Performance of the IEEE 802.2 Type-2 Logical Link Protocol with Selective Retransmission

Ernst W. Biersack

Abstract—**This paper investigates the effects on performance of adding the selective retransmission feature to the IEEE 802.2 type-2 logical link protocol. Simulation results indicate that selective retransmission significantly improves the performance in case of overload and performs as well as another enhancement that was suggested earlier by Bux and Grillo.**

I. Introduction

THE IEEE 802.2 type-2 logical link protocol, or LLC protocol for short, is a connection-oriented link-layer protocol that provides a reliable data transfer across an unreliable transmission medium. The LLC protocol was standardized by the IEEE [1].

Bus and Grillo [2] studied the performance of the LLC protocol in an environment of three token rings interconnected with a backbone token ring via bridges. Their study revealed a serious degradation of throughput when the network was overloaded.[1] The degradation was because of congestion at the bridges when packets arrive at a higher rate than the rate at which they can be forwarded. Bux and Grillo suggested enhancing the LLC protocol with a dynamic flow control scheme. This enhancement significantly improved the performance under overload and was later added to the IEEE 802.2 standard as an option.

We study another modification to the original LLC protocol and its impact on performance. We show that adding selective retransmission to the LLC protocol yields performance equivalent to the enhancement of Bux and Grillo.

II. LLC Protocol

We assume that the reader is familiar with the basic operation the LLC type 2 protocol, referred to as ORIG. A complete specification of ORIG is contained in the IEEE 802.2 standard [1].

We will focus our attention on the behavior of the different variants of the LLC protocol during overload. Before we can present our results we need to introduce some notation and review the error recovery mechanism of ORIG. The LLC protocol is a connection-oriented protocol with error control and flow control. Each data packet, referred to as *I-frame,* carries a sequence number. The sender has a state variable $V(S)$ that contains the sequence number for the next new I-frame to be sent and the receiver has a state variable $V(R)$ that contains the sequence number of the next I-frame expected to arrive. The variable w at the sender contains the window size that limits the maximum number of outstanding I-frames to w. The receiver operates as follows. When an I-frame with sequence number $N(S)$ arrives, the receiver checks if $N(S) = V(R)$. If true, the receiver accepts the I-frame, increments $V(R)$, and generates a receive ready-frame $RR(V(R))$[2] to acknowledge the receipt. If false, the I-frame is discarded and a negative acknowledgment $REJ(V(R))$ is generated asking the sender to retransmit all I-frames with sequence numbers equal or larger than $V(R)$. The sender and receiver use timers to prevent deadlock due to the loss of a RR- or REJ-frame. When an I-frame is not acknowledged, and *ACK-timer* expires and the sender polls the receiver for an acknowledgment. When an I-frame that the receiver asked to be retransmitted does not arrive, the receiver will time out and send another REJ-frame.

Bux and Grillo used simulation to study the behavior of ORIG and suggested a dynamic flow control algorithm to improve the performance of ORIG. Whenever the sender receives a REJ-frame it reduces its current window size dw to 1. Subsequently, if dw is smaller that w, the sender increases the window size from dw to $dw + 1$ after the reception of n RR-frames where n is a control parameter that must be chosen appropriately. The LLC protocol with the dynamic window adjustment proposed by Bux and Grillo will be referred to as BG.

For comparison with BG we also investigate the performance of a slightly different window adjustment policy proposed by R. Jain [3]. If the window size has been reduced to 1, the window size is increased from dw to $dw + 1$ after the reception of dw RR-frames. Jain's strategy increases the window size faster while the values of dw are small. We will refer to the LLC protocol with Jain's version for the window adjustment as JAIN.

The LLC protocol bears strong resemblance to data link protocols such as HDLC and LAPD [4], which are used in public data networks. While HDLC has a SREJ-frame to request the retransmission of a single I-frame, LAPD and the LLC protocol do not. Selective retransmission helps save bandwidth at the expense of a more complex receiver. The receiver stores

Paper approved by the Editor for Communication Protocols of the IEEE Communications Society. Manuscript received October 15, 1988; revised March 19, 1991. This paper was presented in part in the Proceedings 13th Conference on Local Computer Networks, Minneapolis, MN, October 1988.

The author was with Bellcore, Morristown, NJ 07962. He is now with Institut EURECOM, 06560 Valbonne, France.

IEEE Log Number 9207344.

[1] The backbone token ring has a bandwidth of 4 Mb/s. When the total data rate offered by the end nodes is 4.0 Mb/s or higher, we consider the network to be overloaded.

[2] The acknowledgment contains the sequence number of the next expected I-frame and is cumulative because it acknowledges all I-frames with sequence numbers smaller than $V(R)$.

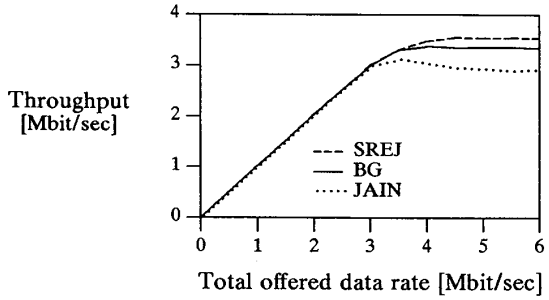

Fig. 1. Throughput versus total offered data rate.

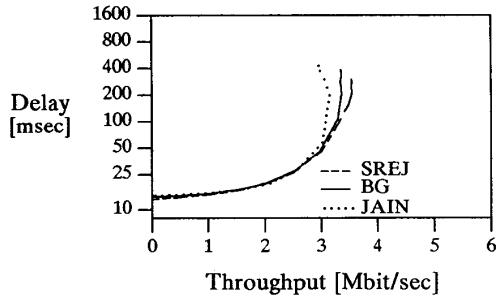

Fig. 2. Mean end-to-end delay versus throughput.

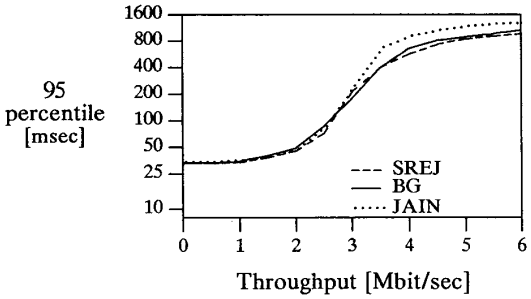

Fig. 3. 95 percentile of the end-to-end delay versus throughput.

I-frames with sequence numbers in $[V(R), V(R) + w - 1]$ that are out-of-sequence. Therefore, only I-frames that are corrupted or lost need to be retransmitted. We will refer to the LLC protocol with selective retransmission as SREJ. The savings in bandwidth due to selective retransmission become more pronounced with increasing window sizes. If a single loss occurs during the transmission of w I-frames, w I-frames will be out-of-sequence. In the case of ORIG, when the network was overloaded we observed that more than 70% of all I-frames received were out-of-sequence and discarded. A drawback of SREJ is that a station using SREJ can not communicate with a station using ORIG, while stations using BG or JAIN can interoperate with a station using ORIG.

III. Results

We replicated the simulation model of Bux and Grillo using their assumptions about the frame length distribution, the timeout values, and the processing times of the various frames. We selected their internet configuration with a 4 Mb/s backbone, 4 Kbytes buffer sizes at the bridges, and an internet traffic only scenario, with 12 stations on every ring sending to stations on another ring. The simulations were run long enough to ensure that the actual values for the throughput and the delay are with a probability of 0.95 within $\pm 2.5\%$ of the value obtained.

The results obtained for this scenario indicate that SREJ under overload has a slightly better user-level performance than BG. The maximum achievable throughput for SREJ is 3.55 Mb/s as compared to the 3.35 Mb/s for BG; see Fig. 1. The delay performance of SREJ, with respect to the average delay and the 95th percentile of the delay, is slightly better than for BG, see Figs. 2 and 3. JAIN does not perform as well as either BG or SREJ. The difference in performance between JAIN and BG is quite striking and illustrates the impact of the window adjustment policy on the performance of the dynamic flow control schemes. The lower performance of JAIN is due to the fact that after a loss JAIN opens the window too fast, which causes a high number of I-frames to be dropped by the bridges. When we ran the same simulations with different values for the buffer size, window size, and the timeout, the performance of BG, SREJ, and JAIN relative to each other stayed the same.

At first glance, the excellent user-level performance of SREJ may come as a surprise because SREJ does not take any explicit steps to throttle the sender in case of overload. However, a closer look at the operation of SREJ under overload reveals that there are two effects that stop the sender when an I-frame is lost.

- When the I-frame $N(S)$ is lost, the sender-window is not advanced beyond $N(S) + w - 1$ until the retransmission of the lost I-frame is completed.
- The loss of a SREJ-frame or of a retransmitted I-frame is not detected until a timer expires and thus further delays the completion of the retransmission.[3]

The recovery of each lost I-frame takes at least one roundtrip time and frame losses during the recovery add extra delay proportional to the timeout values.[4]

Fig. 4 depicts one possible loss-scenario for the proposed method. In this scenario the window size w is five. Two consecutive I-frames 2 and 3 get lost and the recovery of I-frame 3 is delayed by the loss of a SREJ-frame. The way SREJ is implemented, at most one SREJ-frame can be outstanding at any time. When the receiver receives an I-frame with sequence number $N(S)$ it generates a SREJ-frame when $N(S) \neq V(R)$ and there is no other SREJ-frame outstanding. When there is more than one I-frame lost, the first one lost will be recovered and acknowledged. In Fig. 4, the acknowledgment RR(3) then advances the window and allows I-frame 7 to be transmitted. When I-frame 7 is received the recovery of I-frame 3 begins.

To evaluate the performance from the network's perspective, we compare the network efficiency of the different strategies. The network efficiency is defined as the ratio between the

[3] The timeout values are typically larger than the roundtrip time.

[4] If no loss occurs, the transmitter can send wI-frames during one roundtrip time.

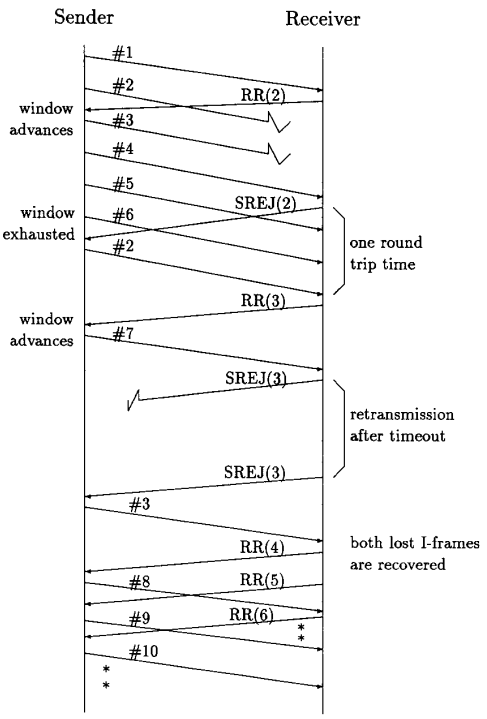

Fig. 4. Transmission with loss.

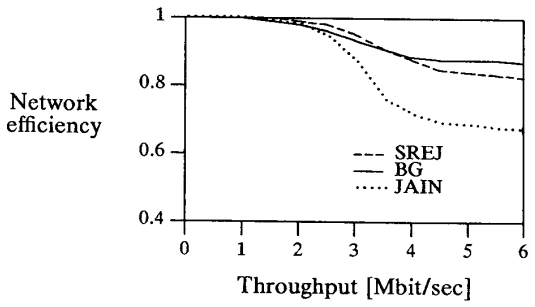

Fig. 5. Network efficiency versus throughput.

number of I-frames delivered to the user at the receiver's side and the number of I-frames transmitted over the network. We see in Fig. 5 that under overload BG has a slightly better network efficiency than SREJ. This is because the loss rate of I-frames for SREJ is higher than for BG. Because JAIN suffers from a significantly higher loss rate than BG or SREJ, its network efficiency is much lower.

An earlier simulation study by Brady [5] investigated various enhancements to the LAPB and LABD protocol and showed that for a scenario with high burst error rates due to noise, SREJ did not perform as well as other variants of the protocol. The disagreement between Brady's and our results is due to the fact that the two studies use completely different simulation scenarios and make different assumptions. In Brady's scenario, frame loss is only due to noise; congestion loss does not occur. Our study is concerned with the behavior of the protocol under overload when a significant number of frames is lost due to congestion; we do not consider errors due to noise. In todays' networks, loss due to congestion is much more likely than loss time due to noise. When an I-frame is corrupted or lost during retransmission, it is not recovered until a timeout occurs. In Brady's scenario, the increased number of timeouts is responsible for the poor performance of the SREJ protocol. In our scenario, as we have explained above, the delayed recovery of lost I-frames significantly improves the overall performance because it eases congestion by throttling the sender.

IV. Comparison of BG and SREJ

This section discusses some complexity and control issues of BG and SREJ. Dynamically adjusting the window size as suggested by Bux and Grillo poses two problems. Since BG takes a REJ-frame as an indication of congestion, it is susceptible to "false alarms" when the receiver generates a REJ-frame because of misordered I-frames[5] or in response to a corrupted I-frame. Another problem is finding the right window adjustment policy. In BG, the parameter n determines how fast the window size is increased. Bux and Grillo present some results that express the sensitivity of the throughput and delay with respect to n and then pick $n = 8$ as a reasonable choice for the scenario simulated. However, no statement is made on how to choose n in general and whether the value of n does depend on the window size w as in Jain's scheme. A simulation study by Nassehi [6] indicates that the choice of the window adjustment policy depends on the bit error rate of the transmission medium. Nassehi compared the policies of BG and JAIN. He found that the policy of Bux and Grillo performed better for low bit error rates and the policy by Jain performed better for higher bit error rates. Our simulations, in which the bit error rate is zero, corroborate this result.

In SREJ, the receiver must store I-frames that are out-of-sequence, which requires memory to hold up to w I-frames per connection.[6] Providing memory and managing this much memory is not difficult. Today, it is feasible to provide a single end node with several Mbits of storage. This is sufficient to hold the I-frames for many connections that are open simultaneously even with window sizes much larger than the ones in our scenario. Efficient memory management techniques are available. When the memory management is done in software, a data structure consisting of small blocks of data that are linked together to hold one I-frame can be used. Such data structures, called *Mbufs,* are used in the kernel implementation of the protocols under the Berkeley UNIX system [7]. If the memory can be accessed only in FIFO-order, Gopal and Rom [8] suggest organizing the memory that holds the I-frames for each connection as a small set of FIFO buffers. Adjacent I-frames are kept in the same FIFO buffer. Every time a new I-frame is received that does not follow immediately the last I-frame stored in any of the FIFO buffers, the I-frame is put into an new FIFO buffer, if an

[5] Mulitpath routing is one example that can lead to reordering.

[6] Even when the receiver discards I-frames that are out-of-sequence it is necessary to provide enough buffer space to hold one window worth of I-frames so that the next higher layer can apply backpressure across the interface to the LLC layer.

empty FIFO buffer is available, and discarded otherwise. If the number of FIFO buffers is one, all out-of-sequence I-frames are discarded. If the number is equal to or larger than the window size, no out-of-sequence I-frame will be discarded. The number of FIFO buffers needed to approximate the performance of SREJ depends on the window size, the loss rate, and the loss pattern. We have implemented this scheme in our simulator. For a window size of ten, as few as two FIFO buffers are sufficient to achieve a performance close to the performance of SREJ. For larger window sizes the number of FIFO buffers needed to achieve the performance of SREJ increases to about five.

ACKNOWLEDGMENT

I would like to thank B. Gerauer for his help with the simulator and the referees for their detailed comments on an earlier version of the paper.

REFERENCES

[1] Institute of Electrical and Electronics Engineers, "Logical link control," IEEE Std. 802.2–1985, IEEE, New York, 1985.
[2] W. Bux and D. Grillo, "Flow control of interconnected token rings," *IEEE Trans. Commun.*, vol. COM-33, pp. 1058–1065, Oct. 1985.
[3] R. Jain, "A timeout-based congestion control scheme for window flow-controlled networks," *IEEE J. Select. Areas Commun.*, vol. SAC-4, pp. 1162–1167, Oct. 1986.
[4] A. Tanenbaum, *Computer Networks, 2nd Edition.* Englewood Cliffs, NJ: Prentice-Hall, 1988.
[5] P. T. Brady, "Performance evaluation of the multireject, selective reject, and other protocol enhancements," *IEEE Trans. Commun.* vol. COM-35, pp. 659–666, June 1987.
[6] M. Nassehi, "Window Flow Control in Frame-Relay Networks," in *Proc. IEEE GLOBECOM 88,* Hollywood, FL, Nov. 1988, pp. 1784–1790.
[7] S. J. Leffler, M. K. McKusick, M. J. Karels, and J. S. Quarterman, *Design and Implementation of the 4.3BSD UNIX Operating System.* Reading, MA: Addison-Wesley, 1989.
[8] I. Gopal and R. Rom, "Improving ARQ protocol performance by multiple FIFO buffers," in *Proc. IEEE INFOCOM 90,* San Francisco, CA, June 1990, pp. 426–432.