REAL-TIME COMMUNICATION AND ADMISSION CONTROL OVER RESPONSIVE LINK

Yuji Fujita, Shinpei Kato, and Nobuyuki Yamasaki
School of Science for Open and Environment Systems
Keio University
Kouhokuku, Yokohama City, Japan
\{fujita,shinpei,yamasaki\}@ny.ics.keio.ac.jp

ABSTRACT
This paper presents a framework of real-time communications over Responsive Link, which is one of the communication standards for real-time networks. In the Responsive Link, each packet has a priority at the hardware level and higher-prioritized packets can overtake lower-prioritized packets on network switches. Although Responsive Link is suitable for real-time communication in multihop networks, it is necessary for Responsive Link to be managed by software in order to establish the sufficient number of connections with timing constraints. In our approach, the communication delay of any connection is guaranteed by the admission control before the transmission to meet the timing constraints of any message which belong to the connection. We first introduce the notion of the virtual deadline assignment between two nodes lying side-by-side on the connection. We then propose an efficient admission control in which the virtual deadline is re-assigned. The simulation results show that the proposed methods improve the acceptance ratio of establishing the connections compared to a orthodox method.

KEY WORDS
Real-Time Communication, Responsive Link, Multihop Network, Schdulability Analysis, Admission Control

1 Introduction
Distributed real-time processing has received considerable attention in embedded real-time systems because of various factors such as performance, fault-tolerance, and so on. In order to realize distributed real-time processing, it is necessary to consider timing constraints of communications between nodes as well as computations on each node.

Currently there are a lot of communication standards. Ethernet is the most well-known global standard for office environments. However it is difficult to guarantee a real-time delivery of messages in the Ethernet due to collisions on the network that occur for the case in which more than one node transmit messages at the same time. In recent years, several researches have been conducted to support real-time communication over Ethernet. In [11], a technique using the switched Ethernet that provides micro-segmentation of traffic, store-and-forward and full-duplex links was proposed. In [5], a scheduling algorithm that determines when to transmit a message on the Ethernet switch was presented. In [9], a timing control of the message transmission by real-time OS was considered. In [8], the authors achieved the guaranteed communication delay of approximately 500 microseconds and insisted that the delay cannot be of any problem in industrial applications. However the guaranteed delay must be shorter in increasingly complex distributed real-time systems in the future.

Several network systems have been devised to meet such time-constrained requirements[2][7]. These systems can guarantee the required real-time capabilities only if the systems are organized as a single local area network and the communication speed is comparatively slow. Hence these systems cannot support multihop networks and cannot provide wide bandwidth communications with timing constraints. Responsive Link[13][12] is a communication standard that has been developed for overcoming the current problems in distributed real-time processing. Although the Responsive Link is suitable for real-time communication in multihop networks as we describe in Section 2, it is necessary for Responsive Link to be managed by software in order to establish the sufficient number of connections with timing constraints.

This paper presents a framework of real-time communications over the Responsive Link in multihop networks. The proposed approach is based on a real-time channel that is an unidirectional connection between the source and destination. Once the connection is established, the real-time channel is guaranteed to meet the user-specified performance requirements. In our approach, the communication delay of any connection is guaranteed by the admission control before the transmission to meet the timing constraints of any message which belong to the connection. Moreover, we propose two methods which achieve high probability of accepting the requests of establishing a real-time channel.

The rest of the paper is organized as follows. The next section describes the feature of the Responsive Link. In Section 3, the network architecture we consider in this paper is introduced. In Section 4, our approach of real-time communication is presented. The methods for improving the acceptance ratio is also proposed. The simulation results show the advancement of our methods in Section 5.
Finally we conclude this work and discuss the future work in the last section.

2 Responsive Link

Responsive Link is the communication standard for real-time communication implemented in Responsive Multi-threaded Processor (RMTP)[12]. The features of the Responsive Link is as follows.

- Each link is composed of two channels: data channel with 64-byte packets for high-throughput transmissions and event channel with 16-byte packets for low-latency transmissions.
- Every packet is assigned with a priority, which controls packet overtaking at Responsive Link switches.
- A packet priority can be replaced with a new priority level at each node.
- Packets of different priorities can take different paths.
- A packet has per-byte ECC for 1-bit error correction and 2-bits error detection.
- Applications can choose variable link speed (800, 400, 200, 100, 50, 25, 12.5[Mbaud])

The Responsive Link has been standardized as the IPSJ Trial Standard (IPSJ-TS 0006:2003), which is domestic standard in Japan[1]. Also it will be standardized by the ISO/IEC JTC1 SC25 WG4. We believe that real-time communication among different kinds of systems can be realized once this international standardization is realized.

This paper makes use of the features of the priority-based packet overtaking and the priority replacement to realize real-time communication on the Responsive Link.

3 Network Architecture

Each node can be connected by the Responsive Link with point-to-point and full-duplex. We consider a network where each packet hops several intermediate nodes, i.e., a multihop network (Figure 1). The transmission mode is selected from cut-through or store-and-forward. Because the cut-through mode is often used in the practical cases, an input packet is transmitted at the time when the header of the packet is read. This mode realizes low latency packet transmissions.

Each packet has 256-level priorities in order to realize packet overtaking. If there is no collision at a node, an input packet is transmitted to output port. If the input packets from the different ports are transferred to the same port, a collision occurs. Then the packets with lower priorities are stored into the buffers to wait for being output and the packet with the highest priority is output first. After the packets with higher priorities are output, the packets with lower priorities can be output. In other words, the packets with higher priorities overtake the packets with lower priorities by hardware. Additionally, the original priority of the input packet can be replaced with a new priority at a node. The original priority is used for overtaking packets at the current node, and the new priority is used at the next node.

4 Real-time Communication

4.1 Routing

There are multiple paths between the source and destination nodes (Figure 1). Adopting dynamic routing enables the connections to choose the optimum path according to the network state. However the dynamic routing algorithm requires a number of message exchanges which incur a large overhead at runtime. And it is difficult to predict the communication delay when each packet goes through a different path. Therefore we take the static routing algorithm such as the Shortest Path First (SPF) in this paper. According to the static routing algorithm, each node determines routing using the static information about the network and the packets on the established connection always go through the same path. Further details of the routing algorithm are out of scope of this paper.

4.2 Real-Time Channel Establishment

To guarantee the delivery of real-time messages before the deadline, a real-time channel[6] must be established before the transmission of packets on the connection starts. When a node requires real-time data, it sends a request message for establishing the real-time channel. For this reason, a real-time channel is tried to be established from the destination node to the source node.

A real-time channel \( M_i \) is characterized by \( \{T_i, C_i, D_i\} \), where \( T_i \) is the period of the message,
$C_i$ is the amount of the message per period, and $D_i$ is the relative deadline. $T_i$, $C_i$, and $D_i$ are all expressed as the number of packet sizes. In this paper, we assume that the relative deadline is equal to the period.

The creation of a real-time channel consists of the following four phases.

- **Request Phase**
- **Routing Phase**
- **Reservation Phase**
- **Relaxation Phase**

After that, the nodes can use the channel. In [6], routing and resource reservation are conducted at the same time, and the creation of a real-time channel consists of two phases. The reason why the four phases are used in this paper is that the number of hops and amount of workload are used for deciding the priority of the packet. Therefore two additional phases are introduced in order to calculate the number of hops and amount of workload.

Figure 2 shows a message flow when a real-time channel establishment is accepted, and Figure 3 shows a message flow when a real-time channel establishment is rejected. The numbers in front of each message represent the order of sending messages. Processes of nodes at each phases are described below. At the request phase, the source node sends a request message to the destination node for the real-time channel establishment. The intermediate node transfers a received request message to the next node. At the routing phase, the destination node sends a routing message in order to determine the path of the real-time channel. The intermediate node which receives the routing message determines the next node, and transfers the routing message to the next node until it reaches the destination node. When the routing message reaches the destination node, the number of hops and amount of workload are calculated for deciding the priority of the packet. At the reservation phase, each node of the connection determines the priority of the connection and checks its schedulability. If it is possible to guarantee the real-time capabilities of all the existing connection and the establishing connection between two nodes lying side-by-side on the connection, the node transfers a reservation message to the next node. In contrast, if it is impossible to guarantee the real-time capabilities, the node sends a reservation error message to the next node and an error message to the previous node. When the intermediate node receives the reservation error message, it transfers the message to the next node without any priority assignment and the schedulability test. At the relaxation phase, the destination node notifies the source node whether the real-time channel is successfully established. If the intermediate node receives a success message, it transfers the message to the next node. In contrast, if the intermediate node receives an error message, it releases the reserved resources and transfers the message to the next node.

### 4.3 Real-Time Packet Scheduling

Since the priority-based packet overtaking of the Responsive Link corresponds to the context switch of the process,
the mechanism of switching on the nodes can be considered as the conventional fixed-priority scheduling of real-time OS. It is widely known that the Rate Monotonic (RM) algorithm[10] is optimal for fixed-priority scheduling. In the RM algorithm, a task with a shorter relative deadline has a higher priority. The DM algorithm is optimal under the condition that the relative deadline is shorter than the period. Here we define the prioritization based on the virtual deadline extended from the DM algorithm as Virtual Deadline Monotonic (VDM).

We consider that the schedulability of the connection with the large number of the hops can be improved by using the virtual deadline described by Equation (1). However there is a possibility of nonuniform workload of each link on the establishing channel. In that case, it is not suitable to divide the deadline equally. The packets at the link with lower workload tend to be wait for other packets, whereas the packets at the link with higher workload tend not to wait for other packets. Therefore it is possible to guarantee the real-time capabilities of the connections by decreasing the virtual deadlines at low workload links and increasing them at high workload links. Consequently we propose the method to set the virtual deadline according to workload as below where \( l_{i}^{n,m} \) is the workload of \( M_{i}^{n,m} \) and \( L_{i} \) is the summation of \( l_{i}^{n,m} \) belong to \( M_{i} \).

In Equation (2), \( (D_{i} - C_{i} \times h_{i}) \) is the duration for which \( M_{i}^{n,m} \) may be waited by other connection. The time is divided according to the workload. The workload is calculated by Equation (3) where \( v_{n,m}^{i} \) is the set of connections through the link from node \( n \) to node \( m \).

\[
v_{d_{i}}^{n,m} = (D_{i} - C_{i} \times h_{i}) \times \frac{l_{i}^{n,m}}{L_{i}} + C_{i}  \tag{2}
\]

In Equation (2), \( (D_{i} - C_{i} \times h_{i}) \) is the duration for which \( M_{i}^{n,m} \) may be waited by other connection. The time is divided according to the workload. The workload is calculated by Equation (3) where \( v_{n,m}^{i} \) is the set of connections through the link from node \( n \) to node \( m \).

\[
l_{i}^{n,m} = \sum_{v_{n,m}^{i} \in M_{i}} \frac{C_{i}}{T_{i}}  \tag{3}
\]

The transmission mode of the Responsive Link can be selected cut-through. When the packet arrives at the node, the packet is transmitted immediately if there are no packets with the higher priorities. It can be considered that the packet transmission is occurred by one packet of delay at the next node in the best case, which is no packet competing for the out port. Therefore the virtual deadline can be represented as below where \( C_{packet} \) is the size of the one packet.

\[
v_{d_{i}}^{n,m} = D_{i} + (h_{i} - 1) \times (C_{i} - C_{packet}) \tag{4}
\]

In addition, when the workload was considered, the virtual deadline is described as below.

\[
v_{d_{i}}^{n,m} = (D_{i} - C_{i} + (h_{i} - 1) \times C_{packet}) \times \frac{l_{i}^{n,m}}{L_{i}} + C_{i} \tag{5}
\]

4.3.1 Virtual Deadline

In our approach, each packet is prioritized based on the number of the hops in order to improve the schedulability of the connection with the large number of the hops. We introduce the notion of virtual deadline which is not a real deadline but is used to determine the priority of the link between two nodes lying side-by-side of the connection. \( M_{i}^{n,m} \) denotes the link from node \( n \) to node \( m \) on the establishing channel \( M_{i} \). Note that node \( n \) and node \( m \) are lying side-by-side. The virtual deadline of \( M_{i}^{n,m} \) expressed by \( v_{d_{i}}^{n,m} \) is simply defined as below where \( h_{i} \) is the number of the hops.

\[
v_{d_{i}}^{n,m} = \frac{D_{i}}{h_{i}},  \tag{1}
\]
line to guarantee real-time capabilities of communication. We conduct the schedulability test based on the Response Time Analysis (RTA) [3]. The worst-case response time is predicted by the RTA, and the connection is tested whether to be schedulable by comparing the worst-case response time with the virtual deadline. The response time in packet scheduling between two nodes lying side-by-side is the time duration between the message arrives at the node and the node finishes sending the message to the output node.

In multihop networks over the Responsive Link, it is impossible to know precisely when the packet input is, because the arrival of the packet depends on the workload of paths. This situation can be considered to be the same as the release jitter problem in [3]. We consider the permissible maximum delay \( T_j - C_j \) of the real-time message \( M_j \) with higher priority as the release jitter. Hence the worst-case response time of \( M_{i,n,m} \) is calculated by Equation (6) where \( h_{p(i)} \) is the set of connections with higher priorities than the priority of \( M_{i,n,m} \).

\[
W_{i,n,m} = \sum_{\forall M_{j,n,m} \in h_{p(i)}} \left( \left( \frac{W_{j,n,m} + T_j - C_j}{T_j} \right) \times C_j \right) + C_i \tag{6}
\]

In Equation (6), \( W_{i,n,m} \) appears on both sides of the equation. It is possible to solve this equation as below.

\[
Z^i_{j+1} = \sum_{\forall M_{j,n,m} \in h_{p(i)}} \left( \left( \frac{Z^i_{j} + T_j - C_j}{T_j} \right) \times C_j \right) + C_i \tag{7}
\]

The iteration starts with \( Z^i_0 = C_i \) and terminates when \( Z^i_{j+1} = Z^i_{j} \). Then the value \( Z^i_{j} \) is the worst-case response time \( W_{i,n,m} \).

### 4.5 Admission Control

When each node checks the schedulability of the establishing channel, it uses the schedulability test proposed in Section 4.4. The test compares the worst-case response time with the virtual deadline of the establishing channel. The test also compares the worst-case response time with the virtual deadline of the already established channel with the lower priorities than the establishing channel. If the worst-case response time is larger than the virtual deadline, the new channel establishment is rejected. Only when real-time capabilities of the establishing channel and the already-established channels passing through the same link are guaranteed, the new channel establishment is accepted.

In the channel establishment proposed in [6], if any test fails at the node and the channel cannot be established, the node will send the message in order to try sending the message towards the destination along another path. Although the channel establishment proposed in this paper cannot select another path towards the destination node because of static routing, there is the method that may accept the channel which cannot be established. We use the virtual deadline, which is not real deadline. In our case, if the message of the channel meets the relative deadline of end-to-end nodes, it is possible to accept the channel establishment even if the virtual deadline is missed.

It is necessary to analyze the schedulability of the establishing channel in order to check to meet the relative deadline. We introduce the following variables. \( o_{i,n,m} \) called the overrun time, is represented as \( o_{i,n,m} = W_{i,n,m} - vd_{i,n,m} \), and \( O_i \) denotes the sum of \( o_{i,n,m} \) belonging to \( M_i \), \( r_{i,n,m} \), called the remaining time, is represented as \( r_{i,n,m} = vd_{i,n,m} - W_{i,n,m} \), and \( R_i \) denotes the sum of \( r_{i,n,m} \) belonging to \( M_i \). Each node has \( o_{i,n,m} \) and \( r_{i,n,m} \) in the table of the node, and \( O_i \) and \( R_i \) are added in the fields of the success messages and the reservation messages. The default values of \( O_i \) and \( R_i \) are 0.

We explain the admission control of each node at the reservation phase. Figure 5 shows the admission control of the intermediate nodes at the real-time channel establishment of \( M_i \). If \( M_{i,n,m} \) denote the set of connections with lower priorities than the priority of \( M_{i,n,m} \). Firstly the worst-case response time of the already established channels with lower priority than \( M_i \) is calculated (line 2), and is compared with their virtual deadlines (line 3). If one any channel does not meet the virtual deadline, the real-time channel establishment of \( M_i \) is rejected (line 3–5). Secondly the node calculates the worst-case response time of

![Figure 5. Admission control of the intermediate node](image)

![Figure 6. Admission control of the destination node](image)
The node compares the worst-case response time with the virtual deadline of $M_i$ (line 7). Finally the node assigns a value to the variables (line 9–13). Finally the node sends the reservation message with $O_i$ and $R_i$ to the next node (line 15). Figure 6 shows the admission control of the destination node at the real-time channel establishment of $M_i$. If $O_i$ remains the default value 0, the destination node sends a success message (line 1–2). If not, the destination node compares $O_i$ with $R_i$ (line 4). If $O_i$ is larger than $R_i$, the real-time channel establishment is rejected (line 5). If $R_i$ is larger than $O_i$, the real-time channel establishment is accepted, and the destination node sends the message so that the intermediate nodes can update the virtual deadline in the relaxation phase as follows (line 7).

When the worst-case response time exceeds the virtual deadline at a node (but the relative deadline is not missed), it is necessary to update the virtual deadline for the schedulability test of the next real-time channel establishment at the relaxation phase. This update uses the values that are got at admission control. If the worst-case response time exceeds the virtual deadline by the schedulability test in the node, the node updates the virtual deadline at receiving the success message according to (8).

$$vd_{\text{new},i}^n = vd_{i}^n + o_i^n,$$  \hspace{1cm} (8)

where $vd_{\text{new},i}^n$ is the updated virtual deadline. If the worst-case response time does not exceed the virtual deadline by schedulability test in the node, the node update the virtual deadline at receiving the success message according to (9).

$$vd_{\text{new},i}^n = vd_{i}^n - \frac{r_i^n}{R_i} \times O_i$$ \hspace{1cm} (9)

5 Evaluation

In order to evaluate the effectiveness of the priority assignment by the virtual deadline and the admission control proposed in this paper, we conducted several sets of simulations. In this section, we compare the acceptance ratio of the real-time channel establishment and the network utilization using the RM algorithm, the VDM algorithm, and the VDM algorithm with the proposed admission control.

In this paper, the network utilization $U$ is defined as below where $N$ is the number of the unidirectional link between two nodes lying side-by-side.

$$U = \sum \frac{C_i}{T_i} \times \frac{1}{N}$$  \hspace{1cm} (10)

In the RM algorithm, a connection with a shorter period has a higher priority. Because the RM algorithm does not use the virtual deadline, the admission control we proposed cannot be applied. In the VDM algorithm, we derive the virtual deadline from Equation (5) because it is the most effective way in Section 4.3.1. As the network topology, we used a two-dimensional torus which consists of 16 nodes in Figure 7 and a tree structure which consists of 15 nodes in Figure 8. We assume that the links connecting nodes consist of two unidirectional links running in opposite directions.

We simulated the real-time channel establishments over two networks. We use the SPF algorithm as the routing algorithm. Every requested channel is generated with uniform distributions according to the following parameters: the period in $[100,1000]$, the relative deadline equal to the period, and the amount of the message per period in $[10,50]$. The source-destination pair of the channel is chosen randomly. First we established a real-time channel between each pair. When the network utilization reached a certain amount, we stopped the channel establishment. Next we send the channel establishment requests and conducted the admission control. The channels were not established regardless of the results of the admission control. After that, we counted the number of the accepted channels, and divided the sum by the all channel establishment requests. The resulting number is the acceptance ratio.

The acceptance ratio under various network utilisations for the two-dimensional torus is shown in Figure 9. The VDM approach improves the acceptance ratio up to 53%. This means that using the VDM algorithm for the priority assignment is much better than using the RM algo-
Algorithm. The proposed admission control shows only slight improvement.

Figure 10 shows the situation of the tree structure. The VDM approach improves the acceptance ratio by approximately 15%, and the proposed admission control further improve by approximately 15%. We consider that the proposed admission control through the links with heavy workload and light workload. In such a situation, although the virtual deadline may be met at the some links the virtual deadline is likely missed at the other links. The paths of the channel is centralized at the links of root node in the tree structure. Therefore high acceptance ratio is achieved by the proposed admission control.

Figure 11 and Figure 12 show the increasing network utilization as the channel establishment is required, and that VDM approach achieves higher utilization than RM. The achievement of high network utilization means that the network can accept the channels with large number of packets. In our approach, the real-time channels which require wide bandwidth can be established by using the network resources efficiently.

6 Conclusion

This paper proposed a framework of real-time communication over Responsive Link. In our approach, the communication delay of any connection is guaranteed by the admission control before the transmission to meet the timing constraints of any message which belong to the connection. In addition, we proposed the priority assignment method and the admission control, which realize high acceptance ratio of the real-time channel establishment. The priority is assigned to the packet based on the virtual deadline which is not a real deadline between two nodes lying side-by-side. In the admission control we proposed, when the channel establishment is rejected by missing the virtual deadline, it is possible to accept the channel establishment by reassigning the virtual deadline if the channel meets the relative deadline of end-to-end nodes. The simulation results show that the proposed methods improve the acceptance ratio of establishing the connections, and achieve high utilization of the network. Thus it is possible to managed real-time communication on Responsive Link efficiently.

When the real-time channel establishment is rejected, it can be accepted by searching another route. Also in order to increase the acceptance ratio, we can introduce dy-
Dynamic routing. For these reasons, it is necessary to work on the routing problem moreover. In addition, we can strictly evaluate proposed method by deriving the limit of the utilization of the real-time network. We will address these problems as future work.

Acknowledgement

This research was supported by the fund of CREST, JST.

References


