# Industrial Hard Real-Time Communication Protocol over Switched Ethernet

ALIMUJIANG YIMING and TOSHIO EISAKA
Dept of Computer Sciences
Kitami Institute of Technology
165, Koen-cho, Kitami, Hokkaido 090-8507
JAPAN
alm_ym2000@yahoo.com, eisaka@cs.kitami-it.ac.jp

*Abstract:* - This paper presents a protocol to support hard real-time traffic of end-to-end communication over non real-time LAN technology. The protocol does not need any modifications in the Ethernet hardware and coexists with TCP/IP suites, and then the LAN with the protocol can be connected to any existing Ethernet networks. We have performed some experiments to evaluate the protocol. Compared to some conventional hard real-time network protocols, the proposed one has better real-time performances and meets the requirements of reliability for hard real-time system applications.

*Key-Words:* - switched Ethernet, hard real-time, traffic scheduling, admission control, EDF algorithm

## 1 Introduction

With the increasingly demand for real-time industrial control systems, the ability of computer networks to handle deadline guaranteed "hard" real-time communication is becoming more and more important [1-2]. High bandwidth and determinism are the critical necessary conditions for hard real-time applications [3]. Unlike traditional, bus-based CSMA/CD Ethernet, the switched Ethernet is a star-based topology which can provide collision domain to each of the ports of a switch. Therefore, end-node cooperation is needed only for bandwidth control, not any more to avoid collisions.

Several researches have been done to treat hard real-time communication [4-7], however, these protocols either bring about complex network structures or need Ethernet hardware modifications to implement.

Recently, new schemes have been proposed based on admission control depending upon quality of service (QoS) and choice of the packet service discipline [8-10]. The common concept of the schemes is establishment of a *Real-time Channel*: a simplex, virtual connection between source nodes and destination nodes, with a priori guarantees for communication performance in switching networks. In order to provide guaranteed services, resources need to be reserved for every accepted connection [11]. However, most of the applications are no guarantee for strict deadline-sorted hard real-time communication.

On the other hand, RSVP (ReSerVation Protocol) [12] allows guaranteed service for multicast applications. Therefore, it is appropriate for multimedia communication such as online video conferencing. However, because RSVP is based on IP, there is no guarantee of deadline in application service lifetime, and have large runtime overhead, there are few applications in industrial real-time control systems.

Based on the knowledge of above-mentioned QoS architectures and protocols, in our research work, we developed and analyzed a protocol to schedule and control the hard real-time traffic based on the industrial hard real-time communication demands, without changing the underlying protocols, while still supporting existing upper protocols for the best-effort traffic. In our work, a key strategy to realize hard real-time communication is to manage the traffic to bypass the TCP/IP stacks. This makes considerably reduce the dwell time in the nodes, and increase the achievable data frame rate by evasion of the non-deterministic behavior inherent in the TCP and IP stacks. This is the main point of our work.

The network is set up with nodes and switches, and the hard real-time communication support is handled by software added between the network layer and the data link layer, in the OSI reference model.

## 2 Hard real-time communication supports

In this section, design and implementation of a switched Ethernet protocol for hard real-time communication are discussed. In subsection 2.1, network architecture is introduced. Subsection 2.2 describes the real-time channel establishment by applying admission control. Subsection 2.3 illustrates the management of hard real-time traffic and best-effort traffic. Lastly, in subsection 2.4, we elaborate feasibility analysis for real-time channel establishment and focus on scheduling of hard real-time data frames.

### 2.1 Network architecture

We applied a full-duplex switched Ethernet which is connected to existing Internet. A switched Ethernet provides some key benefits over traditional Ethernet, such as full duplex and flow control. Therefore, switches enable flexible network configuration of multiple and simultaneous links between various ports.

In our work, a key strategy to realize hard real-time communication is *bypassing* of TCP/IP suites. In order to manage this bypass, both the switch and the end-nodes have software --- real-time layer (RT layer) added between the Ethernet protocols and the TCP/IP suite in the OSI reference model. Fig. 1 depicts the network architecture with real-time channels.

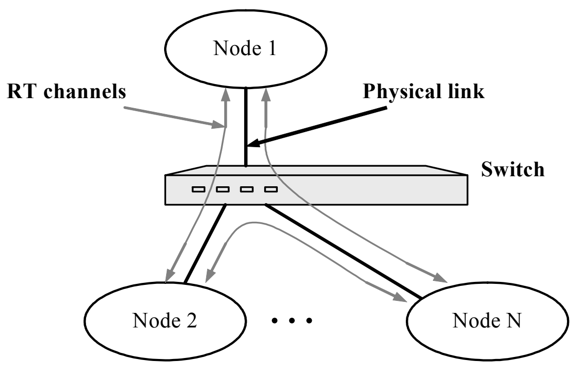


Fig. 1 Network architecture with RT channels

All nodes are connected to the switch and nodes can communicate mutually on the logical real-time channels (RT channel), it is a virtual connection between two nodes of the system respectively. A node can either be real-time node or non real-time node depending on which level QoS is required. Non-real-time node (without RT layer added) can coexist in the network without disturbing the real-time traffic. MAC function, frame buffering and the concentrated transmission arbitration is included in the switch. Therefore, switch has the overall responsibility both for set-up of RT channels and for online control of packets passing through the switch. The RT layer do-nothing to non real-time frames and makes them go through the ordinary circuit with TCP/IP suites.

### 2.2 RT Channel Establishment

Before the real-time traffic is transmitted, the RT channel should be established. As new real-time requests (channel establishment) are made, they go through an admission control module that determines if there is sufficient network bandwidth available to satisfy the request of the node. Admission control is the problem of deciding which requests to accept and which to reject based upon the supported QoS, with the goal of maximizing the total profit accrued by the accepted requests. In other words, admission control is the problem of finding a feasible solution with maximum profit.

If the network establishes a transmission request, it first decides on a path from the sending node to the receiving node of that transmission, through which the transmission is being routed. Then it allocates the requested amount of bandwidth and/or buffer space on all links along that path during the time period in which the transmission is active. The allocated resources are released when the connection is completed.

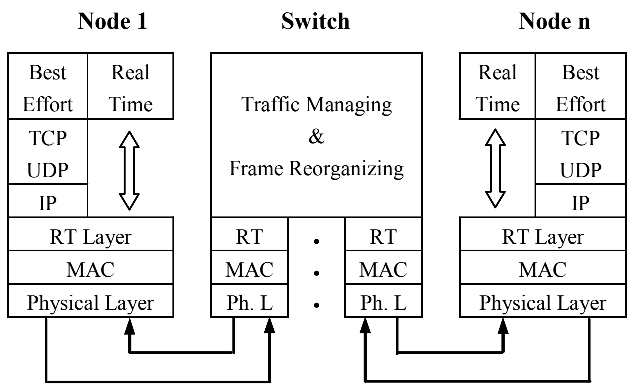


Fig. 2 Real-Time channel establishment

Fig. 2 describes the establishment of an RT channel. When a node wants to send hard real-time frames, it directly accesses the RT layer. The RT layer then sends "RT channel establishment request" to the RT traffic management in the switch. The switch then evaluates the feasibility of traffic schedule of that transmission, by applying the admission control. If the schedule is feasible, the switch responses with the network schedule parameters to the sending node. Otherwise, the switch sends out a set of recommended control parameters to the sending node. These control parameters are suggested based on the status of switch queue and the active queue control law.

## 2.3 Traffic Management

After RT channel is established, only real-time data traffic from the end-node bypasses the TCP/IP stacks and thus considerably reduces the dwell time in the nodes, and increases the achievable data frame rate by evasion of the non-deterministic behavior inherent in the TCP and IP stacks. Here dwell time of a node refers to one of the substantial influence factors for the real-time performance.

The experimental results prove that by using proposed protocol to bypass the TCP/IP stacks can reduce 32% of time comparing with UDP/IP protocol, and more than 50% of time comparing with TCP/IP protocol, even if the proposed protocol needs RT channel establishment and traffic scheduling.

Besides hard real-time traffic, our Ethernet network protocol allows for best-effort traffic without affect the transmission of hard real-time packets. Namely, best-effort traffic (non real-time or soft real-time traffic) come from best-effort protocols (HTTP, SMTP, FTP, etc.) uses the services of the TCP/IP protocol suites and put in an FCFS-sorted (First Come First Serve) queue in the RT layer. In order to achieve this, best-effort traffic is allowed when no hard real-time packets want to transmit.

When a hard real-time packet becomes ready to transmit again, the RT channel management immediately interrupts the best-effort traffic and goes to the corresponding node, so that the hard real-time traffic may start. The last node visited for best-effort traffic should be remembered, so the next round of best-effort traffic packet can start off at that node.

Because there are two different output queues for each port on the switch, "frame recognizing" is necessary. On that account, the switch has two MAC addresses: one is for control traffic (e.g., RT channel request frames); and the other is for hard real-time traffic over RT channels. And then, the switch will be able to recognize the different kinds of frames: control frames, real-time data frames and best-effort data frames that come from TCP/IP stacks.

## 2.4 Scheduling of Hard Real-Time Frames

In our star-like network architecture, every end-node is connected with a private virtual link to a switch, so that there is a private traffic link for each direction. But congestion may occur when one node is suddenly receiving lots of packets from the other nodes. Current switches do not provide any guarantees as to which packets will be sent first. We solved this by providing a switch with bandwidth reservation capabilities inside the switch, and used Earliest Deadline First (EDF) scheduling [13] to make decisions as to which packets are forwarded first. This provides guarantees for both bit rates and strict delivery deadlines. The advantage of using EDF in hard real-time traffic was shown in [14], as it is the optimal and dynamic-priority scheduling algorithm under the resource sufficient situation.

First, we check the feasibility of the real-time traffic according to calculate the total utilization of all frames. RT channel of the *i*-th task is characterized by $\{T_{pd,i}, C_i, T_{d,i}\}$; where $T_{pd,i}$ is period of the data, $C_i$ is time required to complete the execution of the task per the period, $T_{d,i}$ is the relative deadline used for the EDF scheduling.

The task period $T_{pd,i}$ can be described as:

$$T_{pd,i} = T_{n1,i} + T_{n2,i} + T_{ct}, \quad (1)$$

where $T_{n1,i}$ and $T_{n2,i}$ are the deadlines of each real-time frame for upload and download, respectively; and $T_{ct}$ is the delay introduced by the switch. For the input traffic, scheduling analysis can give the worst-case buffering delay, providing thus the hard real-time guarantee.

According to EDF theory, the total utilization of all frames is then calculated as:

$$U = \sum_i \frac{C_i}{T_{pd,i}}. \quad (2)$$

Suppose $T_{pd,i} \le T_{d,i}$ for simplicity, it is well known that EDF scheduling is feasible if and only if $U \le 1$.

If the test for task *i* succeed and real-time channel is established, hard real-time data frame bypasses the TCP/IP stacks and put in a deadline-sorted queue scheduled by RT layer in the switch and end-nodes according to the EDF theory.

Another important role of the RT layer is to handle

"RT data frame processing" (see Fig. 3). The RT layer makes the switch to check the value of an incoming real-time frame, recalculate the Ethernet cyclic redundancy check (CRC) of the real-time data frame, and destination address test of the frame, before putting it to the correct deadline-sorted output queue. This will also be helpful to increase the reliability of the real-time data frames.

The checksums of non real-time frames do not need to be recalculated.

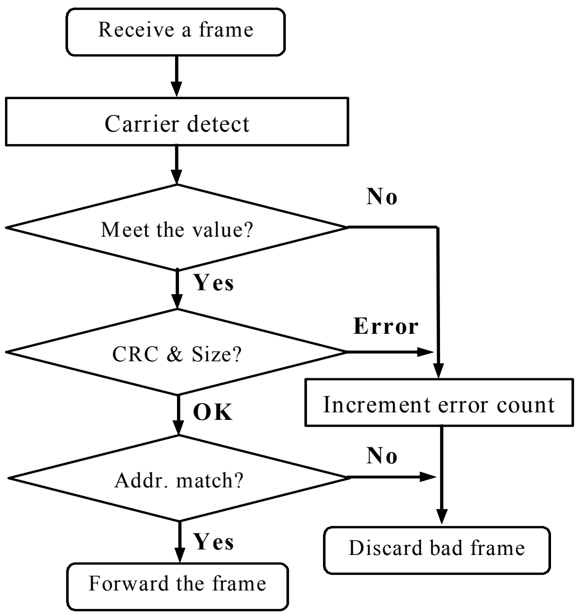


Fig. 3 Real-time data frame processing.

To make a brief summary above, the process of hard real-time traffic and best-effort traffic transmission is shown in Fig. 4, and is outlined as follows:

1. When the switch received a packet from an end-node, it recognizes which application the packet is come from (real-time or best-effort).

2. If the packet come from the real-time application, then the RT layer interrupt transmitting best-effort traffic immediately so that the hard real-time traffic may start. And the information for last transmitted queue of the best-effort traffic should be stored so that the next round of best-effort traffic can start off at that queue.

3-A. The switch will be able to recognize the different kinds of frames: control frames (e.g., RT channel request frames) and real-time data frames. If the received packet is the control frame, then the RT layer must go through an admission control module that determines if there is sufficient network resource (bandwidth and guaranteed time limit) available to satisfy the request of the node.

4-A. If a request is admitted, the switch answers to the requesting nodes with a set of network schedule parameters, and make RT channel virtual connection. And then allocates the requested amount of bandwidth and/or buffer space on this connection link.

4'-A. Otherwise, the switch sends out a set of recommended control parameters to the sending node.

5-B. After RT channel is established, hard real-time data is delivered through the circuit and bypassing the TCP/IP stacks by reading MAC addresses in response parameter.

6-B. The RT layer makes the switch recalculate the Ethernet cyclic redundancy check (CRC) of an incoming hard real-time frame before putting it to the correct deadline-sorted output queue. This will also be useful to increase the reliability of the hard real-time data frames.

7-B. Real-time data passed the above check is then put in a deadline-sorted queue scheduled by RT layer in the switch and end-nodes according to the EDF theory, and then,

8-B. Forward the deadline-sorted data to the destination node. The allocated resources are released when the connection is completed.

3-C. On the other hand, by carrying the final destination MAC address in the Ethernet header when leaving from the source node, non or soft real-time data is delivered through the circuit including the TCP/IP stacks in an FCFS-sorted queue, and transmit the traffics at the idle time of the schedule.

4-C. If a best-effort sender needs to send a large amount of data (for example, a long packet), it tries to make an additional cycling time reservation and transmit its data immediately after reservation. If the time is over and the long packet did not finished yet, it tries to make a reservation again.

The protocol designed with above procedure meets with the deadline both for the periodic and aperiodic data frames, and also realizes faster execution of the real-time tasks.

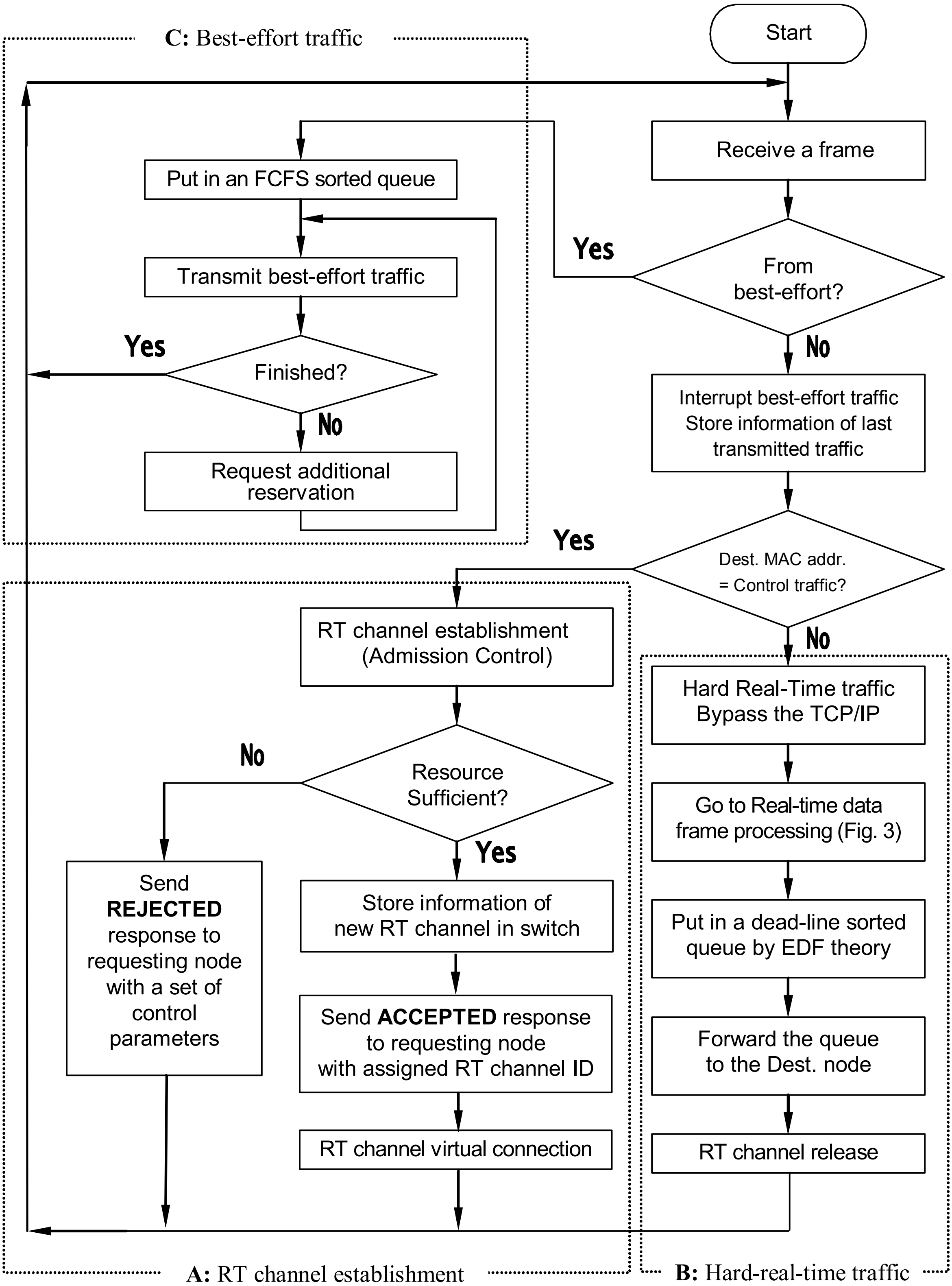


Fig.4 Processes of traffic transmission

## 3 Performance Evaluations

To evaluate our proposed protocol, we made a LAN with a full-duplex switched Ethernet and end-nodes, by using desktop computer with AMD-K7(tm) Processor 700MHz and several embedded Ethernet development boards which is produced by YDK Technologies Inc. that provides a hardware platform based on Altera® ACEX™ devices (see Fig. 5). We use a 5-ports Ethernet switch with full-duplex links at 100Mbps. Both fixed and variable sized data packets were used. The size of data packet is from 64 bytes to 1538 bytes. There are no modifications in the Ethernet hardware on the NIC.

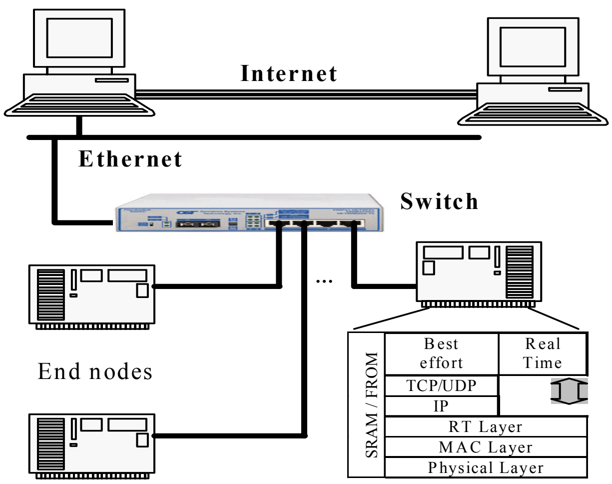


Fig. 5 LAN with full-duplex switched Ethernet

Below we discuss about the transmission latency of the real-time frame in worst-case situation. When all RT-channel starts simultaneously, or all the messages that use all the capability allowances of the RT channel, RT channel equipped with the longest deadline will be scheduled at last so that it may have the worst-case latency. Here for all RT channels, the maximum latency is characterized by:

$$T_{m\_lat} = \max_i \{T_{n1,i} + T_{n2,i} + T_{ct}\}, \quad (3)$$

where $T_{n1,i}$ is the latency from source nodes to switch, $T_{n2,i}$ is the latency from switch to destination nodes, and $T_{ct}$ is the latency of the switch.

Besides utilization and worst-case latency, another important performance is a runtime overhead: $R_i$ defined as:

$$R_i = \frac{T_{pd,i} - L_i \times 8 / B}{T_{pd,i}}, \quad (4)$$

where $L_i$ is the length of data in a request frame, $L_i \times 8$ is the number of bits in the frame; $T_{pd,i}$ represents the period duration from the startup to the end of the frame, and $B$ represents the Ethernet bandwidth.

Utilization, Data frame transmission latency and the frame runtime overhead can be obtained by implementing the proposed protocol to the LAN.

Fig. 6 illustrates the utilization on real-time data frame. From the figure we can learn that the trend of utilization is increasing while the traffic increases, until arriving at the peak value that is more than 90%. Sudden decreases happen on the curve sometimes, which is caused by some short frames having pad field whose utilization are lower than longer frames. The utilization curve is always smooth, because we assume the sufficient resources (bandwidth and time specification) have been obtained in our work, when the RT channel is established. Under this circumstance, there should be no overload exists in the real-time channel. The result shows that the deadlines have been met for all data because utilization of all data frames is less than 100% using EDF scheduling. Dynamic priority scheduling with the EDF algorithm has a distinct advantage over fixed priority scheduling: the schedulable bound for EDF is 100% for all task sets. This means that we can fully utilize the computing power of the CPU. Embedded systems are in general fully loaded, as they attempt to get as close to 100% utilization as possible while still maintaining the necessary predictability.

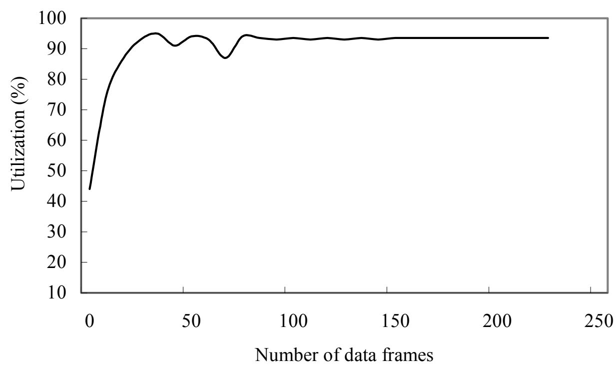


Fig. 6 Utilization on real-time data frame

To the best of our knowledge, except a few implementations of hard real-time communication protocols on Ethernet, most of the protocols are generally soft real-time, which means that there are few protocols provides guarantees for both bit rate and strict delivery deadlines, so that it is difficult to compare them with the proposed hard real-time protocol. Therefore, we made performance comparison of the proposed protocol only with the hard real-time communication protocols: MIL-STD-1553B protocol and RTCC protocol. Fig. 7 shows the comparison of the data frame transmission latency of these three kinds of hard real-time communication protocols. Even for the Ethernet frames that have the data field maximized (1538 bytes in IEEE 802.3 standard), the latency of the proposed protocol is about 620 microseconds. This latency is quite short in a LAN with a full-duplex switched Ethernet at 100 Mbps, and meets the demands of hard real-time communication for industrial distributed control systems.

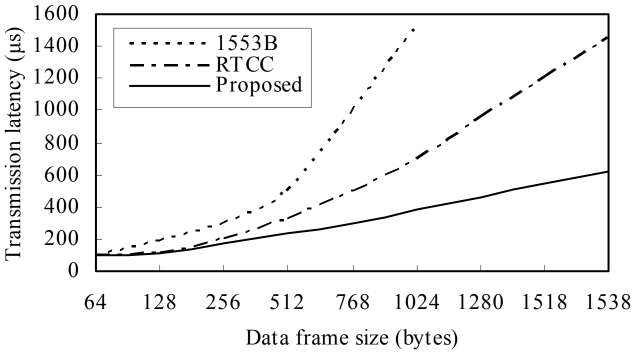


Fig. 7 Transmission latency of data frame

Runtime overhead of data frames are demonstrated in Fig. 8. The figure shows that the runtime overhead of the proposed protocol is higher than the other hard real-time supported protocols at the small-sized data frame. However, as the data frame size become larger (from about 900 bytes), the proposed protocol has better runtime overhead than the other protocols. In both experiments, only MIL-STD-1553B protocol used 1Mbps Ethernet because it is the nominal speed of the protocol.

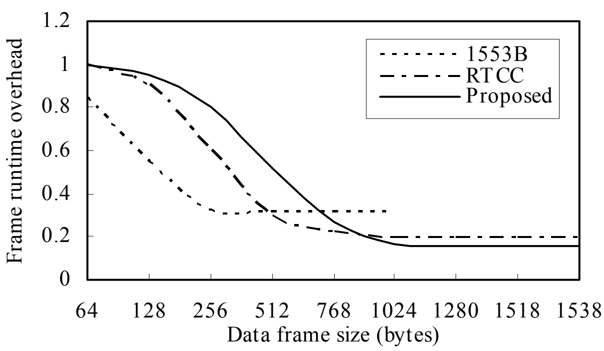


Fig. 8 Runtime overhead of data frame

Furthermore, in order to evaluate the time effectiveness of hard real-time bypassing the TCP/IP stack, we made experiments with ordinary UDP/IP and TCP/IP protocols comparing with the proposed protocol (see Fig. 9). The result shows that by using the proposed protocol to bypass the TCP/IP stacks can reduce 32% of the time comparing with UDP/IP protocol, and more than 50% of the time comparing with TCP/IP protocol, even if the proposed protocol needs for RT channel establishment and EDF scheduling.

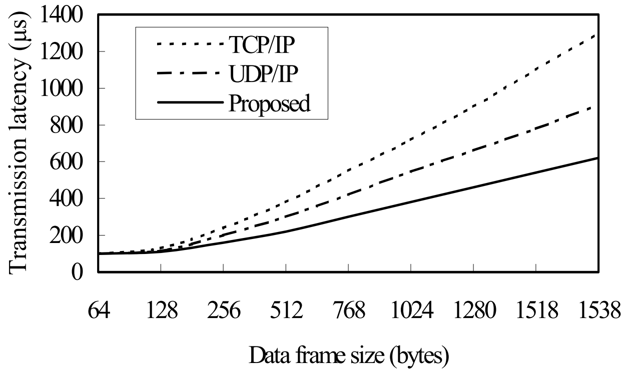


Fig. 9 Transmission latency comparison

# 4 Conclusions

In this paper, we have presented a simple and efficient switched Ethernet communication protocol for industrial hard real-time applications. The proposed protocol establishes a virtual link between source nodes and destination nodes by applying admission control based upon the requested QoS. Hard real-time traffic from the end-node bypasses the TCP/IP stacks and thus considerably speeds up the real-time communication. Real-time traffic scheduling is performed according to dynamic priority EDF algorithm. Therefore, it is flexible and efficient.

In the proposed work, there are no modifications in the Ethernet hardware on the NIC. Therefore, we successfully shared the Ethernet LAN with existing Internet networks, by applying the proposed Ethernet protocol.

Through the comparison with some conventional hard real-time network protocols, we have shown that the proposed protocol has better real-time performances, and well meets the requirements of reliability for hard real-time systems. It can be adopted in industrial hard real-time applications with a limited number of end nodes such as factory-floor embedded system communications, distributed industrial control system communications, parallel signal processing and robotics.

One of our further research subjects is to consider the way of implementing and experiment by using the proposed protocol to the multi-layer, many switches real-time communication systems.

*References:*


[1] J. Stankovic and K. Ramamritham, *Hard Real-Time Systems*, IEEE Computer Society Press, 1988.

[2] C. Krishna and K.G. Shin, *Real-Time Systems*, McGraw-Hill International edition, 1997.

[3] M. Joseph, *Real-Time Systems*, Prentice Hall, 1996

[4] http://www.condoreng.com/support/downloads/tutorials/MIL-STD-1553Tutorial.PDF

[5] Z. P. Wang, G. Z. Xiong, J. Luo, M. Z. Lai and W. Zhou, A hard real-time communication control protocol based on the Ethernet, *Proc. of the PART 2000*, 2000, pp.161-170.

[6] J. Loeser and H. Haertig, Low-latency hard real-time communication over switched Ethernet, *Proc. of the ECRTS'04*, 2004, pp. 13-22.

[7] S. Ouni and F. Kamoun, Hard and soft real time message scheduling on Ethernet networks, *Proc. of the 2nd IEEE International Conference on Systems, Man and Cybernetics of the twenty-first century*, No.6, 2002.

[8] D. Ferrari and D. Verma, A scheme for real-time channel establishment in wade-area networks, *IEEE Journal of Selected Areas in Communications*, Vol.8, No.3, 1990, pp.368-379.

[9] L. Zhang, Designing a new architecture for packet switching communication networks, *IEEE Communications Magazine*, Vol. 25, No. 9, 1987, pp. 5-12.

[10] G. Agrawal, B. Chen, W. Zhao, and S. Davari, Guaranteeing Synchronous Message Deadlines with Time Token Medium Access Control Protocol, *IEEE Transactions on Computers*, Vol. 43, No. 3, 1994, pp 327-339.

[11] A. Leon-Garcia and I. Widjaja, *Communication Networks - Fundamental Concepts and Key Architectures*, McGraw-Hill Osborne, 2001.

[12] M. Schwartz & T. E. Stern, *Routing Protocols, in Computer Network Architectures and Protocols*, (Second Ed.), Ed. C. A. Sunshine Press, 1989.

[13] C. L. Liu and J. W. Layland, Scheduling algorithms for multi-programming in hard real-time traffic environment, *Journal of the Association for Computing Machinery*, Vol. 20, No. 1, 1973, pp.46-61.

[14] Phillip A. Laptane, *Real-Time System Design and Analysis*, IEEE Press, third edition, 2004.