Switched Real-Time Ethernet with Earliest Deadline First Scheduling – Protocols and Traffic Handling

Hoai Hoang, Magnus Jonsson, Ulrik Hagström, and Anders Kallerdahl

School of Information Science, Computer and Electrical Engineering, Halmstad University, Halmstad, Sweden, Box 823, S-301 18, Sweden. {Hoai.Hoang, Magnus.Jonsson}@ide.hh.se, http://www.hh.se/ide

Abstract

There is a strong interest of using the cheap and simple Ethernet technology for industrial and embedded systems. This far, however, the lack of real-time services has prevented this change of used network technology. This paper presents enhancements to full-duplex switched Ethernet for the ability of giving throughput and delay guarantees. The switch and the end-nodes controls the real-time traffic with Earliest Deadline First (EDF) scheduling on the frame level. No modification to the Ethernet standard is needed in the network that supports both real-time and non-real-time TCP/IP communication. The switch is responsible for admission control where feasibility analysis is made for each link between source and destination. The switch broadcasts Ethernet frames regularly to clock synchronize the end nodes and to implement flow control for non-real-time traffic.

1 Introduction

This paper focus on how to form methods to be able to support typical industrial real-time traffic without changing the underlying protocols and while still supporting existing higher-level protocols for non-real-time traffic (e.g., web based maintenance which is highly desirable to coexist with the real-time traffic).

An important trend in the networking community is to involve more switches in the networks (e.g., LAN, Local Area Networks) and a pure switched-based network becomes more and more common. At the same time, the industrial communication community has a strong will to adapt LAN technology (e.g. Ethernet) for use in industrial systems. The involvement of switches does not only increase the performance; the possibility to offer real-time services is also improved. Now when the cost of LAN switches has reached the level where pure switched-based networks have become affordable, the collision possibility in IEEE 802.3 (Ethernet) networks can be eliminated and methods to support real-time services can be implemented in the switches without changing the underlying widespread protocol standard.

Several protocols to support real-time communication over shared-medium Ethernet have been proposed [1] [2] [3]. However, these protocols are either changing the Ethernet standard or do not add guaranteed real-time services. Real-time communication over switched Ethernet has also been proposed (called EtheReal) [4]. The goal of the EtheReal project was to build a scaleable real-time Ethernet switch, which support bit rate reservation and guarantee over a switch without any hardware modification of the end-nodes. Ethereal is throughput oriented which means that there is no or limited support for hard real-time communication, it has no explicit support for periodic traffic so it is not suitable for industrial applications. A review of research on real-time guarantees in packet-switched networks is found in [5].

This paper presents a switched Ethernet network with support for both bit rate and timing guarantees for periodic traffic. Only a thin layer is needed between the Ethernet protocols and the TCP/IP suite in the end-stations. The switch is responsible for admission control, while both end-stations and the switch have EDF (Earliest Deadline First) scheduling [6]. Internet communication is supported at the same time as nodes connected to the switch can be guaranteed to meet their real-time demands when they communicate with each other. This is highly appreciated by the industry since it makes remote maintenance possible, e.g., software upgrades or error diagnostics (see Figure 1). Some implementation experiments have been done [7], but are not covered in this paper.

The rest of the paper is organized as follows. The network architecture is presented in Section 2. In Section 3, it is described how real-time channels are setup and how real-time traffic is treated. Deadline scheduling, including details on the admission control, is then presented in Section 4. The paper is concluded in Section 5.

1530-2075/02 $17.00 © 2002 **IEEE**

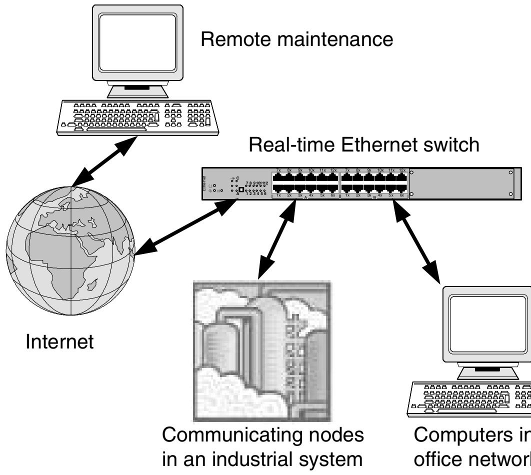

Remote maintenance
Real-time Ethernet switch
Internet
Communicating nodes in an industrial system
Computers in an office network

Figure 1: Both Internet traffic and industrial real-time traffic are supported.

2 Network architecture

We consider a network with the topology of a switched Ethernet (a single switch is assumed in this paper) and end-nodes. Both the switch and the end-nodes have software (RT layer) added to support guarantees for real-time traffic. Every node is connected to other nodes via the switch and nodes can communicate with each other over logical real-time channels (RT channels), each being a virtual connection between two nodes in the system (see Figure 2). In our network configuration, end-nodes have the capability of controlling traffic from the nodes using the Earliest Deadline First (EDF) algorithm. The switch has the same capability.

Full-duplex switched Ethernet is assumed for the network, which supports both real-time and non-real-time traffic. MAC function, frame buffering and centralized transmission arbitration are included in the switch. Non-real-time frames are redirected based on the MAC destination address. An Ethernet switch must contain address table, address learning and other functions needed to support the standard switching. How real-time frames are treated is discussed in the next section.

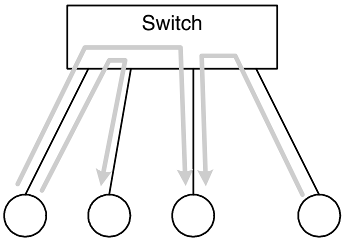


Figure 2: Example of a switched network with some real-time and/or non-real-time channels.

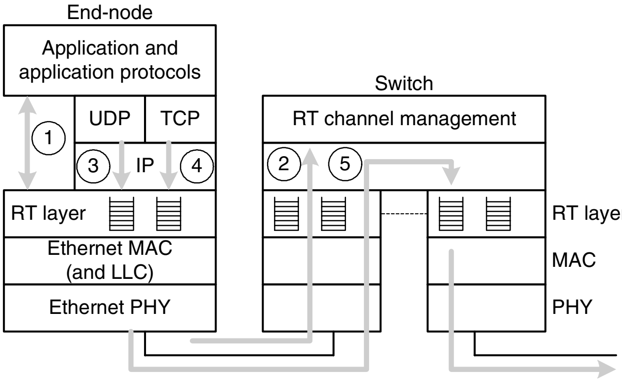


Figure 3: Layers and output queues.

The switch periodically sends synchronization frames to the end-nodes, at an interval, T_{cycle}, of ten maximum sized frames, T_{frame}, i.e.,

$$T_{cycle} = 10T_{frame}. \tag{1}$$

In this way, every node has a uniform comprehension about global time. The resolution of the global time is T_{frame}. In this paper, we assume Fast Ethernet (100 Mbit/s) with a maximum frame size of 12 144 bits which, with some extra time for timing uncertainties and for simplicity, gives $T_{frame} = 125$ μs, which just happens to match the time resolution of many telecommunication systems. The synchronization frames also give flow-control information for non-real-time traffic, i.e., telling the buffer status of the switch. For real-time traffic, the switch always checks, at RT channel establishment, if there is enough buffer space.

The function of and interaction with the RT layer etc shown in Figure 3 is explained below. When an application wants to setup an RT channel, it interacts directly with the RT layer (1). The RT layer then sends a question to the RT channel management software in the switch (2). Outgoing real-time traffic from the end-node uses UDP and is put in a deadline-sorted queue in the RT layer (3). Outgoing non-real-time traffic from the end-node typically uses TCP and is put in a FCFS-sorted (First Come First Serve) queue in the RT layer (4). In the same way, there are two different output queues for each port on the switch too (5).

3 Real-time communication

Below, real-time channel establishment and real-time traffic handling, are discussed respectively. One aim of the section is to explain the function of the switch (see Figure 4 for a flow diagram for the switch) and the end-nodes, in terms of real-time communication support.

3.1 Real-time channel establishment

Before real-time traffic may be delivered, an RT channel must be established. The creation of an RT channel

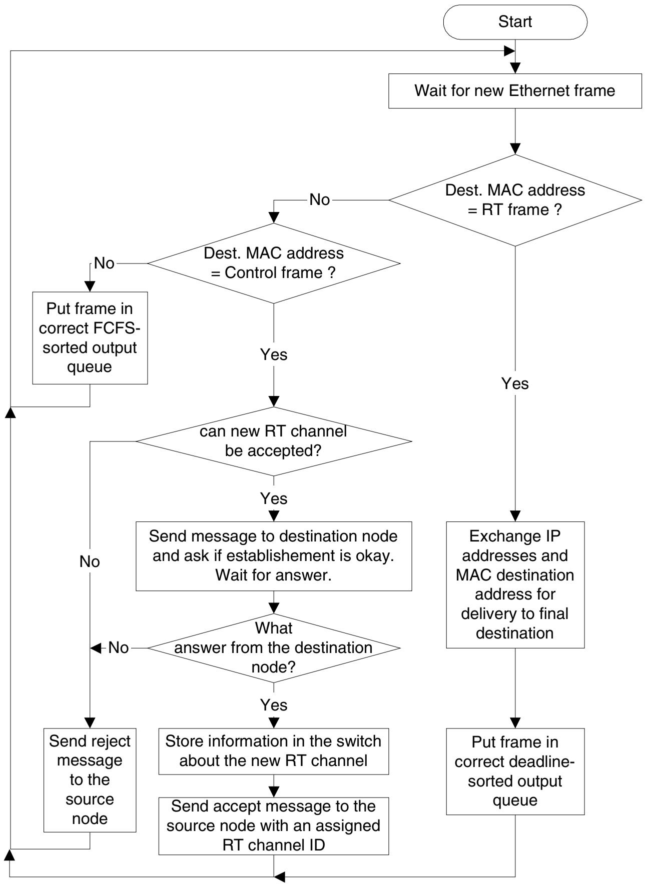

Start
Wait for new Ethernet frame
Dest. MAC address = RT frame ?
No
Dest. MAC address = Control frame ?
No
Put frame in correct FCFS-sorted output queue
Yes
Yes
can new RT channel be accepted?
Yes
No
Send message to destination node and ask if establishement is okay. Wait for answer.
Exchange IP addresses and MAC destination address for delivery to final destination
What answer from the destination node?
No
Yes
Send reject message to the source node
Store information in the switch about the new RT channel
Send accept message to the source node with an assigned RT channel ID
Put frame in correct deadline-sorted output queue

Figure 4: Flow diagram for the switch.

consists of request and acknowledgment communication where the source node, the destination node, and the switch agree on the channel establishment. After that, the nodes can begin to use the channel. Both the switch and the end-nodes have software (RT layer) added which shapes the traffic on the RT channel.

An RT channel with index *i* is characterized by:

$$\{T_{period,i}, C_i, T_{deadline,i}\} \qquad (2)$$

where $T_{period,i}$ is the period of data, C_i is the amount of data per period, and $T_{deadline,i}$ is the relative deadline used for the end-to-end EDF scheduling. Both $T_{period,i}$, C_i, and $T_{deadline,i}$ are expressed as the number of maximal sized frames, i.e., the number of T_{frame}. In this paper,

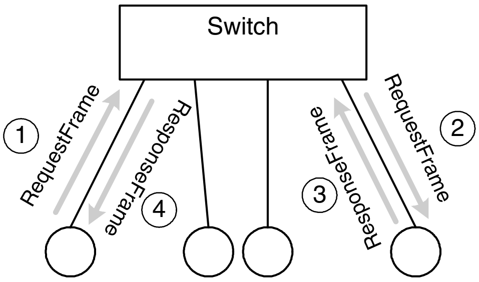


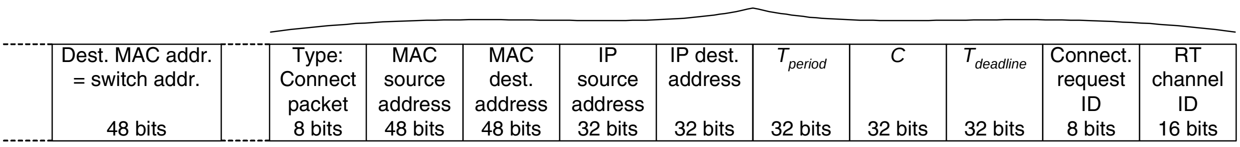
Figure 5: Establishment of an RT channel.

$$T_{deadline,i} = T_{period,i} \qquad (3)$$

is assumed. When an RT channel has been established, the network guarantees to deliver each generated message with a bounded delay, $T_{max\_delay,i} = T_{deadline,i} + T_{latency}$ (see next section), expressed in number of T_{frame}. When a node wants to establish an RT channel, it sends a RequestFrame to the switch (see Figure 5), which includes (see Figure 6): source and destination node MAC and IP addresses and $\{T_{period,i}, C_i, T_{deadline,i}\}$. The connection ID field is set to a source-node unique ID for the ability to distinguish the response in the case of several requests. The RT channel ID field is not set with a valid value yet.

When receiving a RequestFrame, the switch will calculate the feasibility of the traffic schedule between the requesting node and the switch and between the switch and the destination node (admission control). If the switch finds the schedule feasible (see next section), the RequestFrame is then forwarded to the destination node, after adding a network unique ID in the RT channel ID field. The destination node responds with a ResponseFrame (see Figure 7) to the switch telling whether the establishment is accepted or not. The switch will then, after taking notation of the response, forward the ResponseFrame to the source node. If the switch did not find the requested RT channel feasible to schedule, the RequestFrame is not forwarded to the destination node. Instead, a ResponseFrame is sent directly to the source node telling about the rejection.

The switch has two own MAC addresses, one for control traffic (e.g., RequestFrames) and one for real-time traffic over RT channels. The switch will in this way be able to easy (e.g., in hardware) filter out the different kinds of frames: (*i*) control frames, (*ii*) frames belonging to established real-time channels, and (*iii*) non-real-time frames. A non-real-time frame carries the final destination


Data field in Ethernet-frame containing details about the connection request

Dest. MAC addr. = switch addr.		Type: Connect packet	MAC source address	MAC dest. address	IP source address	IP dest. address	T_{period}	C	$T_{deadline}$	Connect. request ID	RT channel ID
48 bits		8 bits	48 bits	48 bits	32 bits	32 bits	32 bits	32 bits	32 bits	8 bits	16 bits

Figure 6: RequestFrame sent by source node trying to establish an RT channel.

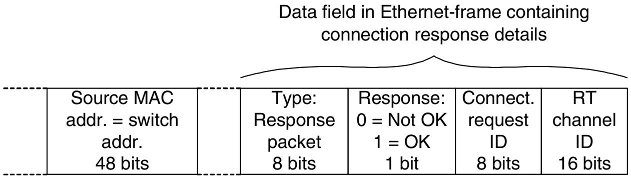

Data field in Ethernet-frame containing connection response details
Source MAC addr. = switch addr. 48 bits
Type: Response packet 8 bits
Response: 0 = Not OK 1 = OK 1 bit
Connect. request ID 8 bits
RT channel ID 16 bits

Figure 7: ResponseFrame.

MAC address in the Ethernet header already when leaving the source node. The end-nodes recognize control frames by reading the MAC source address that is set to the switch address.

3.2 Real-time traffic handling

The RT layer in an end-node prepares outgoing real-time IP datagrams by changing the IP header before letting the Ethernet layers sending it, in the data part of an Ethernet frame, to the switch (see Figure 8). The IP source address and the 16 most significant bits of the IP destination address, 48 bits together, are set to the absolute deadline of the frame. A 48 bit absolute deadline with a resolution of T_{frame} = 125 µs, gives a ‘‘life time”longer than one thousand years. The 16 least significant bits of the IP destination are set to the RT channel ID for the RT channel to which the frame belongs. The Type of Service (ToS) field is always set to value 255. Other values than 255 in the ToS field can be used for future services.

The switch exchanges the source and destination IP addresses and the MAC destination address of an incoming real-time frame with the correct ones (as stored in the switch when the RT channel was established) for delivery to the final destination. Also, the IP header checksum and the Ethernet CRC are recalculated before putting the frame in the correct deadline-sorted output queue. The checksums of non-real-time frames do not need to be recalculated.

4 Deadline scheduling

We assume that Node 1 wants to send real-time traffic to Node 2. The traffic is carried over RT channel i, where $T_{D1,i}$ and $T_{D2,i}$ are the deadlines for real-time frames from Node 1 to the switch and from the switch to Node 2, respectively. The relation between the period duration, $T_{period,i}$, of the RT channel and the deadlines is:

$$T_{period,i} = T_{D1,i} + T_{D2,i} \tag{4}$$

$$T_{D1,i} = T_{D2,i} = T_{period,i} / 2 \tag{5}$$

The scheduling of real-time frames in the switch (and for outgoing real-time frames in the end-nodes) is made according to earliest deadline first, i.e., all incoming real-time traffic is served in deadline order to guarantee the worst-case delay.

When the switch checks the feasibility of accepting a new connection (admission control), it uses an EDF theory modified to reflect the characteristics of the Ethernet network proposed in this paper. According to the basic EDF theory [6], the utilization of real-time traffic is defined as

$$U = \sum \frac{C_i}{T_{period,i}} \tag{6}$$

when the period is equal to the deadline (maximum delay) over the communication link (not true for this system as explained below). To be sure that all deadlines are met, the utilization of real-time traffic must not exceed a certain level, U_{max}:

$$U = \sum \frac{C_i}{T_{period,i}} < U_{\max} \tag{7}$$

where U_{max} in the theoretical case is 100 %, i.e., U_{max} = 1. A lower value of U_{max} is used instead of 100 % utilization when using the network proposed in this paper, with an off-line schedulability analysis or an on-line admission control.

The worst-case maximum utilization for a link between the switch and the destination node, U_{max2}, i.e., utilization that can be gained at full load, is reduced from 100 % to 90 % due to having every tenth possible frame being a synchronization frame. Also, because $T_{D1,i} = T_{D2,i} = T_{period,i} / 2$, the maximum utilization accountable for real-time traffic according to the EDF analysis is 45 % = 90 % / 2,

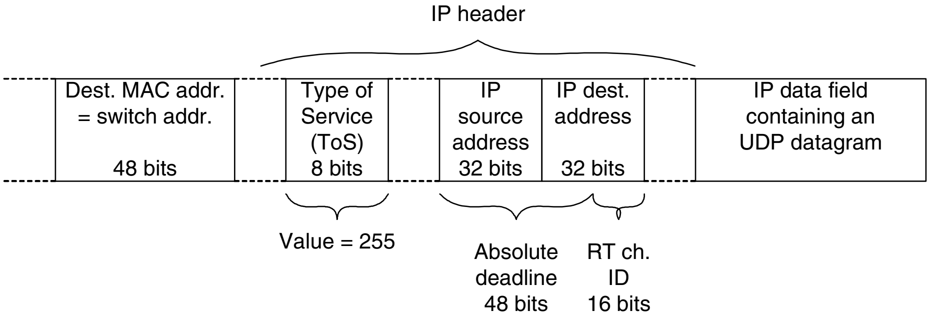
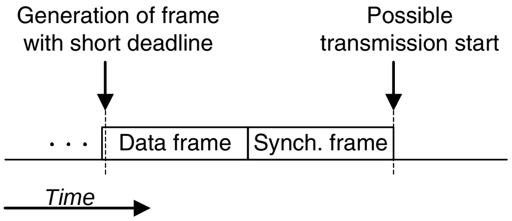

IP header
Dest. MAC addr. = switch addr. 48 bits
Type of Service (ToS) 8 bits
IP source address 32 bits
IP dest. address 32 bits
IP data field containing an UDP datagram
Value = 255
Absolute deadline 48 bits
RT ch. ID 16 bits

Figure 8: Data frame sent over an RT channel.



Generation of frame with short deadline
Possible transmission start
. . .
Data frame
Synch. frame
Time

Figure 9: Worst-case latency when waiting for the completion of one data frame followed by a synchronization frame.

i.e., we only have half of the period duration to get from the switch to the destination node. In summary we have:

$$\sum \frac{C_i}{T_{period,i}} < U_{\max 2} \tag{8}$$

where

$$U_{\max 2} = \frac{(T_{cycle} - T_{frame})}{2T_{cycle}} = 0.45 . \tag{9}$$

In the same way we can calculate the worst-case maximum utilization for a link between the source node and the switch, U_{max1}. However, we get a higher utilization compared to U_{max2} because there is no synchronization frames on the links in this direction. Thus we get:

$$U_{\max 1} = \frac{1}{2} = 0.5 . \tag{9}$$

In practice, a lower value of $U_{\max}$ can be used to always allow part of the bandwidth for non-real-time traffic.

In the worst-case situation, when all RT-channel start at the same time or all messages using its RT channel's full capacity allowance, the RT-channel with the longest deadline will be scheduled at last so it has the worst delay. However, the worst-case delay is, for all RT channels, characterized by:

$$T_{max\_delay,i} = T_{D1,i} + T_{D2,i} + T_{latency} \tag{10}$$

or

$$T_{max\_delay,i} = T_{period,i} + T_{latency} \tag{11}$$

where $T_{latency}$ is the worst-case latency experienced by a frame before it is transmitted even though it has the earliest deadline. The Worst-case latency to be added to the deadline is:

$$T_{latency} = 2T_{link\_prop\_delay} + T_{node\_access} + T_{switch\_access} \tag{12}$$

where $T_{link\_prop\_delay}$ is the maximum propagation delay over a link between an end-node and the switch, T_{node_access} is the worst-case latency for a frame with the earliest deadline to leave the source node, and T_{switch_access} is the worst-case latency for a frame with the earliest deadline to leave the switch. The source node latency is:

$$T_{node\_access} = Q\ T_{frame} \tag{13}$$

where Q is number of frames that can be stored on the NIC (Network Interface Card). In other words, we assume that we cannot interrupt the transmission of frames that have been stored on the NIC, even though they might have later deadlines than other frames.

The switch latency, before being able to forward a frame with the earliest deadline to the destination node, is

$$T_{switch\_access} = \text{MAX}(2T_{frame},\ Q\ T_{frame}). \tag{14}$$

The first term in the MAX expression is the maximum wait time due to the case when a frame is generated just after the transmission of a data frame has been initiated and the synchronization frame should be sent immediately after that (see Figure 9). The second term in the MAX expression tells the same thing as for the source node latency. The MAX operator is used because the first term is included in the second term. If Q is equal to 1 (which can be expected for a switch), we get $T_{switch\_access} = 2T_{frame}$.

Below we calculate an example of a system configuration. We use a 100 Mbit/s Ethernet switch with Ethernet frames that has the data field maximized (1518 bytes in IEEE 802.3 Ethernet), while the number of frames that can stored in the NIC is 2:

$$Q = 2. \tag{15}$$

The time for the maximum sized frame to be sent is:

$$\frac{8 \cdot 1518 Bytes}{100 Mbit/s} = 121 \mu s \tag{16}$$

if not assuming that T_{frame} = 125 μs. If the distance between two nodes is 100 m and Q = 1 for the switch, we have:

$$\begin{aligned} T_{link\_prop\_delay} &= 500 \text{ ns} \\ T_{switch\_access} &= 2T_{frame} = 242\ \mu\text{s} \\ T_{node\_access} &= 2\ T_{frame} = 242\ \mu\text{s} \\ T_{latency} &= 2 \cdot 500 \text{ ns} + 242\ \mu\text{s} + 242\ \mu\text{s} = 485\ \mu\text{s} \end{aligned} \tag{17}$$

In many industrial applications, the $T_{latency}$ that is added to the period in the calculation of the worst-case delay should typically not be of any problem.

5 Conclusions

In this paper, we have presented a switched Ethernet based network concept supporting real-time communication with guaranteed bit rate and worst-case delay for periodic traffic. The Ethernet switch operates at 100 Mbit/s over full-duplex links, and handles non-real-time traffic as well as real-time traffic. In the proposed solution there are no modifications in the Ethernet hardware on the network interface cards, which is important to allow the network to be connected to existing Ethernet networks. Real-time communication is handled in the nodes and the switch, by software added between the network layer and the link layer. Support for real-time communication is made by dynamically setting up real-time channels. Using Ethernet and the TCP/IP suite allows the network to be connected to the office network and to the Internet at the same time as it carries important real-time traffic in, e.g., a manufacturing industry.

References

[1] C. Venkatramani and T. S. Chiueh, "Supporting real-time traffic on Ethernet," *Proc. 15th IEEE Real-Time Systems Symposium (RTSS'94)*, pp. 282-286, 1994.

[2] D. W. Pritty, J. R. Malone, D. N. Smeed, S. K. Banerjee, and N. L. Lawrie, "A real-time upgrade for Ethernet based factory networking", *Proc. IEEE IECON'95*, vol. 2 , 1995.

[3] S.-K. Kweon, K. G. Shin, and G. Workman, "Achieving real-time communication over Ethernet with adaptive traffic smoothing," *Proc. 6th IEEE Real-Time Technology and Applications Symposium (RTAS'2000)*, Washington, D.C., USA, 31 May - 2 June 2000, pp. 90-100.

[4] S. Varadarajan and T. Chiueh, "EtheReal: A host-transparent real-time Fast Ethernet switch," *Proc. ICNP*, Oct. 1998.

[5] H. Zhang, "Service disciplines for guaranteed performance service in packet switching network," *Proc. of the IEEE*, vol. 83, no. 10, Oct. 1995.

[6] C. L. Liu and J. W. Layland, "Scheduling algorithms for multipprogramming in hard real-time traffic environment", *Journal of the Association for Computing Machinery*, vol. 20, no. 1, Jan. 1973.

[7] A. Larsson and R. Olsson, "Implementation outline for a real-time Ethernet switch," *Master thesis, School of IDE, Hamstad University, Sweden*, Jan. 2002.