Halmstad University Post-Print

Real -Time Communication for Industrial Embedded Systems Using Switched Ethernet

Hoai Hoang

N.B.: When citing this work, cite the original article.

©2004 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

Hoang H. Real-time communication for industrial embedded systems using switched Ethernet. In: 18th International Parallel and Distributed Processing Symposium: Santa Fe, New Mexico, April 26-30, 2004 : proceedings. Los Alamitos, Calif.: IEEE Computer Society; 2004. p. 127-130.

DOI: <http://dx.doi.org/10.1109/IPDPS.2004.1303092>

Copyright: IEEE

Post-Print available at: Halmstad University DiVA
<http://urn.kb.se/resolve?urn=urn:nbn:se:hh:diva-393>

Real-Time Communication for Industrial Embedded Systems Using Switched Ethernet

Hoai Hoang.

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

Abstract

This paper gives an overview of research on real-time communication over systems area networks (SANs). Our project is focused on developing real-time services for switched Ethernet networks, keeping the Ethernet and TCP/IP Internet communication standards [1]. Switched Ethernet is extended to allow for periodic real-time traffic using earliest deadline first (EDF) scheduling [1] [2]. We have shown the advantage of supporting special communication patterns apparent in industrial and embedded systems [3]. We have made some implementation that will demonstrate the support of both guaranteed real-time communication and efficient services for parallel and distributed computer systems by an Ethernet switch with added software.

1 Introduction

Industry has been very successful in the design and manufacture of complex embedded systems. Examples include modern vehicles (control, information, entertainment etc), radar systems, telecommunication systems (radio base stations, multi-rack switches and routers etc), wireless communication systems, and the automation industry. These embedded systems are becoming more and more complex, distributed, interconnected and based on distributed computing to a larger extent.

As a result, increased demands are placed on the system, for example, in capacity ("bandwidth"), heterogeneity, and quality of service (QoS). At the same time, there is the ongoing need to keep overall costs down. Together, these issues demand the use of standard networks (low cost), but with added intelligence and support for the different services required in embedded systems (high performance and functionality; low engineering costs for development and upgrading). In many cases, the systems's functions must be enhanced with support for real-time communication with various demands, and/or with time-deterministic behavior. Support for such functional enhancements is very limited in today's networks.

In this project, our goal is to find communication solutions that meet the increasing real-time demands that guarantee performances service heterogeneity and service levels. We want to develop networks architectures, protocols, and methods that can be used in standardized networks.

2 Problem statement and project overview

2.1 Problem statement

The research focuses on real-time communication over non-real-time LAN technologies that have so far been concentrated on multimedia and similar applications. However, there is also large need for research efforts in the field of industrial systems. Interest has been shown to adapt the current high speed Ethernet standard for use in industrial applications, as this would simplify the networking aspects of companies, as well as cut down on costs from installation and maintenance.

Ethernet is of the type *multiple-access network* but can easily be implemented as a pure *switched network* by adding more switches in an intelligent manner. Our focus mainly will be on switched networks, which is reasonable as the prices of Ethernet switches have steadily dropped as a result of the widespread use of the hardware.

The main research question is how to use switched networks to take full opportunity of the possibilities of, e.g., increased analyzability, heterogeneity, performance, and modularity must, however, be further investigated. Typical of other currently proposed solutions are time constraints that are only indirectly treated, e.g., by the use of priority mechanisms. Even if some proposed solutions also treat time constraints, industrial systems typically demand shorter response times and/or cycle times. More recent results on using LAN standards in industrial systems exist, but there is still much to do to support really time-critical systems.

Research must also be concentrated on obtaining support for much more diverse real-time requirements. An important research issue is how to support traffic with diverse real-time requirements at the same time, i.e., and heterogeneous real-time services, together with other time-deterministic communication services for embedded systems. Significant work on quality of service in packet-switched networks has already been carried out, but the focus has been on both real-time and non real-time services to support embedded-systems applications. A main task of this project will be to develop and analyze how methods to support heterogeneous services with real-time characteristics can be implemented in switches.

Focus is on industrial and embedded systems. Application examples include industrial automation, Computer Integrated Manufacturing (CIM), radar signal

processing systems, airplanes, process control, and telecommunication equipment. Other applications that might also be of interest are home networks and broadband access networks, where a mix of real-time and non-real-time traffic is expected.

2.2 Project overview

The project is an extension of our previous work in switched real-time Ethernet in industrial and embedded systems. A simple, star topology network, with a single-switch and end-nodes connected to the switch, has been proposed. Over this simple topology logical circuits are established (so-called real time channels) over which real time traffic is guaranteed time constraints. In this scenario the messages sent from source node to the destination node therefore traverse two links (Figure 1), and we need to provide guarantees for the time to deliver over only two links. We approach this problem by dividing the end-to-end deadline into two, one for the source to the switch, and one from the switch to the destination. The deadline can be partitioned in a number of ways. The method chosen effect the system. We have introduced a deadline partitioning scheme (DPS), which is concerned with proposing distribution of end-to-end deadline along determined paths for messages. This improves performance for, e.g., master slave communication, which is common in industrial applications.

Figure 1: Network configuration, master-slaver traffic pattern with RT channels

The results, and indeed the method in its current form, do not refer more than one switch nor to a mixed topology. The problem is how to form methods for a more complicated network topology and more nodes in the network. We are foremost concerned with adding nodes to the network via more switches that can in turn be connected in various configurations.

3 Results so far and future work

3.1 Network conception

We assume a network with a switched Ethernet topology, in which both the switches and the end-nodes have a RT layer added to support guarantees for real-

time traffic. Each node is connected to other nodes via the switches. Before any real time traffic can be allowed real-time channels must be established. A RT channel is static path across the network, where sufficient bandwidth has been reserved. The test phase thus determines if a given channel is feasible in the respect that all messages will meet their time constraints.

The network supports dynamic addition of RT channels. Regular non-real-time traffic is supported also, by letting any Real time traffic take precedence. Full-duplex switched Ethernet is assumed for the network. End-nodes have the capability of controlling outgoing traffic from the nodes using the Earliest Deadline First (EDF) algorithm. The switch has the same capability. The proposed network is well suited for, e.g., master-slave communication, in which the end-nodes are divided into master nodes and slave nodes. Figure 1 illustrates this in the case where only one switch is present. Time is measured in discrete time units. One time unit is the time it takes to transmit one maximum sized frame.

When going from the easier case of only one switch to more challenging topologies we make some assumptions:

For the interconnections and topology we assume the following:

- Full-duplex links
- Identical transfer rates
- No shared medium
- No media propagation delay
- No noise

For the nodes we assume:

- Infinite buffer space
- Infinite processing capabilities (all computation finish immediately)
- No queue handling times

These assumptions impose some very restrictive (indeed impossible) properties on our system. Some of these will have to be addressed and ultimately overcome. However considering a system where these assumptions are valid is a good starting point.

Another issue that needed not be addressed with the one switch case was that of routing. The real-time channels are static paths once established, but during establishment one of the many paths available must be chosen, as well as found.

3.2 Results this far

At this stage we are investigating the usage of a modified version of the proposal in [4]. Like the simpler case with one switch we want to divide the real-time channels into smaller parts, for tests on individual links. In [4] a different scheduling algorithm is considered (EDD as in contrast to us using EDF).

We are at this stage not considering the routing aspects of the problem. Instead, for comparison and simulation, we are assuming optimal routing. That is to say, real time channels will always be added along the route that minimizes the worst case end-to-end delay. Routing is a substance of future work.

Now, consider a multihop network with the assumptions of the previous section. Several switches are in some given topology. When a channel is to be established it can be done so by special purpose, non real-time traffic. Envision that at a channel request, a special request frame is transmitted from the sending node to the receiving node, gathering information as it goes. Alternatively, for routing reasons that we will not consider further, we might want to send more than one request frame. The issue of routing must be resolved at this stage, finding an appropriate route for the channel that thereafter remains static for the channel duration. Also, the issue of feasibility must be resolved, i.e. the system must in some distributed way determine if the channel is possible to give the guarantees asked of it.

For each channel, with end-to-end deadline D , it will traverse its optimal route of N links if accepted. On each link traversed the channel will have a local node-to-node deadline associated, $d_1, d_2, \dots, d_N$. At the time of the request frame's arrival to the node sending on link j , we can calculate the minimum feasible deadline ($\min D_j$). $\min D_j$ is the lowest possible value for d_j , for which messages of the channel are guaranteed to be transmitted over link j in time. The fact that the route was optimal is reflected in that the sum

$$\sum_{j=1}^N \min D_j$$

is lower than that of all other possible routes. The feasibility test of the channel comes to:

$$\sum_{j=1}^N \min D_j \leq D$$

or

$$0 \leq D - \sum_{j=1}^N \min D_j$$

where

$$slack = D - \sum \min D_j$$

If the channel is feasible for $d_j = \min D_j$, it is evident that is also feasible for

$$d_j = \min D_j + \frac{1}{N} \times slack \quad (1)$$

the assignment in (1) is the same as the one found in [4]. If we assume that each link j has a positive value (a weight) associated with it, w_j , then another scheme of assigning local deadlines can be devised:

$$d_j = \min D_j + \frac{w_j}{\sum w_i} \times slack \quad (2)$$

where $\sum w_i$ is the sum of all weights of the links the channel traverse, one for each link. It is easily proven that with assignment (2), the channel will be feasible. We have formulated this second distribution formula in an attempt to counteract the bad effects of bottlenecks. If the system somehow makes the weights appropriately reflect the "level of bottleneckness" of the link, then the system will more generously distribute the slack of the channels to the bottlenecks.

How should the "level of bottleneckness" be measured? If it was a measure of regular traffic then the utilization of the link could be used. However, that would not appropriately weigh in the effect of time constraints. Surely, if a number of channels are traversing a link, and these channels have a generous end-to-end deadline, then the "level of bottleneckness" should be lower, than if the deadlines were very narrow.

3.3 Future work

In our research so far, we have presented a switched Ethernet based network concept of a single switch supporting real-time communication with guaranteed bit rate and worst-case delay for periodic traffic. We are interested in making it work on a larger scale, with more challenging topologies. Because we want the proposed system to be able to support as many channels as possible under traffic patterns such as Master-Slave communication, we are concerned with devising a system that can have as many channels, with as much demanding traffic as possible.

We have gotten optimistic results so far, from simulation with the approach described when using asymmetric slack distribution (2). Early results indicate that by distributing slack asymmetrically when adding channels we get better results, i.e., we can have more RT traffic with the asymmetric distribution of slack, than with the symmetric one.

Future work includes further investigating the use of more complex network topologies, i.e., networks consisting of many interconnected switches and links having shared medium. Alternative communication models and scheduling algorithms could be explored as well. Routing issues and problems must also be solved.

Regarding the assumptions in section 2.3, some issues might be taken into the analysis; other might be left out, in which case fault tolerance issues should be addressed.

4 References

- [1] H. Hoang, M. Jonsson, A. Kallerdahl, and U. Hagström, "Switched real-time ethernet with earliest deadline first scheduling - protocols and traffic handling," *accepted for publication in Parallel and Distributed Computing Practices*.
- [2] H. Hoang, M. Jonsson, A. Larsson, R. Olsson, and C. Bergenheim, "Deadline first scheduling in switched real-time Ethernet - deadline partitioning issues and software

implementation experiments," *Proc. 1st International Workshop on Real-Time LANs in the Internet Age (RTLLA'2002) in conjunction with ECRTS'2002*, Vienna, Austria, June 18, 2002.

[3] H. Hoang, M. Jonsson, "Switched Real-Time Ethernet in Industrial Applications- Deadline Partitioning Scheme", *In Proc. of the 9th Asian Pacific Conference on Communication*, Penang, Malaysia, Sept. 2003.

[4] D. Ferrari and D. C. 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, Apr.