Fan, X. and M. Jonsson, "Efficient support for high traffic-volumes of short-message real-time communication using an active Ethernet switch," Proc. 10th International Conference on Real-time and Embedded Computing Systems and Applications (RTCSA'04), Göteborg, Sweden, Aug. 25-27, 2004, pp. 517-533.

Efficient Support for High Traffic-Volumes of Short-Message Real-Time Communication Using an Active Ethernet Switch

Xing Fan and Magnus Jonsson

Halmstad University, Box 823, S-301 18 Halmstad, Sweden
{Xing.Fan, Magnus.Jonsson}@ide.hh.se

Abstract. There are several different types of communication traffic with real-time demands apparent in distributed industrial and embedded systems, for example, group communication and process synchronization. The length of these messages is often very short but the traffic volume might be really high. Standard network protocols do not normally reach very high utilization for such small messages. This paper presents a solution to efficiently support real-time short message communication over switched Ethernet. In our proposal, the Ethernet switch and the end-nodes are enhanced to combine several short messages into an Ethernet frame to improve the performance, and to give the short-message traffic real-time support on two levels, short-frame level and Ethernet frame level. Earliest Deadline First (EDF) scheduling is used in the switch and in the source nodes on both these two levels. We have characterized the performance of the network in terms of channel utilization and the number of accepted real-time channels, by simulations of the network assuming Fast Ethernet. We also show, by example, that we can reach an improvement of the possible short-message rate of 66%.

1 Introduction

In parallel distributed processing, a large part of computation overhead comes from communication. This can be minimized if the network protocol offers the user services aimed at specific types of communication used in these applications. In modern and future parallel and distributed applications and embedded systems, such as radar signal processing systems, network equipment, In-Vehicle Networks (IVNs), automation industry networks and control, there are Barrier Synchronizations (BS), Global Reduction (GR) or group communication, in which messages from one or more senders are delivered to a large number of receivers [1] [2] [3]. Moreover, a major fraction of communication patterns in these domains are small messages suffering relatively high overhead and poor performance during transmission [4]. For example, radar signal processing algorithms in general work on relatively short vectors and small matrices, but at a fast pace and with large sets of vectors and matrices. Even if incoming data from the antenna contains data is viewed in three dimensions (channel, pulse and distance), it is quite easy to divide the data into small tasks that each can be executed on a separate processing element in a parallel

computer. By reducing the combined overhead of protocols like MAC (18 bytes), LLC (4 bytes), IP (20 bytes) and UDP (8 bytes), the performance can be significantly improved.

Another critical item arises from the Quality of Service (QoS) requirement of these small-messages, whose correct performance is specified in terms of time constrained message transmissions. In radar signal processing, the application example mentioned above, the control traffic is periodic and must have a deterministically guaranteed delay-bound, while the data traffic is periodic and should have a probabilistically guaranteed delay bound.

These problems have received attention in networking and parallel and distributed computing research communities in recent years. There have been several attempts to achieve lower latency and better bandwidth and utilization, for example, reducing the network interface access time [5] [6]. However, the high performance network interface design only alleviate, not overcome the unsatisfying bandwidth characteristics of the small messages. In addition, disadvantages in higher costs are introduced. Moreover, some results have been published on a pipelined fibre-ribbon ring network that supports several simultaneous transmissions in non-overlap segments to achieve higher throughput [7] [8]. Offered services in this network include a guaranteed real-time service and other services for parallel and distributed processing such as barrier synchronization and global reduction.

Number of protocols and schemes have been proposed to improve real-time characteristics of switched Ethernet networks. The EtheReal switch [9] [10] includes a bandwidth reservation scheme inside the switches without any hardware or operating system modification. Support for periodic real-time traffic on an extended switched Ethernet network, using Earliest Deadline First (EDF) scheduling, have been reported [11]. A thin software layer is added between the Ethernet protocols and the TCP/IP suite in the end stations. The switch is responsible for admission control where the feasibility analysis is made for each link and direction between the end-nodes and the switch. More results on establishing real-time channels in packet-switched networks and using deadline sorting in the switch to gain real-time support have been presented [12] [13].

However, few of these efforts are targeting both real-time support and good performance for short-message traffic, at the same time. Our approach is to find a more satisfying solution to address both efficiency and time-constraints of small-sized packets. Since switched Ethernet is a good candidate for parallel processing systems, including clusters of workstations, we propose, in this paper, to enhance the Ethernet switch by adding “intelligence” to achieve our goal. The active Ethernet switch is a novel approach to provide good range of features and abilities that makes it a reasonable fit for a variety of applications. The networks are active in the sense that the nodes or the switches can perform customized computations on, and modify, the packet content flowing through them. We have previously published results on how to give efficient support for many-to-many communication over active switched Ethernet [14]. The switch reorganizes the data before transmitting it further. The analysis shows that the active switch solution achieves better performance than the ordinary switch. The active Ethernet switch research is extended in this paper, covering more user services than many-to-many communication, such as barrier-synchronization and global reduction together with general small-message support for

control traffic, etc. Several small messages can be combined into one Ethernet frame, and gain real-time support by using logical connections with guaranteed bit rate and bounded latency. The method of using such logical real-time channels in switched Ethernet has been presented [11] [14]. Considering the mini-frame transmission mechanism, a new two level real-time support is proposed in this paper.

The rest of the paper is organized as follows. In Section 2, we present the Ethernet network architecture and the parallel computing services. The efficient high traffic-volumes short-message real-time traffic handling is described in Section 3. In Section 4, the real-time analysis is described, while the simulation analysis is reported in Section 5. Finally, Section 6 offers conclusions.

2 Network architecture and parallel computing services

Switched Ethernet enables some key benefits over traditional Ethernet, such as full duplex, and flow control. It also directs network traffic in an efficient manner, establishing a sort of direct line of communication between two ports for each frame, and maintains multiple simultaneous links between various ports. Switched Ethernet is so prevalent and frequently used now and will probably take over much of the industrial bus and network markets in the future since it involves a cost-effective off-the-shelf technology. Taking these advantages, we have now put our focus on switched Ethernet to find efficient support for short-frame real-time communication.

There are a variety of communication patterns used in the global data movement and process control operations in parallel and distributed processing systems having very short messages. Data movement operations are often applied to different dimensions of data arrays, for example, sending a single datum to many other processors for use in a computation, array summation, determining the maximum and minimum values of an array, rearrangement of data for different purposes like transposing a matrix, or rotating data blocks and exchanging data in certain dimensions. In addition to data manipulation operations, control operations are an essential part of parallel processing. Barrier Synchronization (BS) is an operation to control the flow of processes in a distributed processing system. A logical point in a control flow of an algorithm is defined, at which all processes in a processing group must arrive at before any of the processes in the group are allowed to proceed further. Global Reduction (GR) is similar to BS, where data is collected from distributed processes when they signal their arrival at the synchronization point. A global operation, for example, sum, product or a logical operation, is performed on the collected data by the switch. At the end of the GR, all participating nodes have access to the same data.

When a BS point is encountered in the application program in a node connected to our Ethernet switch, the node sends the encountered BS-ID to the switch. The switch collects the BS messages in an internal table till all the nodes in the BS group have reached the BS point. Then the switch will send to all the participating nodes, notifying them to proceed. For GR, the switch performs the required operation on the collected GR data from the participating nodes, and sends the calculation result to all of them. The other parallel processing operations are performed in a similar way.

3 Traffic handling

The active switched Ethernet proposed in this paper provides control to enhance the real-time short-frame communication of the network by adding software to both the switch and the end-nodes. There are two main tasks of the switch. The first is efficient transmission support to high traffic-volumes of short messages. The second task is to support timely delivery of guaranteed periodic real-time short-frame traffic.

3.1 Protocol definition

To obtain efficient transmission, we combine a number of mini-frames (the term "mini-frame" is used with the same meaning as "short-message" in this paper) into one Ethernet frame before transmission to reduce the overhead of the traffic.

The contents of the short-message combined Ethernet frame are shown in Figure 1. A combined frame consists of the Ethernet header (LLC header), a parallel processing header and parallel processing data in the mini-frames. The first byte in the parallel processing header indicates the number of the mini-frames in the Ethernet frame, while each of the following 2-byte offset fields indicates the start position of the related parallel processing data in the Ethernet frame. Each mini-frame consists of five fields: ServiceID (1 byte), GroupID (1 byte), SeqNo. (1 byte), data length (1 byte), and data (0-64 bytes). The field labeled ServiceID indicates the kind of service for the mini-frame. Some service types and their assigned values are shown in Table 1. The GroupID field tells which communication group the mini-frame belongs to. The SeqNo. is used as an index if the parallel processing data is divided into several mini-frames. The data length field tells the length of the data (in bytes) in the mini-frame. The last field of the mini-frame contains the data. In this paper, the data field in a mini-frame is limited to 64 bytes, allowing for, e.g., a vector of eight 2×32 bit complex numbers. According to this definition, the maximum number of mini-frames in an Ethernet frame, Q, is 21 following the IEEE 802.3 standard of maximum total frame length.

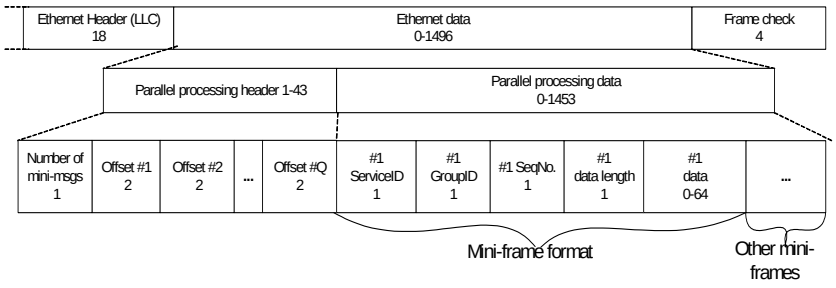


Fig. 1. Format of the mini-frame combined Ethernet frame. The number in each field gives the size of the field measured in bytes

Table 1. The assigned value to the parallel processing services

Assigned Value	Service Type	Assigned Value	Service Type
1	BS	7	Logical OR
2	Global sum	8	Logical AND
3	Global Multiplication	9	Exclusive OR
4	Global Min	10	Multicast
5	Global Max	11	Gather
6	Corner-turn	0, 12-255	Reserved for other services

For simplicity, we assume, in this paper, that the programmer (or the compiler) statically allocates the necessary parameters for the parallel computing, e.g., BS_IDs, GR_IDs, and the IDs of the participating nodes off-line before run-time. With minor adjustments, dynamic allocation is also possible but is not investigated in this paper.

The improvement of the short message rate is shown by the following analysis. We denote R_I , R_O and R_{IP} as the maximum short message rate (the number of short messages per second) in our active switched Ethernet sending 21 mini-frames per frame, in a normal switch where one Ethernet frame carries one short message, and in a normal switch using UDP/IP, respectively, which can be derived as follows:

$$\begin{aligned} R_I &= \frac{N_{BW}/8}{L_{EF}/Q} = \frac{QN_{BW}}{8L_{EF}} \\ R_O &= \frac{N_{BW}}{8\left(L_{MF}+H_{EF}+H_{MF}\right)} \\ R_{IP} &= \frac{N_{BW}}{8\left(L_{MF}+H_{EF}+H_{MF}+H_{IP}+H_{UDP}\right)} \end{aligned} \tag{1}$$

where N_{BW} is the bandwidth of the network (bits/s), L_{EF} is the length of Ethernet frame (bytes), L_{MF} is the length of a mini-frame (bytes), H_{EF} is the length of the Ethernet header (bytes), H_{MF} is the length of the mini-frame header (bytes), H_{IP} is the length of the IP header (bytes), and H_{UDP} is the length of the UDP header (bytes). The short message rate improvements, IMP_1 (compared with the normal Ethernet switch) and IMP_2 (compared with the normal Ethernet switch using IP and UDP), would look like:

$$\begin{aligned} IMP_1 &= \frac{R_I - R_O}{R_O} \\ IMP_2 &= \frac{R_I - R_{IP}}{R_{IP}} \end{aligned} . \tag{2}$$

We calculate IMP_1 and IMP_2 with the assigned values for the parameters, which are listed in Table 2. The result of IMP_1 is about 27%, while the result of IMP_2 is about 66%, which prove the significant performance improvement by using our active Ethernet switch.

Table 2. The assigned value to the parameters in the performance improvement calculation

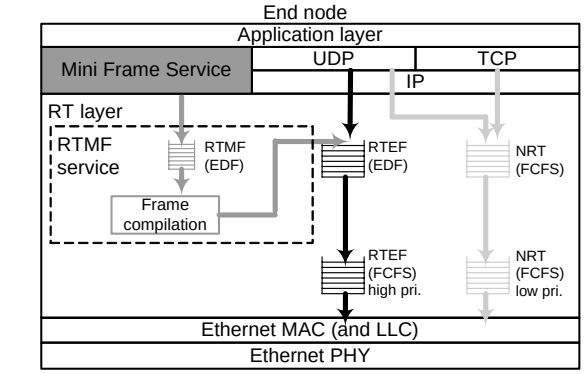
Parameter	Assigned Value	Parameter	Assigned Value
L_{EF}	1518 bytes	L_{MF}	64 bytes
H_{EF}	22 bytes	H_{MF}	6 bytes
H_{IP}	20 bytes	H_{UDP}	8 bytes
N_{BW}	100 Mbits/s	Q	21

3.2 Two-level real-time support

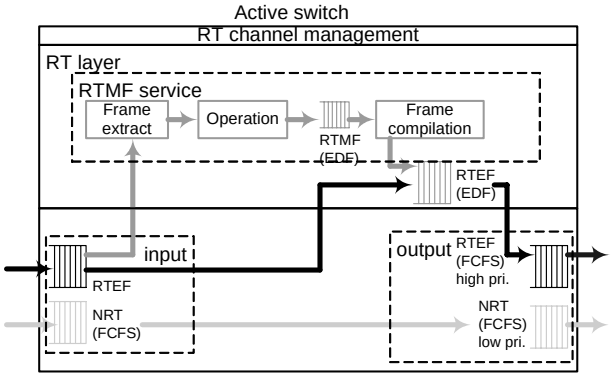
Figure 2a and 2b shows a layered view of the real-time mini-frame support at the end node and at the active switch, respectively. In the system, there are different kinds of traffic and different traffic input/output queues: non-real-time traffic and queues, real-time mini-frame traffic and queues and other real-time traffic and queues. The real-time service is done by the RT (real-time) layer, where service for real-time mini-frame traffic is provided by the RTMF service function block.

In each end node, the real-time mini-frames are put into the real-time mini-frame queue (RTMF-queue) first, while other real-time traffic is put directly into the real-time Ethernet frame queue (RTEF-queue) in the RT layer. If there are several mini-frames in the RTMF-queue, they are combined, as many as possible but less than the maximum limit of 21, into one Ethernet frame by the frame compilation. Then the mini-frame combined Ethernet frames are moved into the RTEF-queue. Both the RTEF-queue and the RTMF-queue are sorted according to earliest deadline first (EDF) [15]. Outgoing non-real-time traffic from the end-node typically uses TCP (non-real-time traffic can use UDP as well) is put into an FCFS-sorted (First Come First Serve) queue in the RT layer which has lower priority than the RTEF-queue.

The RT layer in the switch, shown in Figure 2b, contains similar queuing mechanisms as the end-nodes. An incoming real-time Ethernet frame to the switch is directly put into the RTEF-queue in the RT-layer, while the mini-frame combined Ethernet frames are extracted first by the frame extract in the RTMF service block. The specified operation is then performed on each mini-frame, possibly waiting for other mini-frames first, before passing it (or generating a new mini-frame) to the RTMF-queue. For example, if GR is the specified function, the operation block generates a new mini-frame with the result data as soon as all the GR mini-frames with input data in the group are collected. If the mini-frame carries multicast data, it will be put into the RTMF-queues directly.



(a)



(b)

Fig. 2. Real-time mini-frame support layers and queues. (a) in an end node, (b) in the active switch

The network supports dynamic addition of real-time channels for Ethernet frames (RTEF-channels) and real-time channels for mini-frames (RTMF-channels) by the RT channel management software in the switch, each RTEF-channel and RTMF-channel being a virtual connection between two nodes in the system with guaranteed bit rate and bounded delay. A special RTEF-channel is allocated for the mini-frame combined Ethernet Frames, which is called real-time simulated medium (RTSM-channel). The real-time channel establishment and delay bound analysis are described in Section 4.

4. Real-time analysis

Depending on the characteristics of the application, the maximum delay bound of the short messages is specified as one of the QoS metrics. The time constraints are guaranteed by setting up logical real-time channels, which are characterized by certain parameters. The values of the parameters are derived from the delay bound. However, for easy understanding, in this section, the real-time channel establishment is described first, while the delay bound analysis is then derived.

4.1 Introduction

The real-time guarantee for the RT Ethernet frames is uphold by an RTEF-channel with index i, which is characterized by:

$$\{T_{period,i}, C_i, T_{deadline,i}\} \quad (3)$$

where $T_{period,i}$ is the period of data generation, C_i is the amount of data per period, and $T_{deadline,i}$ is the relative deadline used for the end-to-end scheduling. We assume that:

$$T_{deadline,i} = T_{D1,i} + T_{D2,i} \quad (4)$$

where $T_{D1,i}$ and $T_{D2,i}$ are the deadlines for RT Ethernet frames from the source node to the switch and from the switch to the destination node, respectively. $T_{deadline,i}$ can be partitioned in a number of ways. For simplicity, we here assume that

$$T_{D1,i} = T_{D2,i} = T_{deadline,i} / 2. \quad (5)$$

As we mentioned in Section 3.2, a logical RT channel, the RTSM-channel, is allocated specially for whole RT mini-frame combined Ethernet frames. The RTSM-channel is sharing the same physical link with other RTEF-channels. In other words, the RTSM channel is a RTEF channel but only serves for the RT mini-frame combined Ethernet frames. The end-to-end RTSM-channel is characterized as:

$$\{T_{P\_SM}, C_{SM}, T_{D\_SM}\} \quad (6)$$

where T_{P_SM} is the period of mini-frame combined Ethernet frames generation, C_{SM} is the maximum amount of data per period, and T_{D_SM} is the relative deadline used for the end-to-end scheduling. T_{D_SM} is assumed to be partitioned equally into T_{D1_SM} and T_{D2_SM} , the deadline of the RT mini-frame combined Ethernet frames from the source node to the switch and from the switch to the destination node, respectively.

4.2 Real-time mini-frame communication

Below, we explain how to implement real-time support for mini-frame traffic. As shown in Figure 2, our real-time support for mini-frame traffic is on two levels. The

RTSM-channel is allocated firstly, before the logical RT channels for the mini-frames (RTMF-channel) are allocated. The RTMF-channel with index *j* is characterized by:

$$\{T_{P\_MF,j}, C_{MF,j}, T_{D\_MF,j}\} \tag{7}$$

where $T_{P\_MF,j}$ is the period of RT mini-frame generation, $C_{MF,j}$ is the maximum amount of data per period, and $T_{D\_MF,j}$ is the relative deadline used for the end-to-end scheduling. We assume $T_{D\_MF,j}$ is equally divided into two deadlines as well, for RT mini-frames from the source node to the switch and from the switch to the destination node, $T_{D1\_MF,j}$ and $T_{D2\_MF,j}$:

$$T_{D1\_MF,j} = T_{D2\_MF,j} = T_{D\_MF,j} / 2. \tag{8}$$

For easy understanding, $C_{MF,j}$ is expressed as the number of maximum sized mini-frames, while all the other parameters of the RTEF-channels, RTSM-channel and RTMF-channel are expressed as the number of maximum sized Ethernet frames.

4.3 Feasibility analysis

As mentioned above, we use EDF as the scheduling algorithm for both RTEF-channels and RTMF-channels, in both the switch and in the end-nodes. The feasibility tests are done on two levels: first the feasibility test of accepting the RTSM-channel (normally done at system start up), and then the feasibility test of accepting a RTMF-channel. Both of them are done in two steps, utilization constraint and workload constraint, each step being a test of its own.

In the feasibility test of the RTSM-channel or the additional of a new another RTEF-channel, the utilization constraint is checked first. According to basic EDF theory, the utilization for a physical link, *U*, should be less than or equal to the maximum value, 100%:

$$U = \left(\frac{C_{SM}}{T_{P\_SM}} + \sum_i \frac{C_i}{T_{Period,i}} \right) \leq 100\% . \tag{9}$$

In order to describe the second step of the feasibility test, we make the following definitions, by looking upon the real-time channel as a periodic task.

Hyperperiod. The *Hyperperiod* for a set of periodic tasks is defined as the length of time from when all tasks' periods start at the same time, until they start at the same time again.

BusyPeriod. A *BusyPeriod* is any interval of time in which a link is not idle.

Workload function. The *workload function* $h_{EF}(n,t)$ is defined as the sum of all the capacities of the tasks with absolute deadline less than or equal to *t*, running on the physical link *n*, where *t* is the number of time slots elapsed from the start of the hyperperiod. The duration of a time slot corresponds to the duration of a maximum-sized frame.

The RTSM-channel and several RTEF-channels might exist in the network at the same time. The workload functions $h_{EF1}(n,t)$ and $h_{EF2}(n,t)$ are calculated for each physical link from an end node to the switch, and from the switch to an end node, respectively, as follows:

$$h_{EF1}(n,t) = (1+\left\lfloor \frac{t-T_{D1\_SM}}{T_{P\_SM}} \right\rfloor)C_{SM} + \sum_i (1+\left\lfloor \frac{t-T_{D1,i}}{T_{Period,i}} \right\rfloor)C_i$$

$$h_{EF2}(n,t) = (1+\left\lfloor \frac{t-T_{D2\_SM}}{T_{P\_SM}} \right\rfloor)C_{SM} + \sum_i (1+\left\lfloor \frac{t-T_{D2,i}}{T_{Period,i}} \right\rfloor)C_i \quad . \tag{10}$$

According to the second constraint, the following expression must hold for all the values of t:

$$h_{EF1}(n,t), h_{EF2}(n,t) \le t \, . \tag{11}$$

It is shown in [16] how to reduce the time and memory complexity of this check. If Equation 11 holds in the first busy period of the hyperperiod in the supposed schedule, then it holds for all t. To only check the following range of t is therefore an improvement of the algorithm above:

$$1 \le t \le BusyPeriod(n) \tag{12}$$

where *BusyPeriod(n)* is the first BusyPeriod, starting at the beginning of the hyper-period on link n. Furthermore, all time slots are not required to be checked, but only the integers t:

$$t \in \left(\bigcup_{i=1}^{K} \{mT_{period,i} + T_{deadline,i} : m = 0,1,...\} \right) \cup \left(\bigcup \{m\,T_{P\_SM} + T_{D\_SM} : m = 0,1,...\} \right) \tag{13}$$

assuming that K denotes the number of RTEF-channels traversing the considered physical link n in the considered direction.

If the above utilization constraint and workload constraint are met, the RTSM-channel or the new RTEF-channel can be accepted. A similar feasibility test is done to check if a RTMF-channel can be accepted or not. However, the constraints for the RTMF-channel are not based on the physical link, but based on the RTSM-channel, which provides bandwidth for the RT mini-frame combined Ethernet traffic.

The utilization of the RTSM-channel, U_{SM} , must be less than the maximum capacity allocated for the RTSM-channel:

$$U_{SM} = \sum_j \frac{C_{MF,j}}{QT_{P\_MF,j}} \le \frac{C_{SM}}{T_{P\_SM}} \tag{14}$$

where Q is the maximum number of mini-frames per Ethernet frame.

In order to describe the workload test for the mini-frames, we calculate the workload functions $h_{MF1}(n,t)$ and $h_{MF2}(n,t)$ for the RTSM-channel from an end node to the switch and from the switch to an end node, respectively as follows:

$$h_{MF1}(n,t) = \sum_{j} (1 + \left\lfloor \frac{t - T_{D1\_MF,j}}{T_{P\_MF,j}} \right\rfloor) \frac{C_{MF,j}}{Q}$$
$$h_{MF2}(n,t) = \sum_{j} (1 + \left\lfloor \frac{t - T_{D2\_MF,j}}{T_{P\_MF,j}} \right\rfloor) \frac{C_{MF,j}}{Q} \quad (15)$$

by looking upon each RTMF-channel as a periodic task. Then the work load constraint says that the following expression must hold for all values of t:

$$h_{MF1}(n,t), h_{MF2}(n,t) \leq \frac{t}{T_{P\_SM}} C_{SM} \ . \quad (16)$$

The improvement of the algorithm can be done in the same way as described above for full-sized frames.

4.4 Delay-bound analysis

When an RTMF-channel has been established, the network guarantees to deliver each RT mini-frame message with a bounded delay:

$$T_{db\_MF,j} = T_{D\_MF,j} + T_{db\_SM} + T_{switch\_pro} \quad (17)$$

where T_{switch_pro} is the process latency experienced by a mini-frame in the switch, and T_{db_SM} is the end-to-end delay bound for the RTSM-channel. The worst case situation for considering the delay bound of the simulated medium is when all the RTMF-channels start at the same time, and the maximum allowed capacity for each RTMF-channel is used. T_{db_SM} is characterized by:

$$T_{db\_SM} = 2T_{link\_prop} + T_{D\_SM} + T_{swi\_acc} + T_{node\_acc} \quad (18)$$

where T_{link_prop} is the maximum propagation delay over a link between an end-node and the switch, T_{swi_acc} and T_{node_acc} are the worst-case latencies for an Ethernet frame with the earliest deadline to leave the source node and to leave the switch, respectively. Since we assume that we cannot interrupt the transmission of frames that have been stored on the NIC (Network Interface Card), even though they might have later deadlines than other frames, we assume:

$$T_{swi\_acc} = T_{node\_acc} = T_{frame} \quad (19)$$

where T_{frame} is the duration of a maximum-sized Ethernet frames. Then the delay bound for mini-frames is:

$$T_{db\_MF,\,j} = T_{D\_MF,j} + T_{D\_SM} + 2T_{link\_prop} + 2T_{frame} + T_{switch\_pro} \quad (20)$$

By Equation 20, the deadline of RTMF-channel can be derived, with the application-specified delay bound and the network specified parameters, for example, switch processing time, link propagation delay.

5. Simulation analysis

In the analysis, we let the RTSM-channel be allocated in advance. Each RTMF-channel is randomly generated with uniformly distributed source and destination nodes. We have simulated a network with a single 100 Mbit/s full-duplex Ethernet switch. Each RTMF-channel is characterized by three parameters: the period, the capacity and the deadline, expressed in the form RTMF{period, capacity, deadline}. Moreover, the parameters of the RTSM-channel are expressed in the similar way. In this paper, we assigned the same value to the period and the deadline, and the deadline is equally split into two parts as the deadline per hop.

For performance metric, we have used the number of accepted RTMF-channels and the utilization of the RTSM-channel. In each simulation, RTMF-channels are added one by one and checked whether accepted or not. A number of such simulations are run to get the average utilization and the average number of accepted channels at different traffic loads, i.e., total number of requested RTMF-channels. We have compared the number of accepted RTMF-channels and the utilization for different traffic and network characteristics.

In the first case (see Figure 3) we consider the relation between the average value of the number of accepted RTMF-channels and the total number of requested RTMF-channels. All the RTMF channels have the same parameters, {40, 2, 40}, where the period and the deadline of the RTMF-channels are expressed in number of maximum sized Ethernet Frames and the capacity in number of maximum sized mini-frames. The parameters for the RTSM-channel are {4, 1, 4}, which are expressed in maximum sized Ethernet frames. The number of nodes is set to 8. In this case, the utilization per RTMF-channel is approximately 0.25 % and the number of accepted RTMF-channels increases fast almost until the number of requested RTMF-channels arrives at 400. The number of accepted RTMF-channels then increases slowly when the traffic intensity per node is increased.

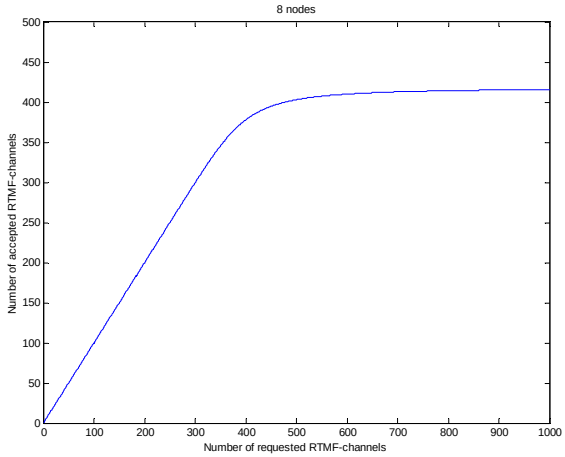


Fig. 3. Number of accepted RTMF-channels vs total number of requested RTMF-channels, where the parameters for RTSM are {4, 1, 4}, and the parameters for RTMF are {40, 2, 40}

If we replace the number of accepted RTMF-channels by the utilization of the RTSM-channel, we get the result shown in Figure 4 (average values after 5000 simulations). The figure shows a rather high utilization up to the theoretical limit of 50 % (since only half of the period duration is used for the transmission per hop, this is the theoretical limit according to the utilization constraint).

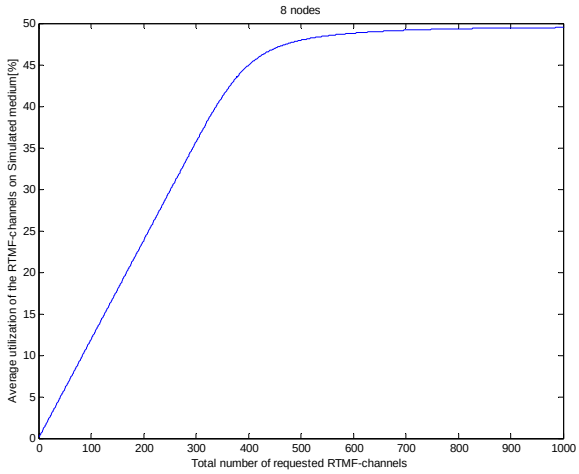


Fig. 4. Utilization of the RTSM-channels vs total number of requested RTMF-channels where the parameters for RTSM are {4, 1, 4}, and the parameters for RTMF are {40, 2, 40}

Figure 5 shows the effectiveness of the network when the parameters of the RTMF-channels are changed but the parameters of RTSM-channel are determined. The simulation shows the same ratio of accepted until the number of requested RTMF-channel passed 250. Then the number of accepted RTMF-channels is less if the deadline of the RTMF-channel is shorter, because shorter deadlines make it harder to meet the workload constraint for the RTMF-channels.

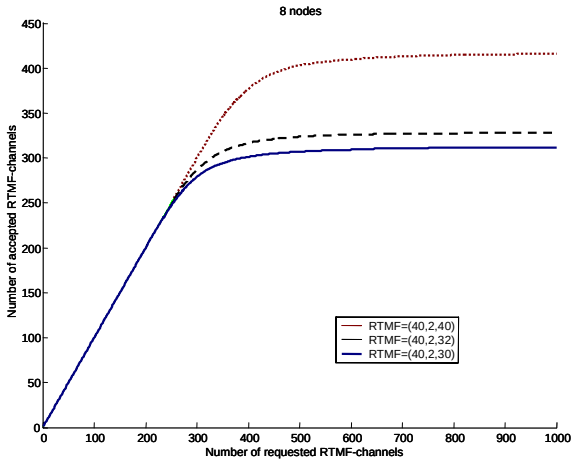


Fig. 5: Number of accepted RTMF-channels with different parameters of RTMF-channel vs. total number of requested RTMF-channels, where the parameters for RTSM are {4, 1, 4}

In contrary, if the parameters of the RTMF-channel are determined but the parameters of the RTSM-channels are changed, the results of the number of accepted RTMF-channels are given in Figure 6. As can be seen, the same accept ratio until the number of requested RTMF-channel passed 300. Then the highest number of accepted RTMF-channels occurs while the RTSM-channel is allocated with the parameters {4, 2, 4}, and the lowest number of RTMF-channels occurs while the RTSM-channel is allocated with the parameters {4, 1, 4}, because C_{SM} / T_{P_SM} is an essential factor of the utilization and workload constraints for the RTMF-channels. Higher value of C_{SM} / T_{P_SM} make it easier to accept a new RTMF-channel.

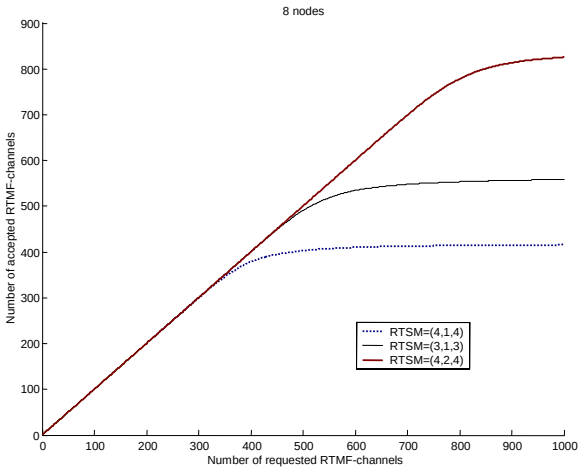


Fig. 6. Number of accepted RTMF-channels with different parameters of RTSM-channel vs. total number of requested RTMF-channels, where the parameters for RTMF are {40, 2, 40}

6. Conclusions

More and more emerging networks require efficiency and time constraint on group communication and other small-sized packet-oriented traffic. In this paper, we have investigated a switched Ethernet based network concept with guaranteed bit rate and worst-case delay for periodic real-time short-frame streams. The study covers both the performance and real-time support by the added software above the link layer in the nodes and the switch. By combining several mini-frames into an Ethernet frame, the performance can be significantly improved. The Ethernet switch operates over full-duplex links, and handles non-real-time traffic and normal real-time traffic as well as real-time short-frame traffic. The support of real-time mini-frames is made on two levels by allocating a special logical channel for the mini-frame combined Ethernet frames first, and then dynamically setting up real-time channels for mini-frames. The simulation analysis on Fast Ethernet has shown rather high average utilization up to the theoretical limit of 50% for traffic with the relative deadline equal to the period. We have shown, by example, that we can reach an improvement of the possible short-message rate of 66%.

This solution can be used for many network applications, like radar signal processing systems that require small messages with high bandwidth, low latency and bounded delay. A similar method can be used for analyzing other related network standards, for example, Infiniband [17] [18].

References

1. Jonsson, M.: Fiber-optic interconnection networks for signal processing applications. In: Proceedings of 4th International Workshop on Embedded HPC Systems and Applications (EHPC'99) held in conjunction with the 13th International Parallel Processing Symposium & 10th Symposium on Parallel and Distributed Processing, (IPPS/SPDP '99, San Juan, Puerto Rico (1999). Published in: Lecture Notes in Computer Science. Vol. 1586, Springer Verlag (1999) 1374-1385 ISBN 3-540-65831-9
2. Teitelbaum, K.: Crossbar tree networks for embedded signal processing applications. In: Proceedings of Massively Parallel processing using Optical Interconnections (MPPOI'98), Las Vegas, NV, USA (1998) 200-207
3. Bergenhem, C., Jonsson, M, Gördén, B., Åhlander, A.: Heterogeneous real-time services in high-performance system area networks - application demands and case study definitions. Technical Report IDE - 0254, School of Information Science, Computer and Electrical Engineering (IDE), Halmstad University (2002)
4. Weber, R., Santos, D., Bianchini. R., Amorim, C. L.: A survey of messaging software issues and systems for Myrinet-based cluster, Parallel and Distributed Computing Practices. In: Special Issue on High-Performance Computing on Clusters, Vol. 2, No. 2 (1999)
5. Mukheriee, S. S., Hill, M. D.: The impact of data transfer and buffering alternatives on network interface design. In: Proceedings of the Fourth International Symposium on High-Performance Computer Architecture (HPCA) (1998).
6 Rzymianowicz, L., Brüning, U., Kluge, J., Schulz, P., Waack, M.: ATOLL: A network on a chip. In: Proceedings of Cluster Computing Technical Session (CC-TEA) of the PDPTA'99 Conference, Las Vegas, NV, USA (1999)
7. Bergenhem, C., Jonsson, M.: Fibre-ribbon ring network with inherent support for earliest deadline first message scheduling. In: Proceedings of Workshop on Parallel and Distributed Real-Time Systems (WPDRTS'2002) in conjunction with International Parallel and Distributed Processing Symposium (IPDPS'02), Fort Lauderdale, FL, USA (2002)
8. Jonsson, M., Bergenhem, C., Olsson, J.: Fiber-ribbon ring network with services for parallel processing and distributed real-time systems. In: Proceedings of ISCA 12th International Conference on Parallel and Distributed Computing Systems (PDCS-99), Fort Lauderdale, FL, USA (1999) 94-101
9. Varadarajan, S., Chiueh, T.: Ethereal: a host-transparent real-time Fast Ethernet switch. In: Proceedings of 6th IEEE International Conference on Networks, Protocols (1998)
10. Varadarajan, S.: Experiences with Ethereal: a fault tolerant real-time Ethernet switch. In: Proceedings of 8th IEEE International Conference on Emerging Technologies and Factory Automation, (2001) 183-194
11. Hoang, H., Jonsson, M., Hagström, U., Kallerdahl, A., Switched real-time Ethernet with earliest deadline first scheduling − protocols and traffic handling. In: Proceedings of Workshop on Parallel and Distributed Real-Time Systems (WPDRTS'2002) in conjunction with International Parallel and Distributed Processing Symposium (IPDPS'02), Fort Lauderdale, FL, USA (2002)
12 Zhang, Q., Shin, K. G.: On the ability of Establishing real-time channels in point-to-point packet-switched networks. In: IEEE Transaction on Communications, Vol. 42 (1994)
13 Rexford, J., Hall, J., Shin, K.G.: A router architecture for real-time communication in multicomputer networks. In: IEEE Transaction on Computers, Vol. 47 (1998)
14. Fan, X., Jonsson, M., Hoang. H.: Efficient many-to-many real-time communication using an intelligent Ethernet switch. In: Proceedings Of International Symposium on Parallel Architectures, Algorithms, and Networks (I-SPAN'2004), Hong Kong, China (2004) 280-287

15. Liu, C. L., Layland, J. W., Scheduling algorithms for multiprogramming in hard real-time traffic environment, Journal of the Association for Computing Machinery, vol. 20 (1973).
16. Stankovic, J. A., Spuri, M., Ramamritham, K., Buttazzo, G.C., Deadline Scheduling for Real-Time Systems - EDF and Related Algorithms. Kluwer Academic Publishers (1998).
17 Infiniband Trade Association: Infiniband architecture specification, Release 1.0. In: http://www.infinibandta.org. (2000)
18 Pelissier, J.: Providing quality of service over Infiniband architecture fabric. In: Proceedings of Hot Interconnects (2000)