FACTA UNIVERSITATIS
Series: **Automatic Control and Robotics** Vol. 10, N° 1, 2011, pp. 115 - 123

ONE REALIZATION OF SWITCHED ETHERNET FAULT TOLERANT COMMUNICATION *

UDC 004.738.4 004.424.451.2

Sandra Đošić, Milun Jevtić

Department of Electronics, University of Niš, Faculty of Electronic Engineering,
Aleksandra Medvedeva 14, 18000 Niš, Serbia
E-mail: (sandra.djosic, milun.jevtic)@elfak.ni.ac.rs

Abstract. *Switched Ethernet network has been considered as a promising network technology for industrial control and monitoring systems. These systems work in real-time so they need high communication reliability. This paper presents one realization of fault tolerant communications over the switched Ethernet. We assumed that faults occur on transmission links and that the network recovers from the faults retransmit frame again. In that situation all real-time transmission requirements must be met. We analyze the possibility to overcome faults using time redundancy. We also propose an algorithm for fault tolerant real-time communication on a switched Ethernet.*

Key words: *Switched Ethernet, Fault tolerant communication, Time redundancy*

1. INTRODUCTION

Nowadays, embedded systems (modern vehicles, radar systems, telecommunication systems, wireless communication systems, automation systems) are becoming more and more complex, distributed and interconnected. There are constant demands to increase the capacity ("bandwidth"), the heterogeneity or the quality of service (QoS). At the same time, there is the ongoing need to reduce the cost of the whole systems. All these facts lead to the idea to use the standard networks (low cost), but with added intelligence and support for the different services required in embedded systems. Ethernet, as an internationally standardized local area network technology [1], is becoming very popular for use for that purpose. In most embedded systems, support for real-time communication with various demands, and/or with time-deterministic behavior is necessary. Ethernet does not guarantee a real-time delivery of messages due to collisions on the network that occur when more than one node transmit messages at the same time.

Received June 03, 2011
* **Acknowledgement**. *This paper is supported by Project Grant III44004 (2011-2014) financed by Ministry of Education and Science, Republic of Serbia.*

As Ethernet can easily be implemented as a pure switched-based network, the collision possibility can be eliminated. Beside that, the involvement of the switches increases the performance of the network and offers the possibility of real-time services such as providing a large amount of bandwidth, micro-segmentation of traffic on the network, and full-duplex links [2]. Because of that, switched Ethernet has been considered as a promising network technology for real-time communications required by the real-time applications, [3], [4]. As switched Ethernet can be used in industrial systems, then it is necessary to develop appropriate protocols for such specialized field of use, [5]. These systems often require hard real-time communication and fault tolerant protocols. Industrial reliable real-time communication on a switched Ethernet is the focus of our research.

In recent years, several protocols to support real-time communication on the switched Ethernet have been proposed. The Ethernet solution, such as Ethernet Powerlink [6], PROFINET [7], Ethernet/IP [8], EtherCAT [9], usually override the standard or reach their limitations when a certain degree of real-time capability is required. The techniques proposed in [10], [11], [12] require some real-time features to be added to the end nodes and/or to the switch in order to regulate the amount of traffic on the network and to control the overrun of the output queue of the switch. In [10] one hard real-time communication method has been proposed. The method is based on EDF (Earliest Deadline First) scheduling over the switched Ethernet. The approach assumes that both end nodes and the switch can schedule messages according to the EDF policy, which requires the addition of RT (Real-Time) layer to support the EDF scheduling above the MAC layer both on the nodes and on the switch. Traffic shaping techniques have been proposed in [11] and [12] to provide a real-time communication by limiting the amount of traffic on the network. Their methods only show that the maximum delay on the network is bounded without considering the explicit deadlines of messages. In [13] one message transmission model, FTT-Ethernet, has been presented. This model supports dynamic real-time message requirements on Ethernet. The method uses a synchronized message transmission based on a master-slave model. The master checks the feasibility of a new message in response to the request of the transmission of the message from the slaves (makes a feasible schedule for the message) and sends the schedule in trigger message to the slaves.

In this paper, we will present a frame (message) transmission model for hard real-time communications of periodic message on a switched Ethernet with possibility to tolerate some fault on links transmission. The proposed model was inspired by [14] and we present its modification related to fault tolerance. We will show how time redundancy can be used for fault tolerance.

The rest of the paper is organized as follows. In section 2, the message transmission model is discussed. In section 3, the proposed fault tolerant communication model is described. The last section offers our conclusions.

2. COMMUNICATION MODEL

We consider a network with topology of a switched Ethernet and end-nodes, Fig. 1. The switch in the network is a standard one; there is no need for modification of its operation. The links between the switch and end-nodes are full-duplex and consist of transmission (TL) and reception links (RL). Both transmission and reception links operate independently and the switch uses a store-and-forward switching mode for forwarding frames from the input ports to the output ports. In the store-and-forward switching, the switch determines the output port and forwards the frame to the output port after receiving the entire frame, [15].

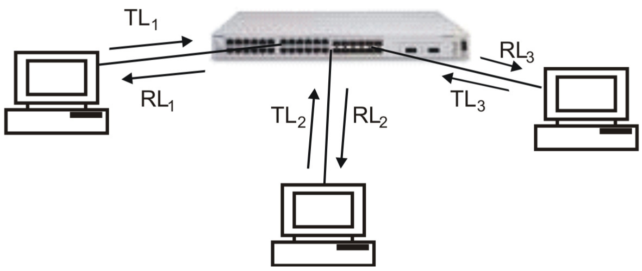

Fig. 1 Switched Ethernet network

Latency on the switched Ethernet network, [16], is a very important fact for the real-time communications. If there is no collision, then the end-to-end communication delay (delay from the source to the destination node L_{total}) can be defined as follows:

$$L_{total} = L_{sf} + L_{sw} + 2*L_{wl} \tag{1}$$

L_{sf} is latency which refers to the basic operating principle of an Ethernet switch: the switch stores the received data in memory until the entire frame is received and then transmits the data frame out the appropriate port. This latency is proportional to the size of the frame being transmitted (*FS*) and inversely proportional to the bit rate (*BR*).

$$L_{sf} = FS / BR \tag{2}$$

L_{sw} is delay caused by the internals of an Ethernet switch known as the *switch fabric*. The switch fabric consists of sophisticated silicon that implements the store and forward engine, MAC address table, VLAN, and CoS, among other functions. This fabric introduces delay (L_{sw}) when executing the logic that implements these functions.

L_{wl} is propagation delay for the electrical signal to propagate from the source to the switch, which is proportional to the length of the cable connecting the source and the switch.

According to [14], the switching delay depends on the switch vendor, but usually it is about 10μs in 100Mbps switches. L_{wl} is less than 1μs in a 100m link. In the case of the minimum size frame and 100Mbps switches, the end-to-end communication delay is about 20μs.

We assume that all of the nodes are synchronized and both the transmission link and the reception link are composed of a sequence of time slots (TS). Each time slot is further divided into smaller basic frame transmission units (FTU) as shown in Fig. 2.

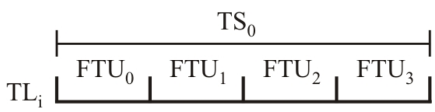

Fig. 2 Link structure

All of the periodic messages have their strict message transmission deadlines and the required network bandwidth should be allocated to the messages before transmission in order for them to be delivered to the intended destination node within their deadlines. The length of FTU is determined according to the ratio of the amount of message traffic of periodic messages.

A periodic message PM_{ij} from node N_i to N_j has a real-time transmission requirement $\{P_{ij}, D_{ij}, C_{ij}\}$, where P_{ij} denote the period, D_{ij} denote the deadline and C_{ij} denote the length of the message. We assume that

$$P_{ij} = D_{ij} = k*FTL \tag{3}$$

where *FTL* is the length of FTU and *k* is an integer number. We also assume that the real-time transmission requirements of all of the periodic messages are known a priori, and the number of FTU in TS is

$$n = \mathrm{LCM}\ (P_{ij}/FTL) \tag{4}$$

LCM means the least common multiple.

In our network, each end-node maintains a transmission message schedule which contains an order of frames to be transmitted on links.

Let us suppose that node N_1 wants to send a periodic message to node N_2, N_3 and N_4. These messages are marked with PM_{12}, PM_{13}, and PM_{14} respectively. Real-time transmission requirements for these messages are shown in Table 1.

Table 1. Real-time transmission requirements

Message	Real-time transmission requirement P_{ij}	D_{ij}	C_{ij}
PM_{12}	2	2	0.25
PM_{13}	3	3	0.75
PM_{14}	6	6	1

If we assume that *FTL* = 1, then in accordance with real-time parameters of messages PM_{12}, PM_{13}, and PM_{14} *n* is 6. Because of that, the time slot TS_0 of link TL_1 is divided on 6 frame transmission units FTU_0 to FTU_5.

The structure of the transmission link and message schedule is shown in Fig. 3. This is an example of transmission link TL_1 (from node N_1 to the switch) and its time slot TS_0.

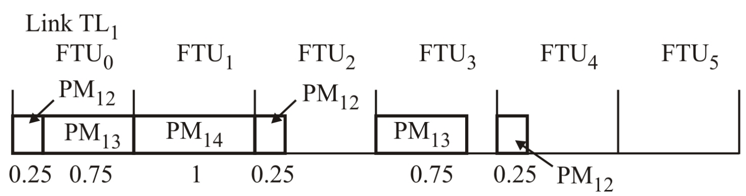

Fig. 3 One example of message schedule on one transmission link

All real-time transmission requirements of the messages are met, so this is an example of successfully communication.

We adopt that the messages on the links are scheduled by the algorithm proposed in [8]. By means of that algorithm, the free space on the link is occupied to meet message real-time transmission requirements in the order of the appearance of the messages.

3. Proposed Fault Tolerant Communication Model

An important fact about the algorithm proposed in [8] is that all end-nodes in the network know when their transmission and reception links when they are occupied or not. We use that fact to modify the algorithm to tolerate a fault on links transmission.

Sometimes, due to different external effects on transmission links (e.g. electromagnetic), frames could be corrupted in transit. The combination of two fundamental mechanisms *acknowledgement* and *timeout* could give information about reliable transmission. An acknowledgment (ACK) is a small control frame that a protocol sends back to its node saying that it has received an earlier frame. The receipt of an acknowledgment indicates to the sender of the original frame that its frame was successfully delivered. If the sender does not receive an acknowledgment after a reasonable amount of time (timeout), then it retransmits the original frame. The retransmited frame must meet all its real-time transmission requirements, as shown in Fig. 4.

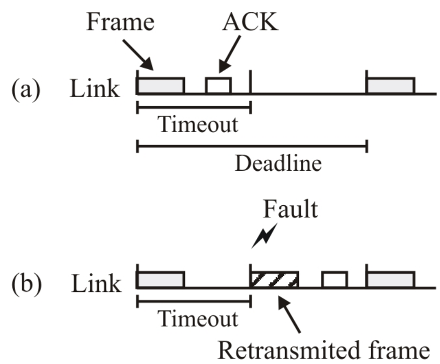

Fig. 4 Acknowledgement and timeout mechanisms (a) acknowledgment arrives before timeout, (b) acknowledgment does not arrive before timeout and the sender must retransmit the frame

We propose using time redundancy for message retransmissions. We consider that every frame has its own acknowledgment and timeout. In Fig. 4(a), it can be seen that acknowledgment arrives before timeout and that frame deadline is met. So, it is an example of a reliable transmission. If the acknowledgment does not arrive before timeout then we assume that transmission fault occurred, Fig. 4(b). Right after timeout, the sender must retransmit the original frame if there is enough free time for that. After that, the sender waits for acknowledgement of the retransmitted frame. In Fig. 4(b) the retransmitted acknowledgment arrives before timeout and also frame real-time requirements are met. This is also an example of successful communication.

For ease of description, we adopt that the time interval needed for frame transmission, waiting for acknowledgment and acknowledgment itself, present as one message. This is illustrated in Fig. 5.

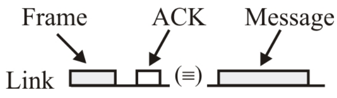

Fig. 5 Frame, waiting for acknowledgment time interval and acknowledgment present as one message

The main problem with retransmission is connected with the length of free time interval. So, it is very important to keep information about the free and/or occupied space on the links. We adopt that every node has this information in the form of lower (*LBE*) and upper (*UBE*) edges of free bandwidth.

We use the fact that all nodes in the network have information about lower and upper edges of free bandwidth for solving the problem of transient fault tolerance. We proposed a transient fault tolerant algorithm for real-time communication process in one Ethernet switched network presented in Fig. 6.

In general, the communication process has two parts. The first one, when there are no faults on the network links and when there is no need for communication modification, step (1). The second one, when a fault on the link occurs and when is necessary to modify communication, from step (2) to step (6). The second part is shown in detail in Fig. 6.

When there is a fault on the link, step (2) appropriate node in the network first has to check free bandwidth on that link, step (3). If such bandwidths exist, then the node has to calculate their lengths. Step (4) checks if free bandwidth equal or greater than the length of the corrupted frame, i.e. if the condition

$$UBE_{ik} - LBE_{ik} \geq C_{ij} \tag{5}$$

is true. UBE_{ik} and LBE_{ik} are upper and lower edges of free bandwidth for link L_i which has m free intervals, so k is a number between 1 and m. If the condition in step (4) is true, then the appropriate node has to calculate if the lower edge of that bandwidth plus length of the corrupted frame is less than or equal to the deadline of the corrupted frame, step (5)

$$LBE_{ik} + C_{ij} \leq D_{ij} \tag{6}$$

If it is true, then it is possible to retransmit the corrupted frame and meet all real-time requirements. The start retransmission time will be lower edge of that bandwidth, LBE_{ik}. After retransmission the communication process is continued normally.

If some of these conditions are not true, then retransmission can not be done. In that case, the node has the information about unsuccessful frame transmission and also about unsuccessful fault tolerance. That situation is not good for the real-time communication and in practice, it should not happen.

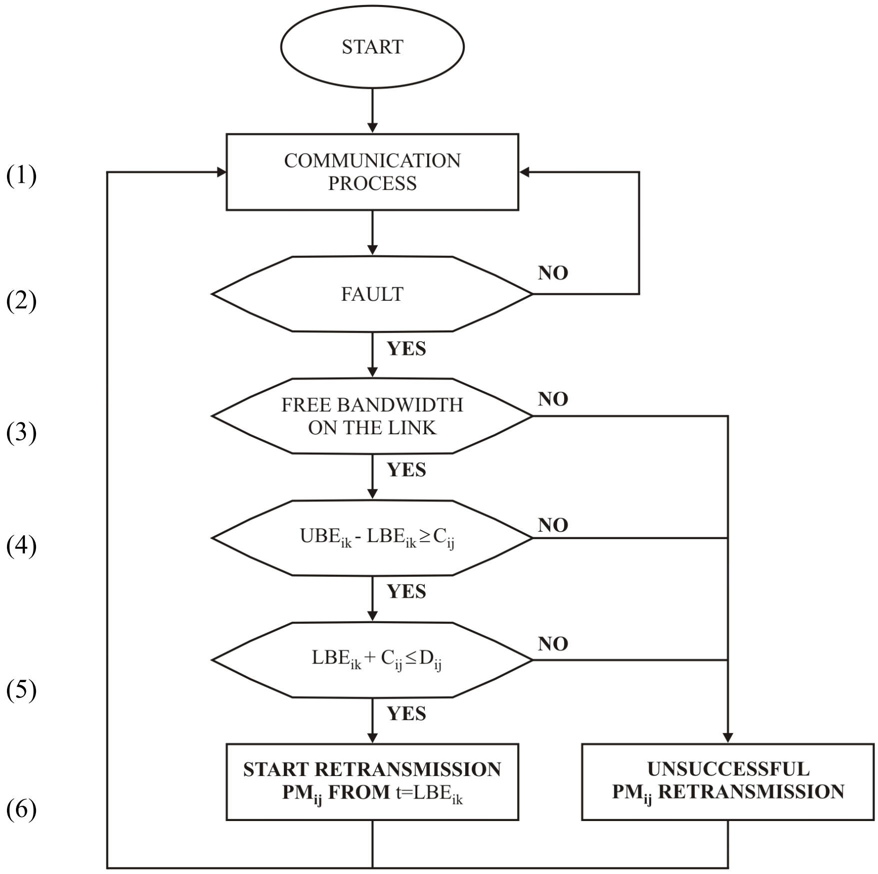

Fig. 6 Proposed transient fault tolerant algorithm for real-time communication process in one Ethernet switched network

We illustrated the proposed algorithm by giving one simple example. Let us consider the same example from Table 1 and suppose that a fault occurs in FTU_0 during frame PM_{13}. It means that acknowledgment did not arrive before timeout and that it is necessary for node N_1 to start the procedure for retransmission. The most important information for retransmission is presented in Table 2 and this is the information about the occupied space on the links.

Table 2 Occupied space on the links before fault

Message	before the fault					
	FTU_0	FTU_1	FTU_2	FTU_3	FTU_4	FTU_5
PM_{12}	0-0.25		0-0.25		0-0.25	
PM_{13}	0.25-1			0-0.75		
PM_{14}		0-1				

Node N_1, according to the proposed algorithm and information presented in Table 2, first has to check free intervals on the link. Link TL_1 has 3 free intervals: in the FTU_2,

FTU_3, FTU_4 and FTU_5. Free intervals in the FTU_2, FTU_4 and FTU_5 are long enough for frame PM_{13} retransmission. According to message PM_{13} deadline, only time bandwidth FTU_2 is appropriate for retransmission. The occupied space on the links during retransmission is shown in Table 3.

Table 3 Occupied space on the links during retransmission

Message	after fault					
	FTU_0	FTU_1	FTU_2	FTU_3	FTU_4	FTU_5
PM_{12}	0-0.25		0-0.25		0-0.25	
PM_{13}	0.25-1		0.25-1	0-0.75		
PM_{14}		0-1				

Messages on the link TL_1 would be scheduled as it is shown in Fig. 7. Retransmitted frame occupied the first available and appropriate frame - FTU_2.

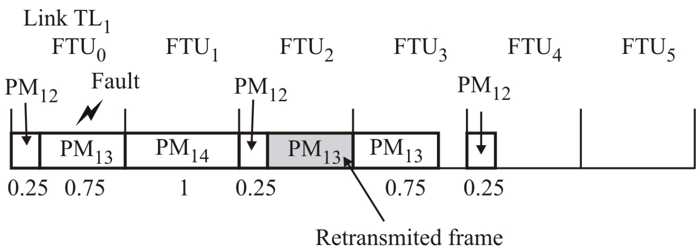

Fig. 7 Order of frames transmission when a fault occurs

All message real-time transmission requirements are met and this is an example of successful communication.

4. CONCLUSION

In this paper, we have presented a switched Ethernet based network concept supporting hard real-time communication. We presented one communication model with the ability to tolerate transient faults which appear on communication links. In the proposed solution, we describe how time redundancy can be used for fault tolerance. The main idea is to repeat the transmission of the message which is affected by the fault. Free time intervals on the links are used for message retransmission. Special attention is focused to the fact that, during retransmission, all real-time transmission requirements must be met.

We also considered, in this paper, the relation between time redundancy and transmission fault frequency. It is easy to conclude: if time redundancy is bigger, the possibility for multiple retransmissions is more possible. If the multiple retransmissions are more possible then real-time communication is more faults tolerant. The tendency is to use time redundancy for improved tolerant fault issued on communication links as better as possible. All of that means reliable communication, which is the most important thing in hard real-time applications.

REFERENCES

1. L. R. Peterson, B. S Davie, Computer Networks: A systems approach, third edition, Morgan Kaufmann Publishers, 2003.
2. J. Jasperneit, P. Neumann, "Switched Ethernet for factory communication", Proceedings of ETFA - 8th IEEE Int'l Conf. on Emerging Technologies and Factory Automation, Antibes, France, 2001.
3. N. Nissanke, Realtime Systems, Prentice Hall, 1997.
4. J. C. Laprie, Dependability: Basic Concepts and Terminology, Springer-Verlag, 1992.
5. M. Jevtić, M. Cvetković and S. Brankov, "Task Execution in Real-Time Systems for Industrial Control and Monitoring", Electronics, Faculty of Electrical Engineering University of Banjaluka, vol. 6, no. 2, pp. 56-61, december 2002.
6. Ethernet Powerlink, available at: http://www.ethernet-powerlink.com.
7. EtherCAT, available at: http://www.ethercat.org.
8. Profinet IRT, available at: http://www.profibus.com
9. Ethernet/IP Specification, Release 1.0, vol. 1-2. ControlNet International and Open DeviceNet Association, June 2001.
10. H. Hoang, M. Jonsson, U. Hagström, A. Kallerdahl, "Switched Real-Time Ethernet with Earliest Deadline First Scheduling - Protocols and Traffic Handling", Proceedings of the International Parallel and Distributed Processing Symposium, vol.2, pp0094b, IPDPS 2002 Workshops, 2002.
11. J. Loeser, H. Haertig, "Low-latency Hard Real-Time Communication over Switched Ethernet", Proceedings of the 16th Euromicro Conference on Real-Time Systems, ECRTS04, Catania, Italy, June 2004.
12. S. Varadarajan, T. Chiueh, "EtheReal: A Host-transparent Real-Time Fast Ethernet Switch", Proceedings of the Sixth International Conference on Network Protocols, October 1998.
13. P. Pedreiras, L. Almeida, P. Gai, "The FTT-Ethernet protocol: Merging flexibility, timeliness and efficiency", Proc. ECRTS (EUROMICRO Conf. Real-Time Systems), pp. 134—142, 2002.
14. H. C. Lee, M. K. Kim, "A scheduling Algorithm of Periodic Messages for Hard Real-Time Communications on a Switched Ethernet", International Journal of Computer Science and Network Security, IJCSNS, vol.6, No.5B, May 2006.
15. Kyung Chang Lee, Suk Lee, "Performance evaluation of switched Ethernet for real-time industrial communications", Computer Standards & Interfaces, Volume 24, Issue 5, pp. 411– 423, November 2002.
16. Application Note 8, "Latency on a Switched Ethernet Network", RuggedCom – Industrial Strength Networks, Ontario, pp. 1-6, April 2008.

JEDNA REALIZACIJA ETERNET SWITCH KOMUNIKACIJE KOJA IMA MOGUĆNOST PREVAZILAŽENJA OTKAZA

Sandra Đošić, Milun Jevtić

Eternet mreže koje koriste switch-eve su sve zastupljenije u industrijskim upravljačko-nadzornim sistemima. Uz rad u realnom vremenu od njih se zahteva da sa visokom pouzdanošću ostvare komunikaciju. Zato je u radu predložen jedan način realizacije komunikacije koja ima i mogućnost tolerisanja otkaza. Pretpostavlja se da se otkazi mogu desiti na prenosnim linkovima i da se mreža oporavlja tj. toleriše otkaze ponovnim slanjem paketa uz zadovoljenje vremenskih ograničenja rada u realnom vremenu. Analizirana je mogućnost prevazilaženja otkaza korišćenjem vremenske redundanse. U radu je predstavljen i algoritam koji opisuje ceo proces komunikacije u realnom vremenu sa mogućnošću tolerisanja otkaza.

Ključne reči: *Switch eternet, Komunikacija sa mogućnošću prevazilaženja otkaza, Vremenska redundansa*