

# Support Industrial Hard Real-Time Traffic with Switched Ethernet

Alimujiang Yiming<sup>1</sup>, and Toshio Eisaka<sup>2</sup>

<sup>1</sup> Kitami Institute of Technology, Kitami, Hokkaido 090-8507, Japan  
alm\_ym@mer.cs.kitami-it.ac.jp

<sup>2</sup> Kitami Institute of Technology, Kitami, Hokkaido 090-8507, Japan  
eisaka@cs.kitami-it.ac.jp

**Abstract:** This paper presents a simple and efficient switched Ethernet communication protocol for industrial hard real-time LAN applications. The network is founded with end nodes and a switch, and hard real-time communication is handled by software added between the Ethernet protocols and the TCP/IP suites. We established a virtual link of the source and destination node by applying admission control based upon the requested QoS, and manages hard real-time traffic to bypass the TCP/IP stacks. This bypassing considerably reduces the dwell time in the nodes, and increases the achievable data frame rate. After the bypassing, hard real-time traffic schedule is executed according to dynamic-priority EDF algorithm.

The protocol does not need any modifications in the Ethernet hardware and coexists with TCP/IP suites, therefore, the LAN with the protocol can be connected to any existing Ethernet networks and support both real-time traffic and best-effort traffic. 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.

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

## 1. Introduction

Networks for industrial control system communication applications are usually some kind of fieldbus. The common characteristics for these networks were high reliability, low data throughput, and a high price tag. Over the years Ethernet transmission rate have increased, and become the most widely used local area networking (LAN) technology in the world today. 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], [3]. High bandwidth and strict deadline guarantee are the critical necessary conditions for hard real-time applications [4]. Unfortunately, traditional Ethernet uses a non-deterministic arbitration mechanism (CSMA/CD) [5] that makes it unsuitable for real-time communications.

However, the development of new equipment for network interconnection, i.e. Ethernet switches, has made a different approach possible. Unlike traditional,

bus-based CSMA/CD Ethernet, the switched Ethernet is a star-based topology which can avoid collision since Data terminal equipment (DTE) connected to a switch communicating in full duplex does not have to use the CSMA/CD access control. Therefore, end node cooperation is needed only for bandwidth control. In addition to this, the full-duplex operation theoretically doubles the bandwidth of the network, and the Ethernet transmission rate and communication reliability have increased over the years. This together with serious attempts to adapt Ethernet hardware to industrial environments, make it an interesting alternative for real-time communication.

Several researches have been done to treat hard real-time communication. MIL-STD-1533 [6] is an early development of the industrial protocol. It is reliable standard interface for token ring LAN. However, bandwidth is one of the main limiting factors for MIL-STD-1533. Another protocol which enables hard real-time communication on Ethernet, is called RTCC (Real-Time Communication Control) [7]. RTCC sits on top of Ethernet. One node is designated bus controller; all others are called remote terminals. A remote terminal has to receive permission from the controller before it may send any message. This centralized approach has the disadvantage that failure in the master renders the entire network useless, unless some sort of recovery protocol is implemented that enables the slaves to choose a new master among them. Other researches using switched Ethernet for hard real-time communication is studied in [8], [9], 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. 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 [10], [11], [12].

To support the QoS demands of applications, both the ATM and the IP community have defined service classes that provided per-flow guarantees to applications [13], [14]. In order to provide guaranteed services, resources need to be reserved for every accepted connection. ATM does this at the data-link layer. For every call it reserves a virtual channel over all links on the route/switch from the source to the destination. At the network layer, resource reservation can be done using Diffserv [15] or Multi Protocol Label Switching (MPLS) [16]. Applications at this level are classified as best-effort, rate sensitive or delay sensitive, therefore, there are no guarantee for strict deadline-sorted real-time communication.

RSVP [17] can be used to do reservation at the IP layer. In RSVP, the admission control and resource allocation policies are based on different levels of QoS specifications --- deterministic or guaranteed, statistical and best-effort. The main feature of RSVP is the maintenance of *soft-state* in the intermediate switches, which must be periodically refreshed. This approach allows RSVP to adept to network load and outages. RSVP is designed to deliver resource reservation requests to related switches. Therefore, RSVP is more appropriately a *state establishment protocol*. RSVP makes resource reservations for both unicast and multicast applications, therefore, it is more appropriate for multimedia communication in a wide area network such as videoconferencing. Because RSVP is based on IP, there is no guarantee of deadline in application service lifetime, and have large runtime overhead, therefore, there are few applications in industrial control systems.

Based on the knowledge of above-mentioned QoS architectures and protocols, in our work, we studied the industrial and embedded application demands on hard real-time communication services, including survey of real-time communication with focus on LAN-technology and switched communication. Then we developed and analyzed a protocol to schedule and control the hard real-time traffic based on the industrial and embedded real-time demands without changing the underlying protocols, while still supporting existing upper protocols for soft real-time and/or non-real-time traffic. We also establish a way (virtual link) between source nodes and destination nodes by applying admission control based upon the requested QoS. However, in our work, a key strategy to realize hard real-time communication is, the proposed protocol manages hard real-time 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. It can be adopted in industrial hard real-time applications fields such as distributed industrial systems, embedded systems and robotics.

One of the goals in our work is to use cheap existing hardware and software whenever possible, to implement a simple and efficient hard real-time communication network and protocol for the industrial distributed control systems and embedded control systems, while still supporting existing upper protocols for soft real-time and/or non-real-time traffic. This requires that the network should be based on existing hardware. Customized hardware would drive up the price of devices considerably and take up a lot of time to design and implement.

An Ethernet LAN using a switch and several end nodes was constructed, and several experiments have been performed to evaluate the proposed protocol. Comparing with the conventional hard real-time communication protocols, the proposed Ethernet protocol has better real-time performances and meets the requirements of reliability for hard real-time systems.

The rest of the paper is organized as follows. In section 2, outline of the network architecture is introduced. Section 3 describes the RT channel establishment by applying the admission control. Section 4 illustrates the management of hard real-time traffic and best-effort traffic. We elaborate feasibility analysis for RT channel establishment and focus on scheduling of real-time frames in section 5. In section 6, performance evaluation of the proposed protocol is presented. The paper is concluded in section 7.

## **2. Network architecture**

We applied a full-duplex switched Ethernet LAN 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. 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 (see Fig. 1). 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. Namely, the RT layer is compatible with existing TCP/IP and handles both real-time and non real-time traffic depending on QoS.

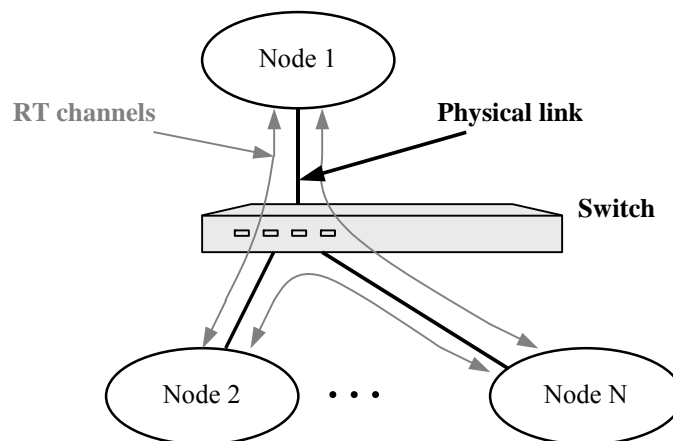


Fig. 1. Network architecture with RT channels

### 3. RT channel establishment

Before the real-time traffic is transmitted, the RT channel should be established. The establishment of RT channel is including request and recognition communication after the source nodes, destination nodes and switch have agreement with channel establishment. As new real-time requests (channel establishment) are made, they must 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.

As for the establishment of an RT channel, each connection request is specified by two distinct nodes: the sending node and the receiving node in the network that want to communicate. Each transmission request has a certain bandwidth requirement and some time specification given by its starting time and its duration. 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 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 a path from the sending node to the receiving node of that transmission, by applying the admission control. If the schedule is feasible, the switch responds 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.

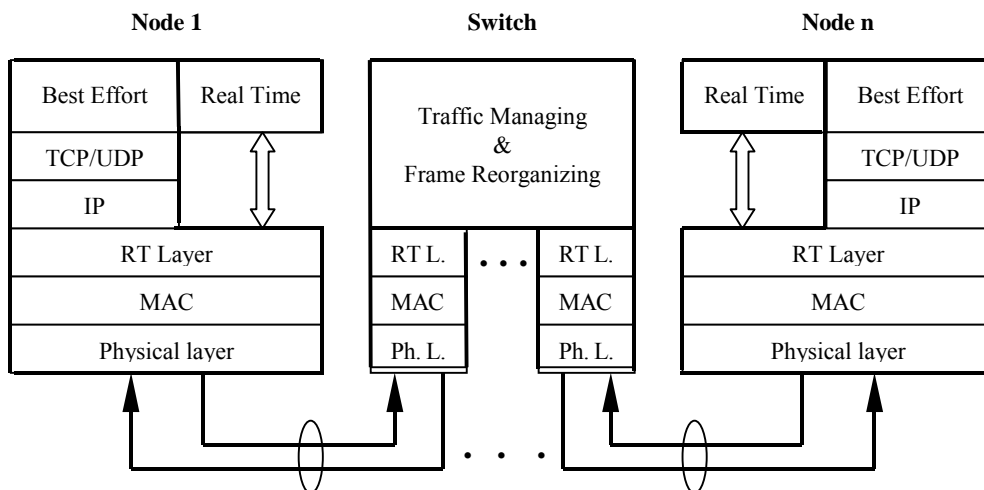


Fig. 2. Real-Time channel establishment

#### 4. Traffic management

After RT channel is established, only real-time data traffic from the end node bypasses the TCP/IP stacks. An RT channel should cross two physical links according to the RT layer: one from the source node to the switch, and the other from the switch to the destination node (uploads and downloads, respectively). The RT channel is required to provide real-time guarantees for both the upload and the download.

Besides hard real-time traffic, our Ethernet network protocol should allow for best-effort traffic which does not 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. If a best-effort sender needs higher bandwidth to send a large amount of data (for example, a long packet), it tries to make an additional 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.

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. According to the RT layer, 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. The end nodes recognize control frames by reading MAC source address that is set to the switch address. Non real-time frame carries the final destination MAC address in the Ethernet header when it leaves from the source node.

## 5. 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 a lot 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 we used Earliest Deadline First (EDF) scheduling [18] to make decisions as to which packets are forwarded first. This provides guarantees for both bit rates and strict delivery deadlines.

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 end-to-end 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. In a fully switched Ethernet there is only one equipment (end node) per each switch port. In case that wire-speed full-duplex switches are used, the end-to-end delay can be

minimized by decreasing the message buffering. The total delay introduced by a switch including:

- . The switching latency (including traffic classification and switch fabric set-up time),
- . The frame forwarding latency which depends on the forwarding mode,
- . The buffering delay when the frame is queued.

The switching latency is a fixed value which depends on the switch performance; the frame forwarding delay can be obtained knowing in which mode the switch is running; buffering delay exists in a switch whatever the switch is with full wire-speed or not. In fact message buffering occurs whenever the output port cannot forward all input messages on time. However, 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  $T_{pd,i} \leq T_{d,i}$  for simplicity, it is well known that EDF scheduling is feasible if and only if  $U \leq 1$ .

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.

The processes of hard real-time traffic and best-effort traffic transmission is shown in Fig. 3, and is summarized 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.

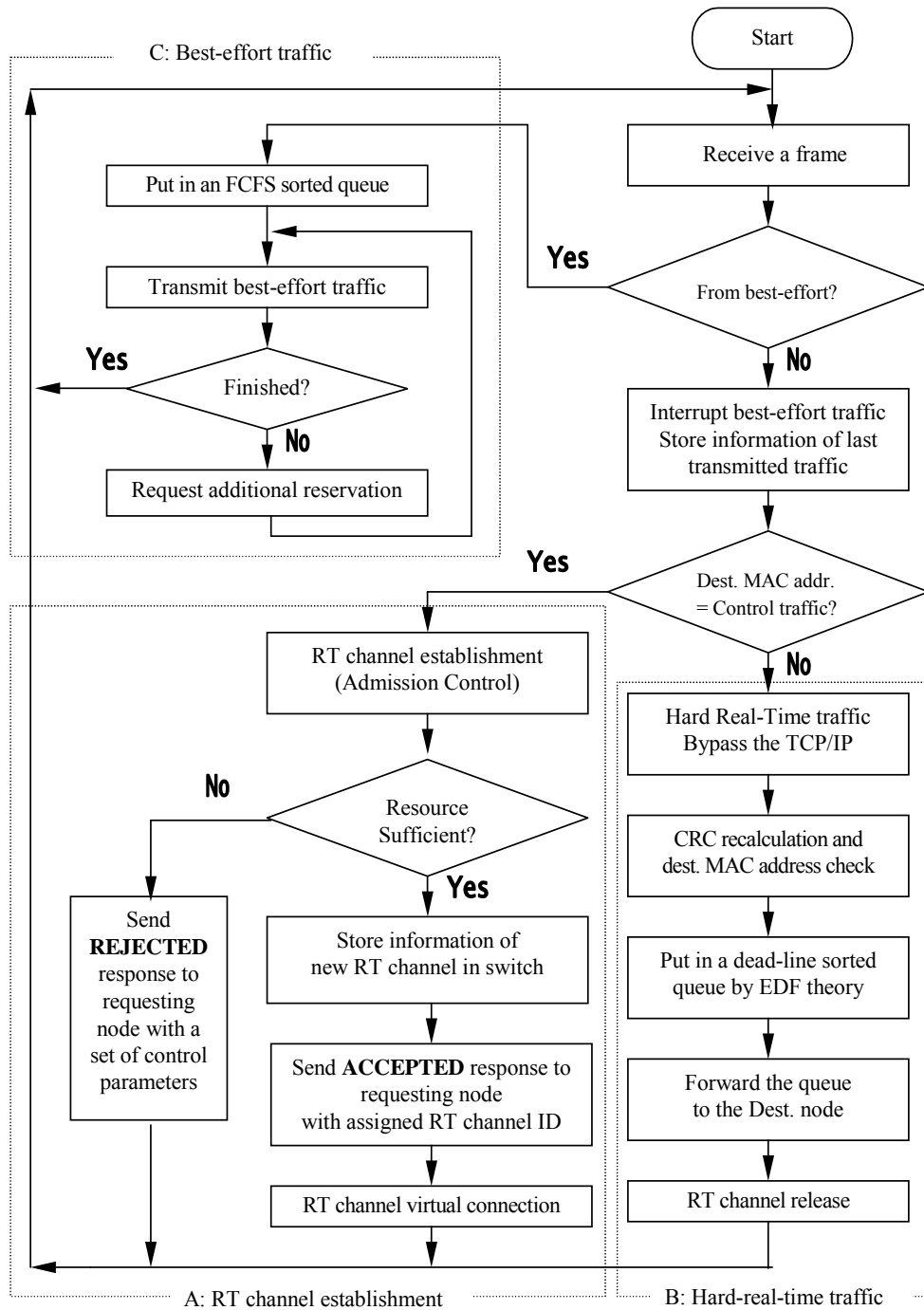


Fig.3. Processes of traffic transmission



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.

## 6. Performance evaluation

In order to evaluate our work, 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. 4).

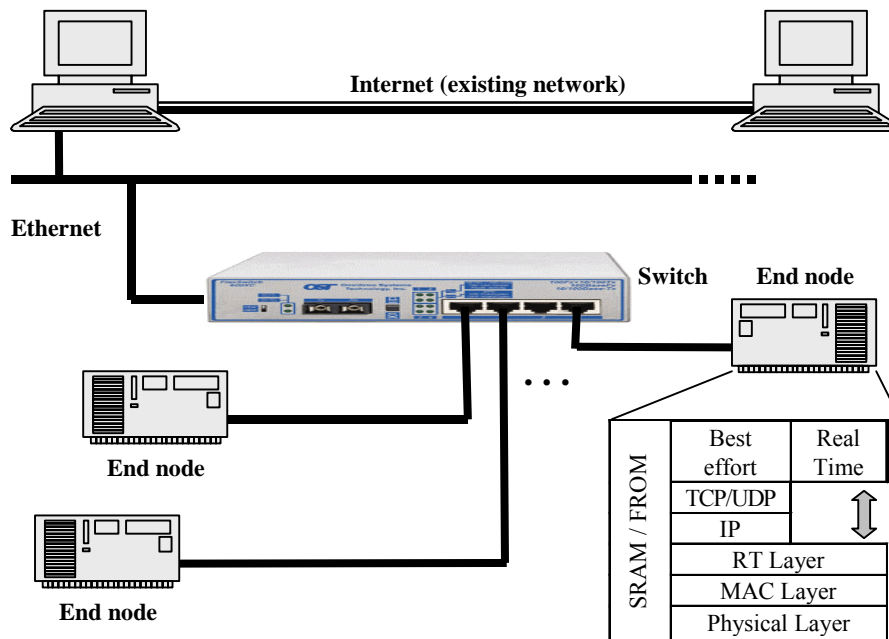


Fig. 4. LAN with full-duplex switched Ethernet

The boards include hardware and software components that provide network connectivity for the embedded systems. The components include:

1. A network-interface card (NIC) that plugged directly into the development board.
2. A SOPC Builder library component.
3. A C language library that provides an Ethernet Network-protocol stack.

We use a 5-ports Ethernet switch with full-duplex links at 100Mbps. The size of data packet is from 64 bytes to 1538 bytes. There are no modifications in the Ethernet hardware on the NIC (Network Interface Card). Ethernet development board is linked with PC via the Ethernet switch. In order to establish interaction and communication with the Ethernet development board, we downloaded the software (RT layer) to Flash Memory on the board. When download the software to the flash memory, the system module pins are logically connected to pins on the ACEX device. The external physical pins on the ACEX device are in turn connected to other hardware components on the board, allowing the Nios embedded processor to interface with SRAM, FROM, LEDs, LCDs, buttons and switch.

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:

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

where  $T_{n1,i}$  is the latency from source nodes to switch,  $T_{n2,i}$  is the latency from switch to destination nodes, and  $T_i$  is the total 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. 5 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 [19]. 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.

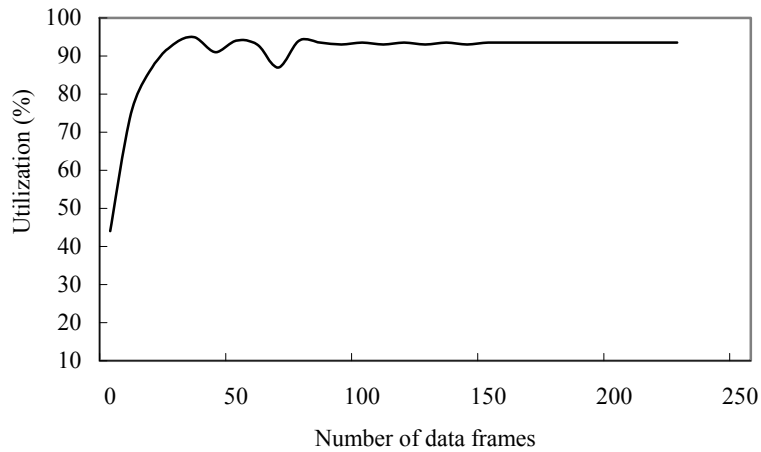


Fig. 5. Utilization on real-time data frame.

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. 6 shows the comparison of the data frame transmission latency of these three kinds of hard real-time communication protocols.

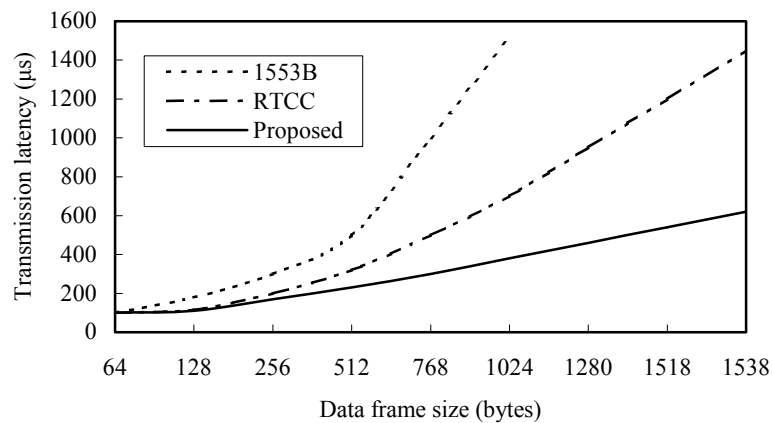


Fig. 6. Transmission latency of data frame

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. Runtime overhead of data frames are demonstrated in Fig. 7. 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.

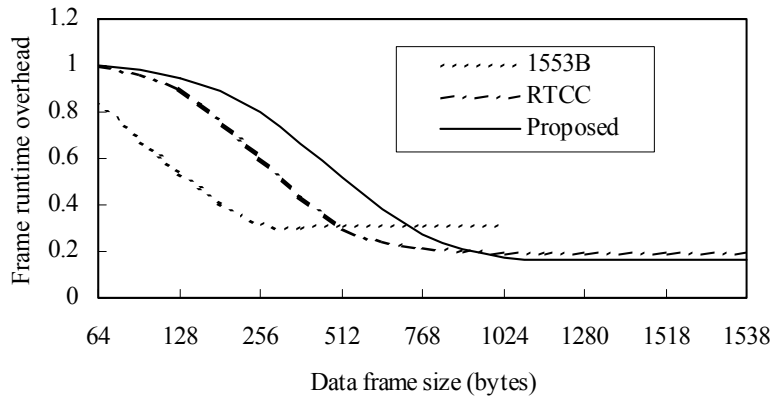


Fig. 7. Runtime overhead of data frame

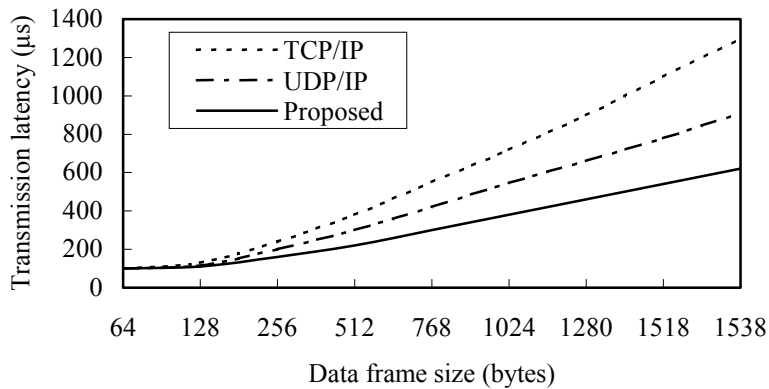


Fig. 8. Transmission latency comparison

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. 8). 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 RT channel establishment and EDF scheduling.

## 7. Conclusion

In this paper, we have presented a simple and efficient switched Ethernet communication protocol for industrial hard real-time applications. The network is set up with nodes and a switch; both switch and end nodes have an RT layer added to support hard real-time traffic. 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 speed up 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. This allows connecting the Ethernet LAN to existing Internet networks. Thus, it can be adopted in industrial hard real-time applications such as embedded systems, distributed control systems and robotics.

We have constructed a simple Ethernet LAN with the proposed protocol and evaluated the 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 meets the requirements of reliability for hard real-time systems.

One of our further works is to implement the proposed protocol in the multi-layer, many switches real-time communication systems.

## References:

1. G. Buttazzo: Hard Real-time Computing Systems, Predictable Scheduling Algorithms and Applications - Real-Time Systems Series 2ND Edition, Springer Verlag, (2004)
2. J. Stankovic and K. Ramamritham: Hard Real-Time Systems, IEEE Computer Society Press, (1988)
3. C. Krishna and K.G. Shin: Real-Time Systems, McGraw-Hill International edition, (1997)
4. M. Joseph: Real-Time Systems, Prentice Hall, (1996)
5. IEEE Std. 802.3, 2000 Edition: IEEE Standard for Information technology -- Telecommunications and information exchange between systems -- Local and metropolitan area networks -- Common specifications -- Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications.
6. <http://www.condoreng.com/support/downloads/tutorials/MIL-STD-1553Tutorial.PDF>
7. 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, Proceedings of 7th Australasian Conference on Parallel and Real-Time Systems (PART 2000), pp.161-170, (2000)

8. J. Loeser and H. Haertig: Low-latency hard real-time communication over switched Ethernet, Proceedings of 16th Euromicro Conference on Real-Time Systems (ECRTS'04), pp. 13-22, (2004)
9. S. Ouni and F. Kamoun: Hard and soft real time message scheduling on Ethernet networks, Proceedings of the 2nd IEEE International Conference on Systems, Man and Cybernetics of the twenty-first century, 6, (2002)
10. D. Ferrari and D. Verma: A scheme for real-time channel establishment in wide-area networks, IEEE Journal of Selected Areas in Communications, vol.8, no.3, pp.368-379, (1990)
11. L. Zhang: Designing a new architecture for packet switching communication networks, IEEE Communications Magazine, vol. 25, n. 9, pp. 5-12, (1987).
12. 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, pp 327-339, (1994)
13. A. Leon-Garcia and I. Widjaja: Communication Networks - Fundamental Concepts and Key Architectures, McGraw-Hill Osborne, (2001)
14. M. Schwartz & T. E. Stern: Routing Protocols, in Computer Network Architectures and Protocols (Second Ed.), Ed. C.A. Sunshine, Plenum Press, New York / London (1989)
15. S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss: An architecture for differentiated service, RFC 2475, 1998.
16. E. Rosen, A. Viswanathan, and R. Callon: Multiprotocol label switching architecture, RFC 2005, 1997.
17. L. Zhang, S. Deering, D. Estrin, S. Shenker, and D. Zappala: RSVP: a new resource reservation protocol, IEEE Network Magazine, 30-9, pp. 8-18, (1993).
18. 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, 20-1, pp.46-61, (1973)
19. Phillip A. Lptane: Real-Time System Design and Analysis, IEEE Press, third edition, (2004)