Ethernet-based Real-Time Control Networks for Manufacturing Automation Systems

Seok-Kyu Kweon and Kang G. Shin
Department of Electrical Engineering and Computer Science
The University of Michigan, Ann Arbor, MI 48109-2122

ABSTRACT

This paper presents a methodology to provide soft real-time guarantees over Ethernet. To resolve unpredictable delay characteristics of Ethernet, we designed, implemented, and evaluated adaptive traffic smoothing. Specifically, a traffic smoother is installed between the TCP/IP layer and the Ethernet MAC layer, and works as an interface between them. The traffic smoother first gives real-time (RT) packets priority over non-RT ones in order to eliminate contention within the local node. Second, it smooths a non-RT stream so as to reduce collision with RT packets from the other nodes. This traffic smoothing can dramatically decrease the packet-collision ratio on the network. The traffic smoother, installed at each node, regulates the node's outgoing non-RT stream to achieve a certain traffic-generation rate. In order to provide a reasonable non-RT throughput, the traffic-generation rate is allowed to adapt itself to the underlying network load condition. Our implementation of the traffic smoother requires only a minimal change in the OS kernel *without* any modification to the current standard of Ethernet MAC protocol or the TCP or UDP/IP stack.

KEYWORDS — CSMA/CD, Ethernet, real-time communication, traffic smoothing.

1 Introduction

Manufacturing automation industry has been pursuing the use of commercial off-the-shelf (COTS) network products for transporting control messages between PLCs (Programmable Logic Controllers). Traditionally, proprietary networks such as Allen-Bradley's RIO (Remote Input/Output) Network have been used in factory automation to meet the control applications' stringent real-time requirements and deal with harsh working environments. However, the low price and the proven stability of COTS networks have made them attractive for automated manufacturing. Although various high-speed networks like ATM and FDDI are available, Ethernet has been drawing significant interests because of its extremely low price, maturity, and stability proven through its wide deployment and acceptance. Despite its popularity and low-cost, Ethernet has a serious drawback when carrying real-time control messages. In an Ethernet LAN, packets transmitted from different nodes may collide with each other. The medium access control (MAC) protocol of Ethernet, CSMA/CD (Carrier Sense Multiple Access with Collision Detection), allows such collisions. These potential collisions make it impossible to guarantee predictable delays in delivering packets to the local nodes.

Lawrence Livermore Laboratories subcontracted through Louisiana Center for Manufacturing Sciences under TIME subcontract Phase III.

In [1], we showed the feasibility of building a real-time control network using Ethernet by installing a traffic smoother at each local node. A traffic smoother regulates the intrinsically bursty packet stream relayed from the TCP/IP layer, making the packet stream as smooth as possible in order to reduce the chance of packet collisions. Traffic smoothing enables us to to provide a statistical bound on the deadline-miss ratio over Ethernet by keeping the network utilization under a certain limit, called the *network-wide input limit*. To keep network utilization under the network-wide input limit, we assigned a portion of the network-wide input limit to each local node, and made each local node limit its packet generation rate below its assigned portion. We called each node's portion of the network-wide input limit the *station input limit* and installed a traffic smoother at the node to enforce it. This traffic smoothing approach which we call *fixed-rate traffic smoothing*, however, is inflexible and hence unscalable. In that approach the network-wide input limit is fixed once we are given a packet deadline and a tolerable packet-loss (or deadline-miss) ratio. So, the station input limits must be reduced as the number of local nodes increases within the same LAN. The smaller the station input limit gets, the smaller throughput provided to non-RT traffic. (Note that real-time traffic is not affected, as only non-RT traffic is smoothed [1].) Non-RT packets may experience very large delays when a very small station input limit is assigned to a local node, as discussed in the next section. In this paper, we propose an *adaptive* traffic smoothing approach to overcome the scalability problem of the approach in [1]. By allowing each local node to vary its maximum traffic-generation rate depending on the current network load, the proposed approach improves its scalability significantly. Apart from this modification, the proposed approach shares the same traffic-smoothing mechanism with the approach in [1]. The traffic smoother is implemented as an interface between the TCP/IP layer and the Ethernet MAC layer. This implementation minimizes the modification in the current standard network protocol. We implemented the adaptive traffic smoother on the Linux OS, built a testbed, and conducted an experimental study. Through this experimental study, we show that the adaptive approach provides a much higher throughput for non-RT packets while still providing good delay characteristics for RT packets.

2 Adaptive-Rate Traffic Smoothing

In this paper, we assume that control stations employ an event-driven approach in generating RT control messages. In this approach, each control station generates at most one maximum-sized (1500 bytes) IP datagram once every several seconds (or several hundred milliseconds), and hence, its rate of generating real-time control messages is very low relative to the Ethernet link capacity. Moreover, control messages arrive pseudo-periodically due to the characteristics of the underlying control system.

Concurrently with RT control messages, bursts of non-RT traffic are generated on an irregular basis by controllers and the central server, mainly for the purpose of monitoring production status and downloading programs or new setup parameters. Because of its burstiness, the arrival rate of non-RT traffic can be quite high during the transmission duration even if its long-term average traffic arrival rate is low. For example, when only a single station transmits a large burst of non-RT traffic (e.g., a file transfer) over an Ethernet LAN, the traffic arrival rate can reach up to 8–9 Mbps. Such temporarily high network utilization makes it very difficult to provide bounded delivery delays for the other stations' RT messages when both RT and non-RT messages are concurrently transported over the same Ethernet LAN. During the transmission of a large burst of non-RT traffic from another station (node), RT messages may experience a large delay because of collisions and possibly due to the "packet starvation" [2].

In order to resolve this problem, we employ a traffic smoothing approach as in [1]. We install the traffic smoother between the TCP/IP layer and the Ethernet MAC (Medium Access Control) layer, in order to minimize the changes in the current standard protocol stack while achieving the good smoothing effect. When a burst of non-RT messages arrive from the TCP/IP layer, the traffic smoother spreads them out by enforcing a minimum packet inter-arrival time at the Ethernet MAC layer to meet the station input limit. More specifically, the traffic smoother regulates the packet stream using a credit bucket, which is the same as the well-known leaky-bucket regulator [3]. The credit buffer has two parameters: credit bucket depth (CBD) and refresh period (RP). CBD limits the maximum number of credits that can be stored in the credit bucket. Up to CBD credits are added to the bucket every RP seconds. If the number of credits exceeds CBD, overflow credits are discarded. When a packet (IP datagram) arrives from the IP layer, if there is at least one credit in the bucket, the traffic smoother forwards it to the Ethernet NIC (Network Interface Card) and removes as many credits as the size of the packet (in bytes). When the number of available credits is smaller than the packet size, credits are allowed to be "borrowed." So, the balance of credits can be negative. If there are no credits in the credit bucket, the packet is held in the buffer until one or more credits become available. By changing RP and CBD, one can control the burstiness of a packet stream while keeping the same average throughput guarantee. In particular, we change RP while keeping CBD to a constant in order to achieve finer-granularity smoothing

In this approach, within a local node, RT packets are given priority over non-RT packets, and only non-RT packets are delayed to keep the station traffic-arrival rate (which includes both RT and non-RT traffic) under the station input limit. That is, transmission of extra RT packets causes non-RT packets to experience additional delays. RT traffic is assumed to arrive pseudo-periodically and thus, is already smooth as discussed earlier.

Unlike a fixed-rate traffic smoother, however, our new traffic smoother, called an *adaptive-rate traffic smoother*, changes the station input limit at each local node depending on the current network traffic arrival rate. That is, if network utilization by non-RT traffic is low, those nodes generating non-RT traffic are allowed to increase their station input limits subject to the condition that the overall network utilization does not cause RT packets to experience delays larger than those in the fixed-rate traffic smoothing approach. Likewise, as network utilization by non-RT traffic gets higher, those nodes generating non-RT traffic lower their station input limits.

In order to implement an adaptive-rate traffic smoother which meets the delay requirement of RT packets while providing improved average throughput for non-RT packets, we must resolve the following two problems: (1) how to detect a change in network utilization and (2) how to adapt to the detected change. An efficient detection mechanism is essential for the adaptation to be fast enough to meet the delay requirement of RT packets. However, since unlike ATM or FDDI, the CSMA/CD protocol is not a reservation-based medium access control scheme, direct information on the current network utilization is unavailable to local nodes. Therefore, each local node must depend on an indirect method of determining network utilization such as detecting packet collisions at its NIC or measuring the buffer-clearing rate at its Ethernet device driver. Or, each local node may use the promiscuous mode to measure the network utilization for a recent period of time. We have chosen the first option for its good responsiveness. In particular, if the traffic smoother is set to vacate the credit buffer immediately upon detection of a collision, transmission of non-RT packets is suspended, except for those packets already in NICs. This increases the chance to deliver the RT packets generated from other nodes sooner, as they do not suffer the "packet starvation" [2] caused by the burst of non-RT packets generated from this node. For this reason, we use packet collision as a trigger to decrease throughout as well as to deplete the current credits.

Next, let's consider the adaptation mechanism. As we argued above, we vary RP while keeping CBD constant to achieve finer-granularity smoothing. Especially, by setting CBD to the Ethernet MTU (Maximum Transfer Unit) (i.e., 1500 bytes), one can set the maximum amount of traffic that can be transmitted up to 2999 bytes. In this approach, one can increase the station input limit by decreasing RP, and vice versa. We use a very simple adaptation mechanism called *Harmonic-Increase and Multiplicative-Decrease Adaptation* (HIMD). HIMD is similar to the slow-start increase and multiplicative-decrease algorithm [4] in decreasing the throughput but *differs* in increasing the throughput. HIMD works as follows. First, HIMD periodically increases the station input limit by decreasing RP periodically in the absence of packet collisions. The size of each decrement is fixed at a constant, and thus, the station input limit is harmonically incremented. This harmonic increment is conservative but easy to implement. When a packet collision is detected, the traffic smoother immediately depletes the current credits, delays the transfer of the non-RT packet, and doubles RP. By choosing an appropriate size of decrement for RP, one can adapt the station input limit very fast.

3 Experimental Evaluation

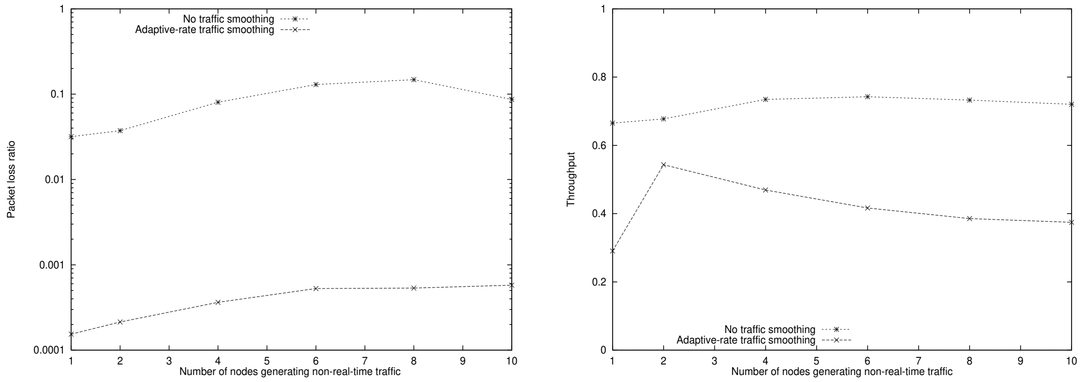
In order to investigate the effectiveness of the proposed approach, we have built a testbed and conducted an experimental evaluation on it. We installed the adaptive-rate traffic smoother at all the local nodes, and measured the delay characteristics of RT packets while measuring the throughput characteristics of non-RT packets. In addition, we conducted the same experiment without employing any traffic smoothing mechanism for the purpose of comparison.

The testbed consists of two 300 MHz Intel Pentium II PCs, five 75 MHz Pentium laptop computers, and four 486 DX/4 laptop computers, and they are connected through a 10BASE-T Ethernet LAN. The collision domain diameter is 10 m. We configure the local nodes as PC-1 — PC-10 and a monitoring station. One 300 MHz Intel Pentium II PC works as the monitoring station, and since our target application is automated factory networking, the rest of the PCs simulate PLCs. We use TCP sockets for transmitting RT control messages as well as non-RT messages. The PCs exchange real-time control information with RT packets. More specifically, PC-1 sends a 100 byte long RT control information which is contained in a high-priority IP datagram to PC-2. Then, PC-2 echos back to PC-1 with a high-priority IP datagram of the same size. Likewise, PC-n and PC-$(n+1)$ exchange RT control information of the same size where $n = 1, \ldots, 9$. PC-10 sends a RT control message to PC-1, and PC-1 echos back to PC-10. We made the inter-arrival time of real-time control messages at each simulated PLC follow an exponential distribution, and set the average message inter-arrival time to 0.3 sec. Since we must count both RT control and echo messages, the network load due to RT messages is $(200 \cdot 8 \cdot 10/0.3)$ bps, i.e., 53.3 kbps. The traffic-generation rate was chosen to reflect the low traffic condition observed in most automated manufacturing facilities.

In addition to RT messages, PCs generate non-RT messages when the monitoring station requests them to send their status information. We call such PCs *activated.* The size of non-RT traffic generated by an application running on an activated PC is 1 Mbytes, and it is transmitted as a sequence of low-priority IP datagrams. This results in a high instantaneous traffic-generation rate (i.e., a burst of non-RT packets) especially at the TCP/IP layer. Once activated, PCs were set to generate non-RT bursts in succession. That is, once they had finished the transmission of a non-RT burst, they start transmission of the next burst immediately. Therefore, the network can be overloaded even with a single activated PC. In reality, however, the maximum achievable network utilization is about 0.75 because of the congestion-avoidance mechanism of the TCP flow control and the Ethernet collision-resolution mechanism.

To investigate the effectiveness of the adaptive-rate traffic smoother, we measured the roundtrip delay of every RT control message and the time to transmit each non-RT burst while transporting both types of traffic over the Ethernet and varying the number of activated PCs. From these measurements, we calculated the deadline-miss ratio of RT packets and the average time to transmit a 1 Mbyte-long non-RT burst. We set the roundtrip deadline of RT messages to 129.6 msec. Since a real-time message is considered lost if its deadline is missed, we treated the deadline-miss ratio as the message-loss ratio.

Figures 1 shows the experimental results. Figure 1(a) shows the deadline-miss ratios of RT messages for different numbers of activated PCs when no traffic smoothing was enforced and when adaptive-rate traffic smoothing was enforced. Figure 1(b) shows the throughput provided to all the activated PCs for transmitting non-RT bursts. This was derived from the number of activated PCs and the average transmission time for transmitting a single burst, considering that the throughput provided to an activated PC is given by the burst size divided by the transmission time. When no traffic smoothing was enforced, the deadline-miss ratios of RT messages were extremely high, i.e., in the range of 10^{-1} as shown in Figure 1(a), although the throughput provided for non-RT traffic reached up to 0.74 as shown in Figure 1(b). Thanks to the flow controls mention above, we could not overload the network. On the other hand, when the adaptive-rate traffic smoothing was enforced on every activated PC, the throughput for transmitting non-RT bursts was reduced, approximately by half, but the RT message deadline-miss ratios dropped dramatically. They ranged from 1.54×10^{-4} to 5.78×10^{-4}, and were much smaller than those achieved in the case of no traffic smoothing. In this environment, the transmission capability of a TCP socket of an activated PC was restricted not only by the TCP flow control and Ethernet collision-resolution mechanism, but also by the adaptation mechanism of the traffic smoother. This explains the lower throughput achieved by the adaptive-rate traffic smoothing. These experimental results indicate that we can build *two virtual networks* — a soft real-time control network and an Ethernet LAN with 5 Mbps transmission capability — using a single Ethernet LAN if the adaptive-rate traffic smoothing is employed.



(a) (b)

Figure 1: Experiment Results: (a) RT message-loss ratio in the case of unlimited traffic-generation rate (b) Throughput of non-RT traffic

4 Conclusion

In this paper, we developed a methodology for providing a soft real-time communication service over an Ethernet LAN which transports both real-time and non-real-time packets. Using traffic smoothing, our methodology can dramatically reduce the packet-collision ratio on the network. We implemented the proposed traffic smoother in the Linux OS, and conducted an experimental study on a testbed. The study showed that the message deadline-miss ratio can be kept well under 10^{-3} for *any* non-real-time traffic arrival rate if all the local nodes are equipped with the proposed traffic smoothers. Moreover, the study showed that the proposed traffic smoother can provide a reasonable average throughput to non-real-time traffic while still yielding a remarkably low real-time message deadline-miss ratio.

References

[1] S.-K. Kweon, K. G. Shin, and Q. Zheng, "Statistical real-time communication over ethernet for manufacturing automation systems," in *IEEE Real-Time Technology and Applications Symposium*, June 1999.

[2] B. Whetten, S. Steinberg, and D. Ferrari, "The packet starvation effect in CSMA/CD LANs and a solution," in *Proc. of 19th Conference on Local Computer Networks*, pp. 206–217, 1994.

[3] R. L. Cruz, "A calculus for network delay, part I: network elements in isolation," *IEEE Trans. on Information Theory*, vol. 37, pp. 114–131, Jan. 1991.

[4] D. E. Comer, *Internetworking with TCP/IP Volume I, Principles, Protocols, and Architecture*. Englewood Cliffs, New Jersey: Prentice-Hall International, third ed., 1995.