is superior to that of a token bus, and very close to that of a token ring. Priority token passing is superior to the token ring, in that it provides more reliable token passing and recovers much faster when there are transmission errors which destroy the token-passing sequence.

III. LSI IMPLEMENTATION

Fig. 5 shows a block diagram of the IINET chip and a network overview. The IINET is intended to provide total

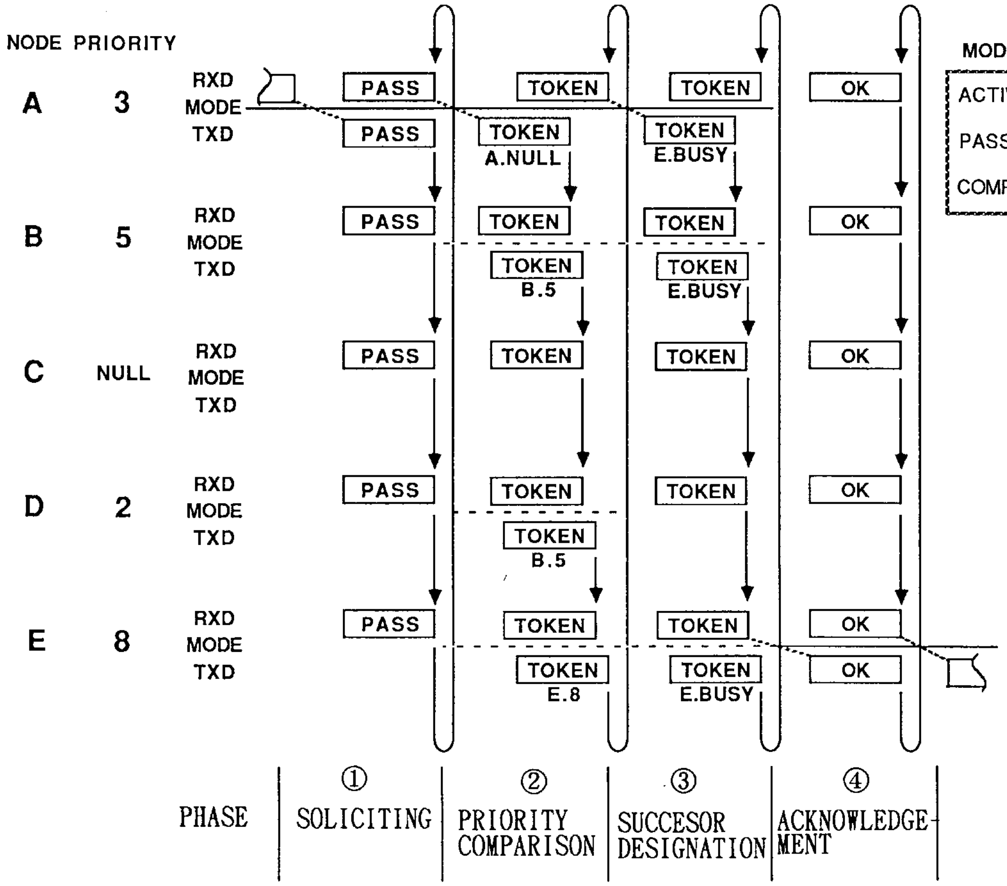

Fig. 3. Priority token rotation.

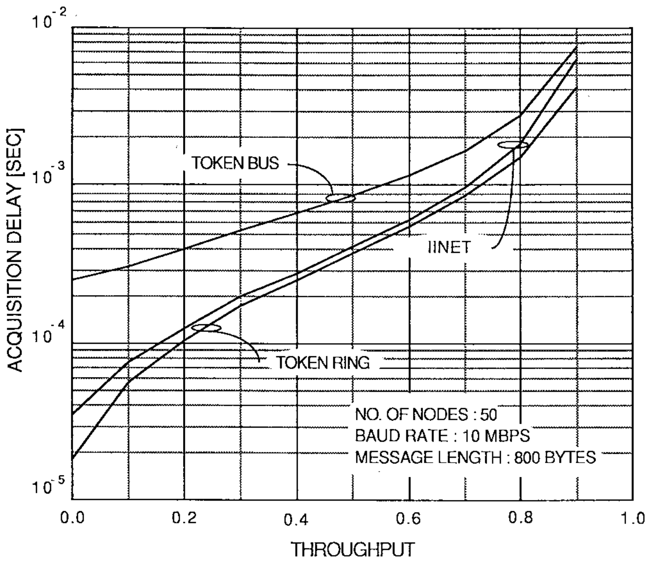

Fig. 4. Comparison of throughput characteristics.

network communication functions with a minimum number of external electronic components. The IINET chip needs an external 2 kbytes of dedicated RAM, of which half is a receive buffer and half is a transmission buffer. The receive buffer can store 31 messages and is accessed as 1-kilobyte FIFO from the microprocessor. The transmission buffer is used as a dual buffer, which can accept the next message while a message is still in the process of transmission. This powerful buffering structure eliminates an additional burden on the microprocessor, due to the high transmission baud rate. For applications where the amount of message transmission and reception is within the performance range of the microprocessor, a DMA interface may not be needed. More detailed descriptions of major blocks can be found in later sections.

TABLE II
STATISTICS OF LSI

PROCESS	CMOS 1.2 MICRON, STANDARD CELL
SCALE	10,000 GATES, 18 Kbit ROM, 700 bit RAM
CLOCK	32 MHz
INSTRUCTION CYCLE	125 ns
BAUD RATE	1 MBPS
POWER CONSUMPTION	200 mW
PACKAGE	120 PIN PLASTIC FLAT PACKAGE

The IINET LSI is implemented using 1.2-μm CMOS technology. Table II summarizes the statistics and Fig. 6 shows a microphotograph of the IINET LSI. The IINET LSI has a 32-MHz oscillator circuit and is designed for four transmission data rate options ranging from 125 kb/s to 1 Mb/s. A digital phase-locked loop circuit, which is used for resynchronization, runs at 16 times the transmission data rate.

The device has a microsequencer-controlled communication protocol processor. The microprogram ROM is 18-b wide and contains 1024 instructions. The instruction execution cycle of the microprogram is 125 ns. The microsequencer is tailored

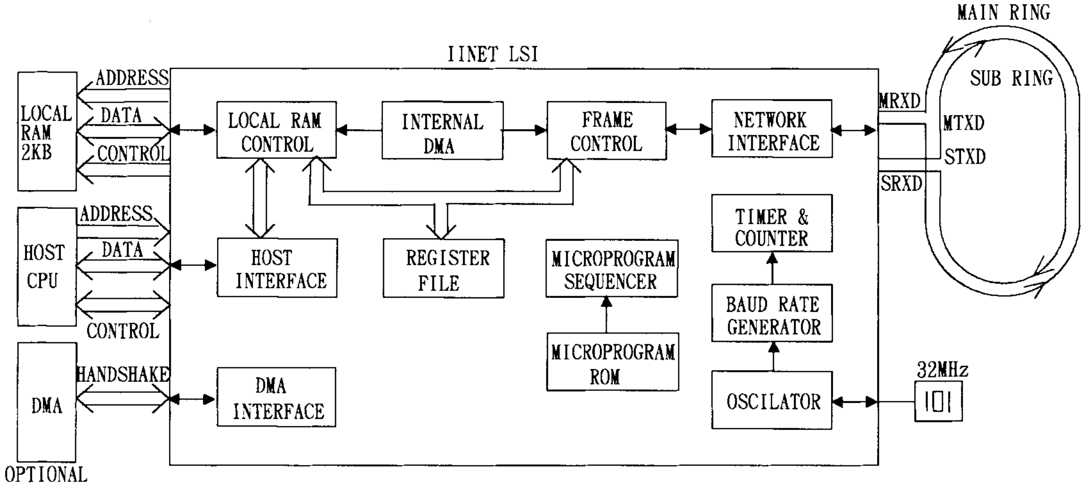

Fig. 5. IINET block diagram.

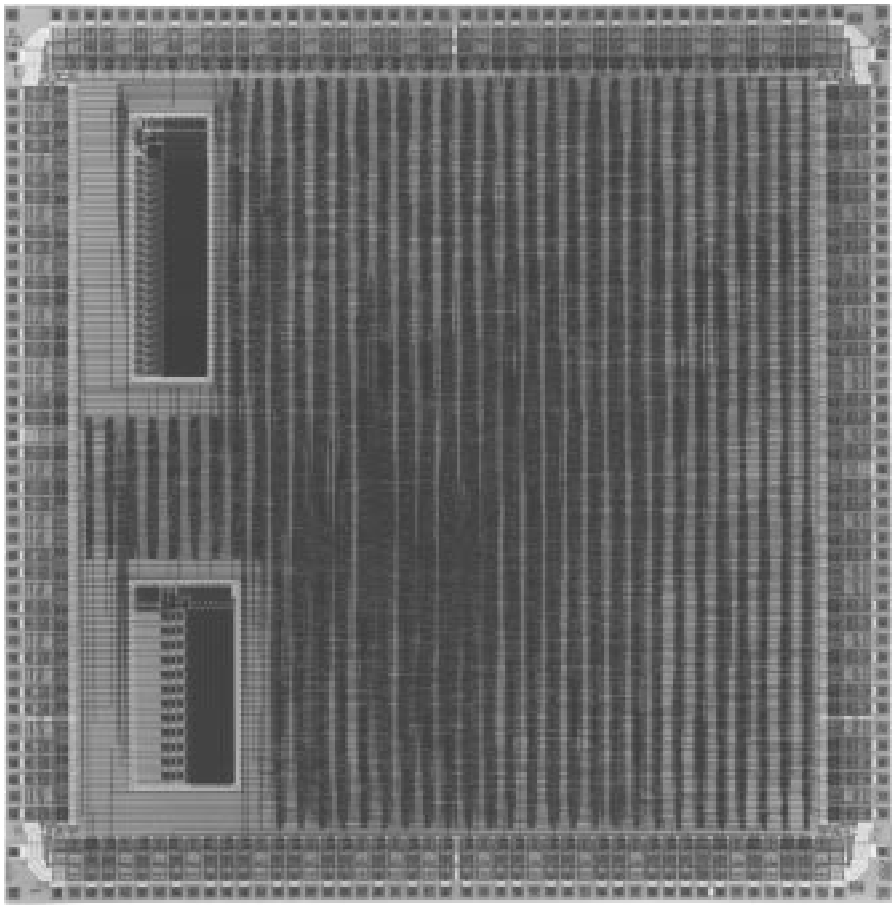

Fig. 6. Microphotograph of IINET LSI.

to describe the state machine, which represents the network communication protocol. A 64*11-b RAM is used as both the receive message delimiter stack and a scratchpad memory. The function of the delimiter stack is described in more detail in Section VI.

IV. NETWORK INTERFACE AND FRAME CONTROL

The network interface block consists of a switch, two digital phase-locked loop circuits, a non-return-to-zero-inverted (NRZI) encoder, and an NRZI decoder, as shown in Fig. 7. NRZI is the usual signal format for HDLC, which reduces the bandwidth of the transmission signal using HDLC's zero insertion and deletion techniques and easily detects framing errors. The IINET has two input and output pairs, MRXD and MTXD, and SRXD and STXD, to accommodate the loopback reconfiguration function. The function of the switch is to select an input port and an output port from the two inputs and the two outputs. A node is in normal configuration when it

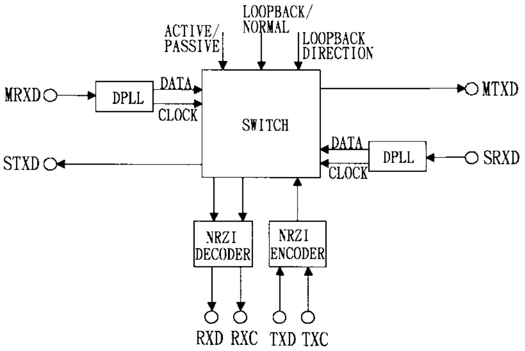

Fig. 7. Network interface.

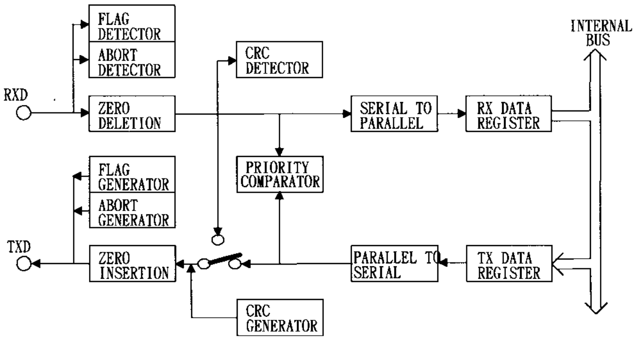

Fig. 8. Frame control.

receives from the MRXD port and transmits to the MTXD port. It has two loopback configurations to isolate failed segment. It is in loopback A configuration when it receives from the MRXD port and transmits to the STXD port. It is in loopback B configuration when it receives from the SRXD port and transmits to the MTXD port. Active and passive switching is applied between the specified ports in each configuration. The unspecified input and output ports are logically connected in each configuration. The digital phase-locked loop produces a synchronized clock from the received signal. The NRZI encoder and decoder encode and decode the NRZ signal to and from the NRZI individually.

Fig. 8 shows the frame control block. It has serial priority comparator circuits with a switch, in addition to the standard HDLC frame format encoder and decoder. The comparator is activated in echo mode and bitwise compares the value of the priority field of the received token frame with the node's priority. The switch is on the received signal side at first, and the position of the switch is fixed on either side at the first unmatched bit position. It switches to the node's signal side if the value of the bit is higher for the node's priority and, otherwise, stays on the received signal side. To make the comparison at the first unmatched bit position, the priority field has to be serialized from the highest weight bit position first. For this purpose, the bit sequencing is inverted in the priority field from the commonly used LSB-first sequencing. For example, if the received priority is 100 (B01100100) and the node's priority is 98 (B01100010), the first 5 b in MSB-first sequencing (B01100XXX) are common and can be transmitted regardless of the result of the comparison. This is a mechanism which executes both a priority comparison process and repeat or overwrite processes together and without significant delay. It is to be noted that the serial comparator is simply a 1-b comparator with a latch. This implementation of token passing achieves a very small overhead of logic complexity added to the standard HDLC frame control.

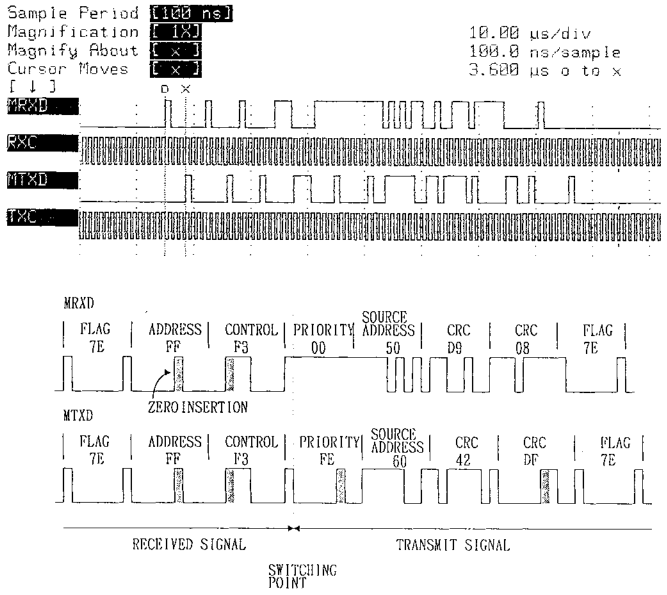

Fig. 9. Echo mode waveform.

Fig. 9 shows how priority token passing is executed. Fig. 9(a) shows an experimental priority token-passing wave train and Fig. 9(b) is for explanatory purposes. It is shown that only 3.6-b time is needed for priority comparison and resyncronization and that the complete HDLC frame is generated using the received signal until it comes to the bit where the priority field values are unmatched. In both figures, the signal level is inverted. The low level represents one and the high level represents zero.

V. AUTONOMOUS LOOPBACK RECONFIGURATION

In a ring network, the media is composed of simple point-to-point unidirectional transmission links between nodes. This has the benefit of straightforward applicability for noise-immune fiberoptic links. However, the ring topology is fatal in the case of a single segment failure. In a token ring, a hub is mandatory and effectively changes the ring topology to a star topology, but also introduces additional cost overhead.

If a ring network is configured as a counter-rotating dual-ring network with a loopback reconfiguration, it can isolate any failures, regardless of the causes, as is well known [10]. Implementing a loopback reconfiguration algorithm typically produces a large overhead and, therefore, only expensive networks, such as FDDI [11], used to offer this functionality. The IINET exploits a dual-ring network with loopback reconfiguration functions, without significantly increasing the cost overhead.

One of the two rings, which is continuously used in the normal state, is called the main ring. The other ring is called the subring. When a consistent transmission error that cannot be recovered by repetitive retries has been detected, the

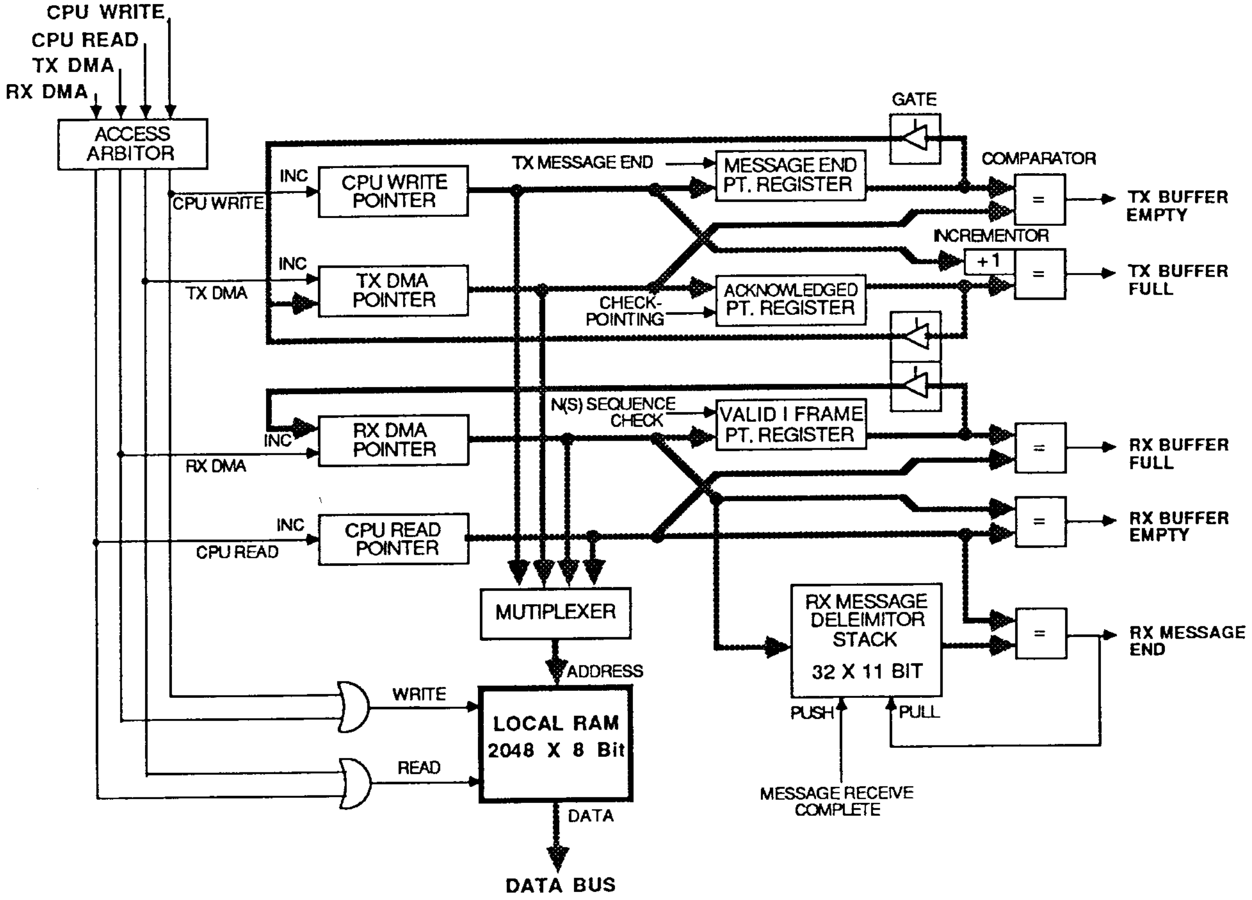

Fig. 10. Local RAM control.

loopback reconfiguration mechanism is activated. All nodes on the ring except the token-holding node are divided into two groups, depending on their locations relative to the token-holding node and the failed segment. Each node recognizes its own group membership by testing whether it receives frames on the main ring from the token-holding node or it is timing out. Depending on the condition, it switches the configuration to the rings, to either loopback A configuration or loopback B configuration, respectively, and further waits for reception of connection check-sequence frames. While the nodes stay in either loopback configuration, the token-holding node initiates a connection check sequence with the adjacent node, one for each group. The adjacent node then initiates the same sequence between it and the next adjacent node, if the previous check has been completed successfully. The connection check sequence is executed in this chain fashion in each group until it finally fails at the nodes adjacent to the failed segment. When the whole chain of check sequences has been completed, every node on the network has recognized the status of the connections of all four links between its adjacent nodes. This loopback reconfiguration algorithm does not need centralized control and is implemented by simply adding several states to the state diagram.

VI. MESSAGE TRANSMISSION

Error-free IINET messaging between transmitting and receiving nodes is achieved through the adoption of standard HDLC normal response mode (NRM) message transmission. HDLC NRM is defined in a network where a primary node is fixed and where a full duplex transmission channel is provided. After a node is identified as the token-holding node, the token-holding node is considered as a primary node of the HDLC NRM. By changing the switch to active mode at the designated receiver node, the ring constitutes a full duplex transmission channel between the primary node and the receiver.

Fig. 10 shows local RAM control. It provides transmit and receive buffers closely tailored to the HDLC's window control function. Two pointers of TX DMA and RX DMA have registers which store the pointers of acknowledged transmission, controlled by checkpointing, and send sequence number checks, respectively. The CPU write pointer has a pointer register which keeps the location of the last byte of the transmit message in the transmit FIFO and which provides dual buffer capability. The dual buffer function facilitates the microprocessor to write the next transmission message into the FIFO register, while the current message is in the process of transmission. The CPU read pointer has a delimiter stack which produces separating signals for as much as 31 messages in a 1-kbyte receive FIFO register. The microprocessor can read messages in the receive FIFO individually via this delimiter stack function.

In a typical LAN protocol implementation, simple LLC Class 1 service, unacknowledged connectionless service [12], is used in the logical link control sublayer, which has only transmission error detection and does not provide transmission error recovery. When a typical LAN is applied to real-time applications, a transport layer, which is at a higher level than the LSI's functions and produces a large overhead for small controllers, becomes mandatory. LLC Class 3 is intended to be applied for some real-time applications. It adds acknowledgment to LLC Class 1 service only, to provide faster response, however, it offers no recovery when notification of an acknowledgment is missed.

The HDLC NRM has an equivalent function of LLC Class 2, connection-oriented service, but in a more basic form. The HDLC NRM eliminates the possibility of message loss or duplication. Therefore, the IINET does not require the microprocessor to implement the transmission error recovery function as the upper layer services, as long as the communication is restricted to a single network. When multiple network configuration is required, the IINET needs the network layer services and, therefore, may need the transport layer services for end-to-end error recovery. The error recovery function in the transport layer may be omitted if the router has an adequate flow control to prevent message overflows.

To facilitate flow control implementation, the IINET provides a means for the microprocessor to cancel pending message transmission requests and to transmit flow control messages with the earliest possible opportunity, over all pending messages in the network.

The IINET provides broadcasting by designating the destination address as 256 (0xFF) and 15 multicast groups by designating the destination address as from 240 (0xF0) to 254 (0xFE). Destination addresses from 1 (0x01) through 239 (0xEF) are for point-to-point messages.

VII. SUMMARY

A network communication protocol with excellent applicability to a wide variety of control applications has been developed and implemented in silicon. It has been shown that priority token passing needs only simple circuitry, in addition to standard HDLC frame format control circuits. Because of the simplicity of the algorithm, elaborate message transmission functions can also be implemented on a relatively small-scale LSI chip. It is shown that all communication functions needed by real-time distributed controllers has been successfully implemented into a single chip.

The IINET supports a distributed control system architecture, which provides higher functionality, reliability, and safety than a stand-alone architecture. Its simplicity, requiring only minor components to provide a complete network communication function, has achieved reasonable communication cost ratio applicable to typical controllers. Liner reduction of cost range of network communication will result exponentially with an expanded number of controllers newly connected into a network. The IINET has considerably lowered the grade of controllers which can be connected into a real-time distributed control network and facilitated a variety of new distributed control applications in various automation areas.

The standard has to provide communication protocol options for various requirements. It has been proven in office automation areas that subnetworks of different lower-layer communication protocols can be interconnected cost-effectively by routers. New technology has to be added to the standard if an area of applications has not been adequately covered by the existing standard. Interoperability, which is more a matter of distributed application than communication technology, will be provided by the interconnection of subnetworks. The authors believe that the proposed priority token passing and the IINET will contribute to the communication protocol standardization and its popularization for real-time control applications.

REFERENCES

[1] K. Kobayashi, M. Kameyama, and T. Higuchi, "A token ring network for building automation," in *Proc. SICE'86 Conf.*, 1986, pp. 791–794.

[2] K. Kobayashi, M. Kameyama, and T. Higuchi, "New communication network protocol for cell level factory automation and its LSI implementation," in *Proc. IFIP CAPE'89*, pp. 565–572, 1988.

[3] *Token-Passing Bus Access Method and Physical Layer Specifications*, IEEE Std 802.4-1995 (ISO/DIS 8802/4).

[4] *Token Ring Access Method and Physical Layer Specifications*, IEEE Std 802.5-1985 (ISO/DIS 8802/5).

[5] *High Level Data Link Control Elements of Procedures*, ISO 4335, 1987.

[6] *TMS380 Adapter Chipset User's Guide*, Texas Instruments, Houston, TX, 1986.

[7] *High Level Data Link Control Procedures-Frame Structure*, ISO 3309, 1984.

[8] W. Bux, "Local-area subnetworks: A performance comparison," *IEEE Trans. Commun.*, vol. COM-29, pp. 1465–1473, Oct. 1981.

[9] A. Sethi and T. Saydam, "Performance analysis of token ring local area networks," in *Computer Networks and ISDN Systems 9.* Amsterdam, The Netherlands: Elsevier, 1985, pp. 191–200.

[10] R. Rom and N. Shacham, "A reconfiguration algorithm for a double-loop token-ring local area network," *IEEE Trans. Commun.*, vol. 37, pp. 182–189, Feb. 1988.

[11] *Fiber Distributed Data Interface (FDDI)-Token Ring Media Access Control (MAC1111)*, ISO 9314-2, 1989.

[12] *Logical Link Control*, IEEE Std 802.2-1985 (ISO/DIS 8802/2).

Koji Kobayashi was born in Nagano, Japan, in 1952. He received the B.E. and M.E. degrees in electronic engineering from Tohoku University, Sendai, Japan, in 1976 and 1978, respectively.

He is currently Manager of the Engineering Department, Building Systems Division, Yamatake-Honeywell Co. Ltd., Isehara, Japan. His general interests include distributed control system architecture, network communication protocol for real-time control applications, and pattern matching for finger print identification.

Michitaka Kameyama (M'79–SM'91) received the B.E., M.E., and D.E. degrees in electronic engineering from Tohoku University, Sendai, Japan, in 1973, 1975, and 1978, respectively.

He is currently a Professor in the Graduate School of Information Science, Tohoku University. His general research interests are intelligent integrated systems including robot electronics, VLSI processors for real-world applications, and multiple-valued VLSI systems.

Dr. Kameyama received the Outstanding Paper Awards at the 1984, 1985, 1987, and 1989 IEEE International Symposiums on Multiple-Valued Logic, the Technically Excellent Award from the Society of Instrument and Control Engineers of Japan in 1986, the Outstanding Transactions Paper Award from the Institute of Electronics, Information and Communication Engineers of Japan in 1989, and the Technically Excellent Award from the Robotics Society of Japan in 1990.

Tatsuo Higuchi (M'70–SM'83–F'92) received the B.E., M.E., and D.E. degrees in electronic engineering from Tohoku University, Sendai, Japan, in 1962, 1964, and 1969, respectively.

He was a Professor in the Department of Electronic Engineering, Tohoku University, where he is currently a Professor and Dean of the Graduate School of Information Science. His general research interests include the design of 1-D and multi-D digital filters, liner time-varying system theory, fractals and chaos in digital signal processing, VLSI computing structures for signal and image processing, multiple-valued IC's, multiwave opto-electronic IC's, interconnecion-free bio-electronic IC's, and fault-tolerant computing.

Dr. Higuchi served as the Program Chairman of the 1983 IEEE International Symposium on Multiple-Valued Logic (ISMVL) and as the Symposium Chairman of the 1992 IEEE ISMVL. He received the Outstanding Paper Awards at the 1984, 1985, 1987, and 1989 IEEE International Symposiums on Multiple-Valued Logic (with M. Kameyama *et al.*), the Outstanding Transactions Paper Award from the Society of Instrument and Control Engineers of Japan in 1984 (with M. Kameyama), the Technically Excellent Award from the Society of Instrument and Control Engineers of Japan in 1986 (with M. Kameyama *et al.*), the Outstanding Transactions Paper Award from The Institute of Electronics, Information and Communication Engineers of Japan in 1989 (with M. Kameyama *et al.*), the Technically Excellent Award from the Robotics Society of Japan in 1990 (with M. Kameyama *et al.*), and the IEEE Ambrose Fleming Premium Award in 1994 (with T. Aoki *et al.*). He is a member of the Society of Instrument and Control Engineers of Japan and the Robotics Society of Japan.