PHYSICAL LAYER REDUNDANCY METHOD FOR FAULT-TOLERANT NETWORKS IN PICNET

Jae Min Lee

Control Information Systems Lab., School of Electrical Engr. and ERC-ACI, Seoul National University, Seoul, 151-742, Korea

Abstract: In this paper, the physical-layer redundancy method is proposed for the fault-tolerant industrial network. The proposed method consists of the fault detection method and the correction method. The fault detection method uses events created by the state transition in the IEEE 802.4 MAC sublayer and the periodic status frame check for the detection of the fault. The fault correction method corrects the fault with the automatic physical layer switching to the stand-by physical layer due to the event created by the fault detection method. The proposed method is realized with dual physical layers, the Dual Channel Manager for switching and the Redundancy Management Module that has the Fault Detection Sub-module and the Fault Correction Sub-module. As results of the practical implementation, the proposed method guarantees high reliablility and fast fault-correction in PICNET.

Keywords: Physical layer redundancy, Fault tolerance, Industrial network, Redundancy management

1. INTRODUCTION

One of the most important changes in the process industry has been the innovation of industrial networks. The industrial network is the digital communication link among intelligent field-level devices. The distributed control system or the computer integrated manufacturing system is usually composed of field-level devices and the industrial network, which requires high reliability as well as availability. In the embedded system, the network is one of the most important elements since it takes charge of the exchange of information, such as control commands or measured values of sensors. However, even for highly reliable networks, faults are inevitable. Therefore, all of industrial networks have fault- tolerance, using H/W or S/W redundancy, such as dual systems, redundant network interface unit(NIU)s and N-version programs, etc(J. Laprie and Beoness, 1990; Moon and Kwon, 1998; Nelson, 1990).

The well-known fault-tolerant industrial networks are as follows. ADMAP duplicated the physical layer of a Mini-MAP system(Shiobara, 1987). Many Manufacturing Automation Protocol (MAP) applications duplicated the physical layer for fault-tolerance(Kleins and Zwoll, 1992; Moon, 1998). WorldFIP has achieved fault-tolerance by the medium redundancy(Laterrier, 1995). CENTUM uses the pair and spare redundancy that has two CPU boards and two CPUs in each board together with bus and I/O redundancy (CENTUM, 1998). On TTP, node failures and communication failures are masked by replicating nodes and two buses and sending the message twice on each bus(Gruensteidl and Kopetz, 1991; H. Kopetz and Krug, 1997; Pallierer and Fuchs, 1997). LONWORKS realized fault-tolerance by the self-healing ring that will automatically redirect signals in case of a medium's fault(Echelon, 1995). The NIU redundancy method that has the dual layer structure from the LLC sublayer down to the physical layer to cope with faults of those layers is proposed by Moon(Moon and Kwon, 1998). The fault detection method by the IEEE 802.4 media access control(MAC) protocol is also studied(Moon and Kwon, 1998; Moon, 1998).

Although these methods are good for the fault-tolerant system, a more highly reliable and faster method is required especially for the nuclear power plant or embedded systems. The NIU redundancy method is highly reliable but has a little time delay in switching to the stand-by board. The system redundancy method is also high reliable but has time delay in transferring previous information to the stand-by system. Therefore,

the faster and higher reliable fault detection and correction method is needed.

In this paper, a new physical layer redundancy method is proposed to satisfy these needs. The proposed method duplicates physical-layer and adds the Dual Channel Manager (DCM) and the Redundancy Management Module for the faster channel switching. For the fault detection, the proposed method uses events created by the state transition in the IEEE 802.4 MAC sublayer and the periodic status frame check. Then, the fault is corrected by the automatic physical layer switching to the stand-by channel during the state transition of the fault correction method.

For proving the validity of this method in a practical system, the proposed method is implemented in the Plant Instrumentation and Control Network(PICNET) that was developed by Seoul National University and KEPRI, Korea(S. Lee and Lee, 1999). As results of the implementation, the proposed method guarantees highly reliable and fast fault-correction. The PICNET will be used for the middle-level network of the Distributed Control System(DCS) of the nuclear power plant.

In Section 2, the faults covered by the proposed method are classified, and the proposed fault detection method is presented. The proposed fault correction method is shown in Section 3. The proposed method is implemented in PICNET in Section 4. Then, its time analysis is showed in Section 5. Finally, the conclusion is drawn in Section 6.

2. FAULT DETECTION METHOD

By the proposed method, faults are detected with two techniques. In first technique, faults are detected by events due to the state transition in the MAC sublayer. According to the IEEE 802.4 MAC protocol, events are created when faults are occurred in physical layer(ISO/IEC, 1990). Secondly, the periodic status frame check is used to prepare for the first technique's failure. For the fast and reliable fault detection, we use only six events, instead many events are occurred by the corruption of logical ring. If a fault is occurred in the data link layer, the first method may not detect the fault. At this time, the periodic status frame check technique detects a fault. In the proposed method, every nodes transmit its own status frame and receive the others' status frame. If one node had faults and did not transmit status frame, another node can detect the fault. In this section, the faults that can be detected by the proposed method are defined. Then, the fault detection method using events and status frame check is presented.

2.1 *Definition of faults*

In a network, each fault can be classified by its location as the physical layer faults, the transmitter and receiver faults, the upper layer faults, and noise. Medium open, short and imperfect medium connection are classified as the physical layer faults. The transmitter or receiver's malfunction is the transmitter and receiver faults. Noise can sometimes give distortion to the signal and make transient fault. Host's malfunction or interface error between the host and the NIU is classified as upper layer faults. The proposed method can detect and correct physical layer faults. The transmitter and receiver's fault and the upper layer fault also can be detected and they can be corrected by the system redundancy.

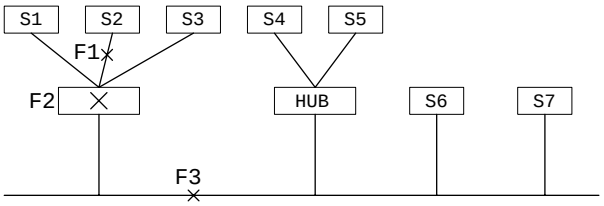

Fig. 1. Examples of faults in physical layer.

Figure 1 shows examples of physical layer faults when the network uses only single medium. When a medium open is occurred, the single node may be isolated or all nodes can be separated in several groups of nodes. F1 shows the single node's isolation. F2 also separates into single nodes, and F3 divides into two groups. These faults make network abnormal and the corruption of the logical ring by the IEEE 802.4 protocol communication. Therefore, faults in the physical layer can be detected by supervising the operation of the LLC sublayer because various errors are reported from the LLC sublayer.

2.2 *Fault detection method by events*

MC68824 token bus controller(TBC) is used as MAC controller (Motorola, 1987). The state transition in steady state of MAC is shown in Figure 2 and the names of states are listed in Table 1. When each NIU in a network is on or reset, the first state of NIU is OFFLINE. Then, if the token bus controller is initialized, the state transits to the IDLE state, and bus idle timer is started at that time. If no signals are detected on the bus, the timer is reset and restarted. If the value of the timer is larger than six or seven slot time, it can be regarded as no token in the logical ring, and the state transits to Claim_Token. In this state, the token contention process can transmit the predefined claim_token frame according to the address of MAC in the NIU. The survived NIU on the contention can achieve the token and the state

Table 1. Numbers and its corresponding states.

Number	State
0	OFFLINE
1	IDLE
2	DEMAND_IN
3	DEMAND_DELAY
4	CLAIM_TOKEN
5	USE_TOKEN
6	AWAIT_IFM_RESPONSE
7	CHECK_ACCESS_CLASS
8	PASS_TOKEN
9	CHECK_TOKEN_PASS
10	AWAIT_RESPONSE

transits to the Use_Token state. The state of the remained NIUs transits back to the IDLE state. The state of the NIU that received the token from the previous NIU, also transits to USE_TOKEN. In this state, the NIU checks whether the transmission time and frames in four transmit queue are remained and can transmit the frame by priority order. At this time, the NIU repeatedly exchanges state between AWAIT_IFM_RESPONSE and CHECK_ACCESS_CLASS. If the transmission time expired or no frames are remained in queue, the state transits to PASS_TOKEN state. If the NIU knows the next station, the NIU transmits the token frame. Then, the NIU waits in the CHECK_TOKEN_PASS state. Otherwise, the state of the NIU transits to the AWAIT_RESPONSE state(ISO/IEC, 1990).

In CHECK_TOKEN_PASS state, the successful transmission of the token is checked. When the transmission of token is finished, the state of the NIU transits to the IDLE state. Otherwise, the token is transmitted again. If the transmission of token is failed again, the NIU transmits who_follows frame and waits the predecessor of the next station's transmission in the AWAIT_RESPONSE state. If the transmission of token is failed continuously, the NIU transmits solicit_successor_2 frames and determines the next station. The NIUs received frames, such as who_follows, solicit_successor_1 or solicit_successor_2, transmits set_successor frame by the response window algorithm. If the NIU in the AWAIT_RESPONSE state receives the set_successor frame, it means the the NIU that has transmitted set_successor frame is determined as the new next station. Then, the state of the NIU in the AWAIT_RESPONSE state transits to IDLE. The token passing process is ended. If there is no set_successor frame in the physical layer, the state of the NIU in the PASS_TOKEN state transits IDLE and the no_successor_8 transition is logged to the network manager(ISO/IEC, 1990).

To detect faults in the physical layer, the state transition events in MAC occurred by the corruption of the logical ring are used. TBC provides the ability to monitor the events mentioned in

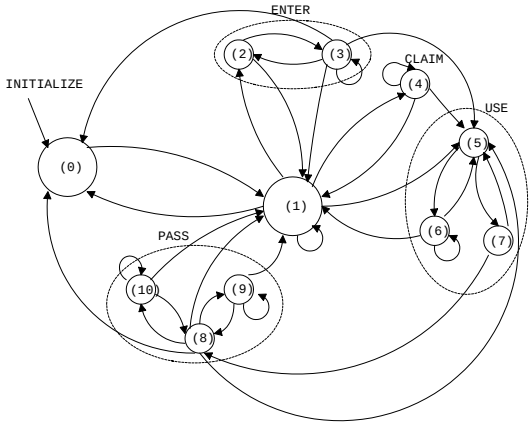

Fig. 2. State transition in IEEE 802.4 MAC.

Table 2. Interrupts and its transitions.

Interrupt	Transition
SC	new_successor
	won_contention
	hear_successor
NS	initialize
	exit_ring
	no_successor_8

the above by the interruption. TBC sets the bits of the interrupt word by these events. For the fast fault detection, we use only six events, such as new_successor, won_contention, hear_successor, initialize, exit_ring and no_successor_8. The first three events create SC(successor changed) interrupt, and the rest events NS(no successor) interrupt. In Table 2, the interrupts and the corresponding transitions are showed. Thus, because we programmed the interrupt service routine using only two interrupts, the proposed method is simple and fast to detect faults and implemented in the Fault Detection Sub-module.

2.3 *Fault detection method by status frame check*

The proposed method also has the other fault detection method, status frame check. Each station produces the status frame and broadcasts it periodically. All other stations' status frames are received and stored in a certain memory allocated three bytes per one station. Two bytes are for the data of the latest received status frame, and one byte is the previous value of the live counter. The received status frame contains the station's status, such as the live count value of the NIU, the status of its host, the status of two physical layers and the information of dual mode. Figure 3 shows the data in the status frame used in the proposed method. This frame is produced in the Status Frame Production Sub-module and checked in the Status Frame Check Sub-module.

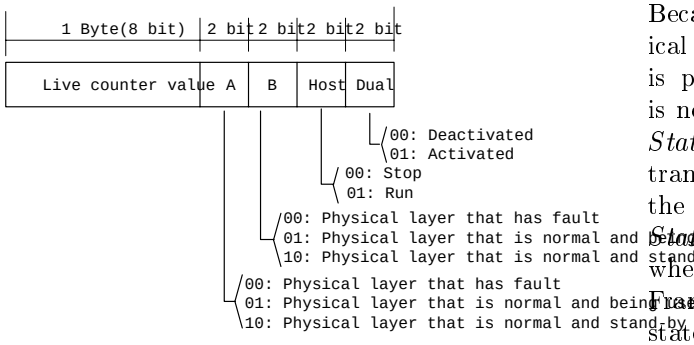

Fig. 3. Data in a status frame.

3. FAULT CORRECTION METHOD

After the fault is detected by the proposed method, the fault is corrected by the automatic physical layer switching. Then, the faulty NIU does its operation in normal status. For the fast fault correction, the station transmits frames to both physical layers and receives only one physical layer. The physical layer that uses in transmission and reception is defined as the active physical layer, and the physical layer using only in transmission as the stand-by physical layer. The proposed fault correction method is the event-based method. The state, that is specified by the condition of the physical layer and the status of dual mode, is transited by the events created in Fault Detection Sub-module. Figure 4 shows the state transition of the fault correction method.

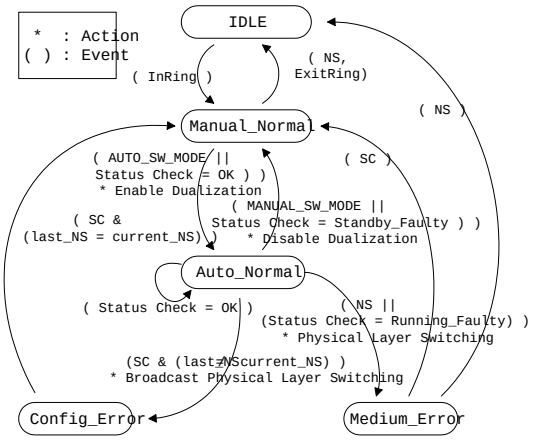

Fig. 4. Block diagram of state transition of the fault correction method.

The states of the proposed fault correction method consists of five states, such as Manual_Normal, Auto_Normal, Config_Error, Medium_Error and IDLE. If a NIU's power is off, the state of the NIU is IDLE. Then, the initialization procedure is finished and the state of the NIU transits to the Manual_Normal state. At this state, the logical ring may not be constructed perfectly and some other NIUs may not be finished its initialization procedure. But, the NIUs which are contained in the logical ring can communicate. Because the dual-mode is not activated, the physical layer is not changed. When the logical ring is perfectly constructed and the token passing is normally working or $AUTO\_SW\_MODE$ and $StatusCheck = OK$, the state of all the stations transits to Auto_Normal. $AUTO\_SW\_MODE$ is the command that the dual mode is activated. $StatusCheck = OK$ is the event that is occurred when both physical layers are good in the Status Frame Check Sub-module. In the Auto_Normal state, the dual mode is activated and the physical layer can be changed when a fault is detected in the active physical layer. If the stand-by channel is faulty, the $StatusCheck = Standby\_Faulty$ event is occurred and the state transits to Manual_Normal. Thus, the dual mode is deactivated. When the active physical layer becomes faulty and NS event is occurred, the state transits to the Medium_Error state, and the stand-by physical layer becomes the active physical layer.

The SC event means that the next station has changed. It can be occurred by the station's missing or having faults in the physical layer. Therefore, the state transits to the Config_Error state. During the transition, the station broadcasts the physical layer switching command to reconstruct the logical ring. At this time, the SC event is occurred in all stations, and the state of all stations transits to the Manual_Normal state. This state will not be changed until the faulty station is normalized again. After the faulty station is fixed, the $StatusCheck = OK$ event is occurred. Then, the state transits to the Auto_Normal state. If both physical layers have faults, the state transits to IDLE. This state will not be changed until the reconstruction of the logical ring is restarted.

4. IMPLEMENTATION IN PICNET

The proposed method is implemented in PICNET, a plant instrumentation and control network. Two physical layers are implemented in one NIU board, and the Redundancy Management Module in network management, the Dual Channel Manager are added. Figure 5 shows the implementation of the proposed method. The Dual Channel Manager switches the active physical layer to the stand-by physical layer by the events in the Fault Correction Sub-module. The Dual Channel Manager transmits data using both physical layers and receives signals from only one active physical layer. The Dual Channel Manager also monitors the status of the active physical layer and the stand-by physical layer that are recorded in the internal status register. This information is read by the Physical Layer Status Check Sub-module, and sent to the Status Frame Production Sub-module. The received status frames

from other stations are checked periodically in the Status Frame Check Sub-module. This sub-module creates events and sends them to the Fault Correction Sub-module. The Fault Detection Sub-module also creates events, such as NS and SC based on TBC's interruption and sends them to the Fault Correction Sub-module. The state transition in the Fault Correction Sub-module is event-triggered. During its transition, the action is executed. The result of the Fault Correction Sub-module is recorded and reported to the network manager. Based on this report, the network manager will fix the faulty physical layer or replace it as new one.

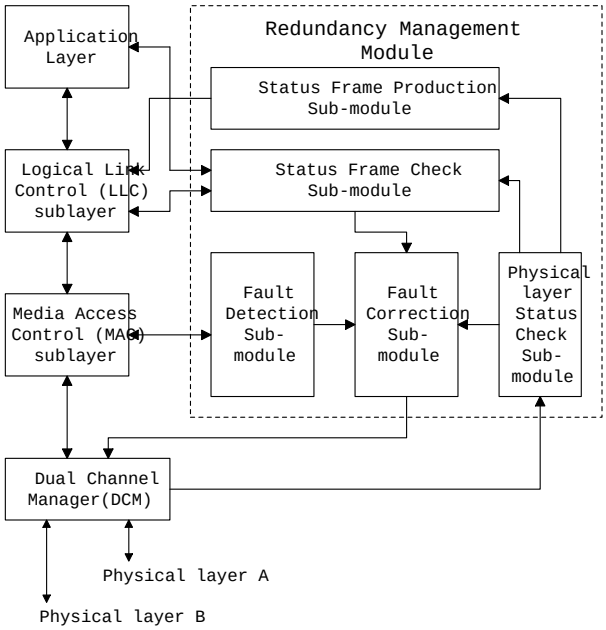

Fig. 5. Block diagram of the redundancy management module.

5. TIME ANALYSIS

In this section, the elapsed time of fault detection and fault correction using the physical layer switching is analyzed. The time was analytically calculated based on the PICNET spec. The elapsed time of the physical layer switching was measured in the prototype of PICNET. The PICNET spec. for analysis is as follows.

- The data rate: 5 Mbps
  (The MAC_Symbol_time : 0.2μs)
- 1 octet_time : 1.6μs
- 1 slot_time : 100 octet_time = 0.16 ms
- ring_maintenance time : 40 μs
- token frame size : 21 byte
  (the address : 2 byte)
- physical layer switching time : 350μs
- response window time : 1 slot_time

For minimum elapsed time calculation, we assume that the token-hold station transmits no frames except control frames and there is no delay in buffers or queues. According to the token and the number of stations, All of stations are divided 4 category, such as sole station with token, sole station without token, group stations with token and group stations without token. The stations without token create new token due to the IEEE802.4 MAC algorithm. Thus, fault detection time is longer than that of stations with token.

5.1 *Sole station with token*

The sole station with token try to pass the token but failed repeatedly. Then, the TBC of the station creates NS interrupt and the fault is detected. Next, the physical layer is switched to the standby physical layer. Finally, the logical ring is reconstructed. the elapsed time of fault detection($T_{s,t}$) is as follows:

$$\begin{aligned} T_{s,t} &= pass\_token + repeat\_pass\_token \\ &\quad + who\_follows + repeat\_who\_follows \\ &\quad + solicit\_any + no\_successor\_8 \\ &\quad + physical\ layer\ switching\ delay \\ &= 0.16ms \times (1+1+3+3+2) + 0.35ms \\ &= 1.95ms \end{aligned}$$

5.2 *Sole station without token*

In sole station without token, no_token transition is firstly occurred by bus_idle_timer_expired. Then the station creates new token in CLAIM_TOKEN state. Finally, the station failed to pass token and NS interrupt is occurred. The token_pass failed delay time was calculated in 5.1. The next procedure is same as that of sole station with token. The elapsed($T_{s,nt}$) is as follows:

$$\begin{aligned} T_{s,nt} &= bus\_idle\_timer\_expired \\ &\quad + token\ generation\ delay \\ &\quad + token\_pass\ failed\ delay \\ &\quad + physical\ layer\ switching\ delay \\ &= 0.16ms \times 7 \\ &\quad + (8 \times 0.16ms + 9 \times 0.0336ms) \\ &\quad + 1.6ms + 0.35ms \\ &= 4.6524ms \end{aligned}$$

5.3 *Group stations with token*

When the medium is cut and the logical ring is divided several groups, the elapsed time is varied by whether successor is in the same group or not.

When the token passed to the successor in the other group, the SC interrupt is occurred. If the next station of the successor is in the same group, the elapsed time($T_{g,t}$) is as follows:

$$\begin{aligned} T_{g,t} &= pass\_token + repeat\ pass\_token \\ &\quad + set\_successor\ reception\ delay \\ &\quad + physical\ layer\ switching\ delay \\ &= 0.16ms \times 2 + 0.0336ms + 0.35ms \\ &= 0.7036ms \end{aligned}$$

Otherwise, the elapsed time contains the contention time delay for won_patch transition. However, in calculation of the minimum elapsed time, the contention_time delay is assumed 0. Thus, the elapsed time is as follows:

$$\begin{aligned} T_{g,t} &= pass\_token + repeat\_pass\_token \\ &\quad + who\_follows + repeat\_who\_follows \\ &\quad + contention\_time\ delay \\ &\quad + set\_successor\ reception\ delay \\ &\quad + physical\ layer\ switching\ delay \\ &= 0.16ms \times (1 + 1 + 3 + 3) \\ &\quad + 0.0336ms + 0.35ms \\ &= 1.6636ms \end{aligned}$$

5.4 *Group stations without token*

At first, in group station without token, the token is generated like sole station without token. And, the contention timer delay is 1 slot_ time. The others is same in 5.2. The elapsed time ($T_{g,nt}$) is as follows:

$$\begin{aligned} T_{g,nt} &= bus\_idle\_timer\_expired \\ &\quad + token\ generation\ delay \\ &\quad + contention\_time\ delay \\ &\quad + set\_successor\ reception\ delay \\ &\quad + physical\ layer\ switching\ delay \\ &= 0.16ms \times 7 \\ &\quad + (8 \times 0.16ms + 9 \times 0.0336ms) \\ &\quad + 0.16ms + 0.0336ms + 0.35ms \\ &= 3.246ms \end{aligned}$$

6. CONCLUSIONS

In this paper, the new physical layer redundancy method is proposed for realization of the fault-tolerant network, and it is implemented in PICNET. The proposed method duplicates the physical layer and adds the Dual Channel Manager and the Redundancy Management Module. For the fault detection, the proposed method uses events created by the state transition in the IEEE 802.4 MAC and the periodic status frame check. Then, the fault correction method is executed using the automatic physical layer switching. Next, the network is back to normal state. As results of time analysis, we can show that the proposed method guarantees highly reliable and faster fault-correction. This method is also cost-effective because the proposed method realized fault-tolerance only by the physical layer redundancy. The proposed method and the system redundancy method can be simultaneously applied to the internal control network in an airplane or a ship for the higher reliability. In the future, the evaluation of the proposed method will be studied.

7. REFERENCES

CENTUM (1998). Centum cs3000 integrated production control system. *Yokogawa Internal Report.*

Echelon (1995). Lonworks for audio computer control network applications. *Echelon Internal Report.*

Gruensteidl, G. and H. Kopetz (1991). A reliable multicast protocol for distributed real-time systems. *8-th IEEE Workshop on Real-Time Operating Systems and Software, Atlanta, GA, USA.*

H. Kopetz, R. Hexel, A. Kruger D. Millinger R. Nossal A. Steininger C. Temple T. Fuhrer R. Pallierer and M. Krug (1997). A prototype implementation of a ttp/c controller. *SAE Congress and Exhibition, Detroit, MI, USA.*

ISO/IEC (1990). Iso/iec 8802-4, information processing system -local area networks - part 4: Token-passing bus access method and physical layer specifications. *IEEE Inc.*

J. Laprie, J. Arlet and C. Beoness (1990). Definition and analysis of hardware- and software-fault-tolerant architectures. *IEEE Computer, Special Issue on Fault-Tolerant Systems* **23**(7), 39–51.

Kleins, H. and K. Zwoll (1992). Map mining - a communications system for mining applications. *EMUG MAP/TOP EVENTS Conference Proceedings. SYSYTEC 92.*

Laterrier, P. (1995). The fip protocol. *WorldFIP internal report.*

Moon, H. (1998). Performance analysis and design of a communication network for industrial automation. *Ph. D. Dissertation, Seoul National Univ.*

Moon, H. and W. Kwon (1998). A fault detection and recovery mechanism for the fault tolerance of a mini-map system. *Journal of Control, Automation and Systems Engineering* **4**(2), 264–271.

Motorola (1987). Mc68824 user's manual. *Motorola Inc.*

Nelson, V. (1990). Fault-tolerant computing : Fundamental concepts. *IEEE Computer, Special Issue on Fault-Tolerant Systems* **23**(7), 19–25.

Pallierer, R. and E. Fuchs (1997). A tool for the evaluation of the ttp/c protocol. *8-th European Workshop on Dependable Computing, Experimental Validation of Dependable Systems, Goeteborg, Sweden.*

S. Lee, M. Yoon, H. Moon C. Shin and B. Lee (1999). Real-time characteristic analysis of a communication protocol for the distributed control system of npps. *Proceedings of the 14th KACC* pp. D102–D105.

Shiobara, Y. (1987). Advanced map for real-time process control. *Proceedings of IECIN87, Cambridge, Massachusett* pp. 883–891.