THE EFFECT OF BANDWIDTH MANAGEMENT ON THE PERFORMANCE OF A WINDOW-BASED FLOW CONTROL

Daniel T. Luan and David M. Lucantoni

Daniel T. Luan *is a member of technical staff and* ***David M. Lucantoni*** *is a distinguished member of technical staff in the Teletraffic Theory and System Performance Department of AT&T Bell Laboratories in Holmdel, New Jersey. Both are working on congestion control and flow control for integrated services packet networks. Mr. Luan joined the company in 1981. He has a B.S. in electrical engineering from National Taiwan University, an M.S. in management science from National Chiao-Tung University, and a Ph.D. in operations research from Yale University. Mr. Lucantoni joined AT&T in 1981 and has a B.S. in mathematics from Towson State University, Baltimore, Maryland; and an M.S. in statistics and a Ph.D. in operations research from the University of Delaware.*

In this paper, we derive an algorithm for computing various performance measures for a window-based flow control, subject to bandwidth management. Bandwidth management is a mechanism for maintaining a fair allocation of network resources among customers. We provide an analytic procedure for computing the transmission rate allowed by the window, as well as the throughput of successfully transmitted frames (excluding retransmissions) that the customer can achieve. We will show that, when the network is at a given level of congestion, a customer can choose an appropriate window size to optimize the throughput of successfully transmitted frames, subject to bandwidth management.

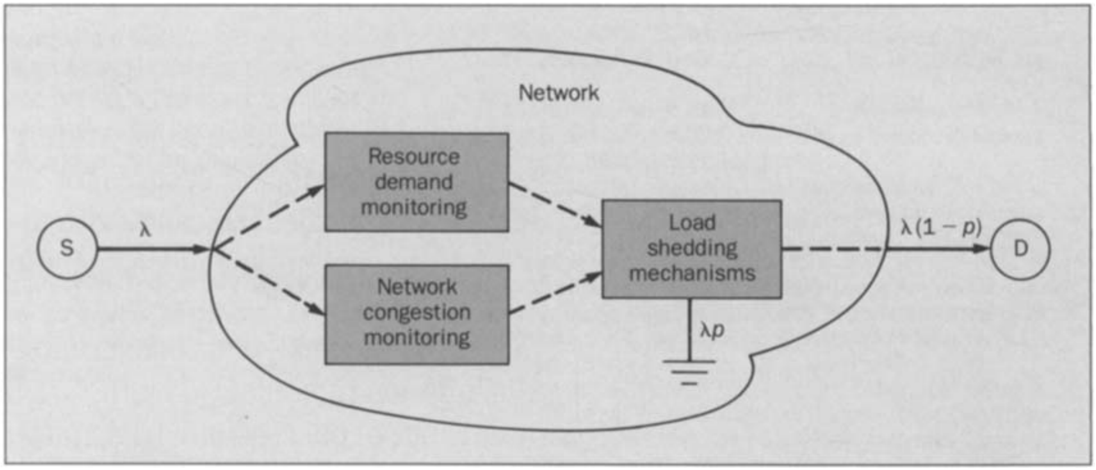
Introduction

This study analyzes the performance of an end-to-end window-based flow control, subject to bandwidth management (BWM). In particular, we model a high-level data link controller (HDLC-type) layer 2 protocol, based on positive acknowledgments for correctly received frames. The D-channel link-access procedures (LAPD) are currently under consideration by standards bodies for use in frame transport via end-to-end "frame relay."[1] We explicitly model the effect of frame losses on performance.

The effect of lost frames takes on a greater significance when some type of bandwidth management is used by the network.[2,3] The objective of BWM is to keep some customers from overloading the network and degrading performance for all customers. With BWM, at the initiation of a call, a customer negotiates with the network for an average or guaranteed throughput. The network then monitors the rate at which such bandwidth and resource demands are being created by end users. When overload conditions exist, frames associated with demands greater than those that were negotiated can be discarded.

When a frame is lost, there are two main recovery mechanisms. If other frames in the same window are not lost, the receiver will notice out-of-sequence frames and send a REJECT frame to the

Figure 1. A high-level description of bandwidth management. S = source; D = destination.



transmitter. The transmitter then retransmits all frames beginning from the lost frame. Further, if the same frame is lost again, or if all frames in the window are lost, the transmitter will eventually time out and retransmit the entire outstanding window of frames. The time-out mechanism is also incorporated explicitly into our analysis. The main performance measure we consider is the good throughput or "goodput" (i.e., the throughput of frames, excluding retransmissions, that the customer can achieve).

Most previous analyses of window-based flow controls use closed queueing network models to capture the effect of the window and, thus, are not able to model lost frames explicitly.[4] On the other hand, various models of automatic-repeat-request (ARQ) schemes usually capture the effect of retransmitting lost frames by excluding the time-out recovery and the window-imposed limitations on the transmission rate.[5] References 6 and 7 contain simulation studies on certain LAPD-based protocols. Brady investigates the benefits of the "multi-reject" option;[6] Rege and Chen propose and evaluate additional flow/congestion control measures to support LAPD frame relay.[7] Reference 8 contains an analytical model of HDLC that admirably incorporates all relevant parameters and mechanisms of the protocol without bandwidth management. The key here is to derive the so-called "virtual transmission time" as the mean time between successive, successful transmissions of frames. Our model uses the idea of identifying the *epochs* (particular instants of time) of successful retransmissions of previously lost frames as regeneration points. All relevant performance measures can thus be studied by observing the system behavior between two regeneration points. Further, in the presence of BWM, there is a mutual dependence between frame loss rate and throughput. Our approach easily captures this effect in a simple iterative procedure.

We will show that, when the network is at a given level of congestion, a customer can choose an appropriate window size to optimize goodput, subject to BWM. In a follow-up paper, we use the model developed here as the basis for the analysis of an adaptive end-to-end window flow control.[9] That is, given a fixed negotiated bandwidth, the customer can dynamically adapt the window size based on perceived network congestion.

Bandwidth Management

As explained above, bandwidth management maintains a fair allocation of network resources among end users and keeps some end users from overloading the network and degrading the performance of others. BWM also allows users to use excess capacity in the network when it is available. It is preferable to have bandwidth management limit a customer's throughput only when the excess throughput would cause congestion in the network.

As shown in Figure 1, there are always three components to any BWM mechanism. The first component monitors the resource demand of the end user. Resource demand could be measured in terms of bandwidth desired or real time required to process the transmitted frames. The resource demand monitor compares the actual demand with the level that the end user negotiated with the network. The second component is a monitor of network congestion levels. The results of these two monitoring components are fed into the load shedding component. When (and only when) the network is congested, excess frames (frames above the negotiated level) are discarded. When there is no congestion in the network, an end user is permitted to use network resources above what was originally negotiated.

For the remainder of this paper, we assume that some type of BWM mechanism with the following properties is in place.

If the congestion in the network is low, an end user may transmit above the negotiated bandwidth while encountering only negligible frame losses. However, if network congestion increases, the frame-loss probability associated with the excess traffic above the negotiated bandwidth level will increase. In heavy congestion, all excess traffic will be lost with high probability. We assume that call acceptance/denial algorithms are in place in the network to ensure that the loss of traffic sent within the negotiated bandwidth will be negligible. This is so that, in heavy congestion, traffic up to the negotiated bandwidth level will still encounter only negligible losses. The end user might view frames sent above the negotiated level as "opportunity" frames, because the user transmits these "at risk" (i.e., realizing that they may not get through).

The Model and Analysis

We analyze throughput for the standard LAPD layer 2 protocol in "multiple-frame operation." When the transmitter has frames to send, it may send W frames consecutively, where W is the window size. At the correct reception of each frame, the receiver sends an ACK (acknowledgment) back to the transmitter. At the reception of each ACK, the transmitter sends the next frame if one is ready to be sent. If a frame is lost before it has been received, there are two major recovery mechanisms.

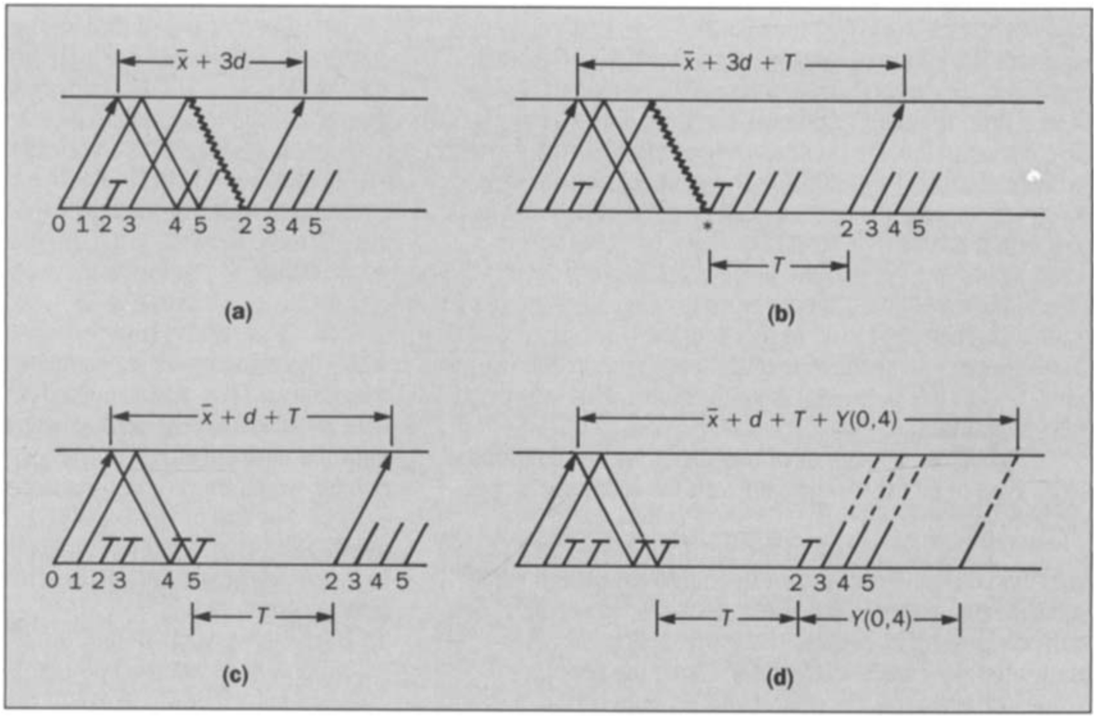
First, if any of the succeeding frames in the window is correctly received, a REJECT is sent back to the transmitter. This REJECT message specifies which frame the receiver is expecting. The transmitter then retransmits an entire window's worth of frames beginning with the lost frame. That is, selective rejects are not allowed. In addition, only one REJECT message can be outstanding. This means that, once the receiver sends a REJECT, it will wait until the lost frame has been successfully retransmitted and received.

The second recovery mechanism is activated when the transmitter does not receive any response from the receiver. This can happen if all frames in a window are lost. In this case, the transmitter times out and retransmits the entire window of frames, starting from the first one for which there was no acknowledgment.

For the present analysis, we consider one-way information transfer only. The return channel is assumed to be errorless and congestion-free. We also make the following assumptions:

1. Because we want to find the throughput determined by window size, we assume that the transmitter always has a full window's worth of frames to transmit.
2. Frames transmitted during a window are separated on the average by d seconds, the average time required to transmit a frame onto the access line.
3. The time-out interval, T, is longer than the average round-trip delay of a frame. This prevents premature time-outs. In our analysis, T includes the delay time for polling the receiver after the time-out expiration.
4. In general, there is a limit on the number of time-out retransmissions. When this limit is reached, the higher layer and management entities are notified and procedures for re-establishment of multiple-frame operation may be initiated. For ease of presentation, we assume no such limit in the analysis; however, it is not difficult to incorporate this effect explicitly in our model.
5. To ensure that window size is the limiting factor for throughput, we assume that the average round-trip delay, $\bar{x}$, is longer than the time the transmitter needs to transmit a full window's worth of frames (e.g., $\bar{x} > Wd$).

Figure 2. Typical recovery scenarios for $W = 4$, $i = 2$. T = time-out interval.



6. We assume that each frame has a fixed probability of loss. This probability will be a function of the transmission rate that is allowed by the window flow control as well as the BWM mechanism. In reality, frames that are transmitted above the negotiated level will have a higher loss probability than frames sent below that level. In our analysis, however, we assume that the loss probability of an arbitrary frame is an appropriate weighted sum of the respective probabilities of loss for frames sent above and below the negotiated level.
7. We compute average performance measures. For example, given the average round-trip delay, $\bar{x}$, and assuming that all frames experience a fixed delay equal to $\bar{x}$, we compute the expected throughput of the virtual circuit.

Based on the first assumption listed above, we note that these results will be applicable when throughput is limited by window rotation. This is the region of interest when assessing the effect of the window on sustained throughput. The most critical assumption above is that of independent frame-loss probabilities for successive frames. However, this will lead to conservative estimates for actual throughput.

With a window size of 1, only time-out recovery is possible for frame loss; the analysis of this case is straightforward.

The following arguments relate to the nontrivial case of a window size greater than 1. A typical scenario is shown in Figure 2. Each time that the receiver correctly receives a *retransmitted* frame (i.e., a frame that had previously been lost), it is expecting the next frame in sequence that the transmitter is sending. There will always be $W-1$ frames transmitted immediately after the correctly received frame was transmitted. It is then obvious that these epochs (of correctly received, retransmitted frames) form *regeneration* points. Therefore, we need only compute the expected time between regeneration points, the expected number of successfully transmitted frames between these points, and the total number of transmitted frames between these points.

Let $Y(i, W)$ be the expected time between two successive, successful receptions of retransmitted frames, given that, after the first successful reception, the ith frame is lost, $i \geq 1$, and the window size is W.

THEOREM 1. *For $W \geq 2$ and $1 \leq i \leq W$, we have*

$$\begin{aligned} Y(i, W) = & \sum_{k=i+1}^{W-1} p^{k-(i+1)}(1-p)(\bar{x} + kd) \\ & + \sum_{j=W}^{W+i-1} p^{j-(i+1)}(1-p)(2\bar{x} + (j - W)d) \\ & + p^{W-1}(\bar{x} + (i-1)d) + T\left[\sum_{k=1}^{W-2} p^k + 2p^{W-1}\right] \\ & + p^W Y(0, W), \qquad 1 \leq i \leq W-1, \end{aligned}$$

$$\begin{aligned} Y(W, W) = & \sum_{j=W+1}^{2W-1} p^{j-(W+1)}(1-p)(2\bar{x} + (j - W)d) \\ & + p^{W-1}(\bar{x} + (W-1)d) \\ & + T\left[\sum_{k=1}^{W-2} p^k + 2p^{W-1}\right] + p^W Y(0, W) \qquad (1) \end{aligned}$$

where

$$Y(0, W) = (1-p^W)^{-1} \times \left\{\sum_{k=1}^{W-1} p^{k-1}(1-p)(\bar{x} + kd) + T\left[\sum_{k=1}^{W-2} p^k + 2p^{W-1}\right]\right\}$$

Note that

$$\sum_{k=i}^{j} = 0$$

when $i > j$. And finally, for $i = NW + j$, with $N \geq 0$ and $1 \leq j \leq W$, $Y(i, W) = N\bar{x} + Y(j, W)$.

Proof. The proof of this result is obtained by enumerating all possible scenarios. It is tedious, so we will only show the argument for $W=4$ and $i=2$. From Figure 2 we see that, if the first frame after the last successful retransmission to be lost is $i=2$, there are several possible ways to recover. If frame 3 (the fourth frame in the current window) is correctly received, a REJECT message is sent to the transmitter. (See Figure 2a.) Note that the probability that frame 3 is not lost is $1-p$. When the REJECT is received, the transmitter retransmits four frames, beginning with frame 2. Now, with probability $1-p$, frame 2 will be correctly received and the total cycle time is $\bar{x} + 3d$. With probability p, frame 2 will again be lost. (See Figure 2b.) Frames 3, 4, and 5 have no effect on the system because there can be one outstanding REJECT message at most. In this case, after a time-out interval of length T, the transmitter will again retransmit frame 2. From this point on, any attempt to retransmit frame 2 has a probability of success equal to $1-p$ and each unsuccessful attempt results in another time-out interval of length T. Therefore, the total expected time before the correct reception of frame 2 is

$$\begin{aligned} (1-p)\Bigg[(\bar{x} + 3d)(1-p) + & \\ & \sum_{k=1}^{\infty} [(\bar{x} + 3d) + kT]p^k(1-p)\Bigg] \\ & = (1-p)(\bar{x} + 3d) + pT \qquad (2) \end{aligned}$$

Similarly, we see that with probability $(1-p)p$, frame 3 is lost and frame 4 (which happens to be in the next window) is correctly received. A REJECT is sent and, if frame 2 is now correctly received (with probability $1-p$), the expected time until the end of the cycle is $2\bar{x}$. A second loss of frame 2 will result in a geometric number of time-out intervals so that in this case the expected cycle time is

$$\begin{aligned} p(1-p)\left[(2\bar{x})(1-p) + \sum_{k=1}^{\infty}[2\bar{x} + kT]p^k(1-p)\right] & \\ = p(1-p)(2\bar{x}) + p^2T & \qquad (3) \end{aligned}$$

If frames 3 and 4 are lost but frame 5 is successful [with probability $p^2(1-p)$], a similar argument shows that the expected cycle time is $p^2(1-p)(2\bar{x}+d)+p^3T$. If frames 3, 4, and 5 are all lost (with probability p^3), we have the case shown in Figures 2c and 2d. Because the full window was lost, the only recovery mechanism is the time-out. If the first retransmission of frame 2 is successful, the expected cycle length is seen to be $\bar{x}+d+T$. If it is not successful, we have a slightly different case. (See Figure 2d.) We need to compute the expected time from the second loss of frame 2 until its first successful reception. We call this expected time $Y(0,W)$. $Y(0,W)$ differs from all the intervals we have studied so far in that the first frame in the window is lost. Recall that all other cycles begin with the first frame in the window being successful. At first glance, $Y(0,W)$ appears to be the same as the time marked by the asterisk (*) in Figure 2b until the end of the cycle; but in that case, there can be no more REJECT messages. However, in Figure 2d, a successful transmission of any frame other than 2 will result in a REJECT. The expression for $Y(0,W)$ in the theorem is obtained by analogous enumerations. It can be verified that the expressions in the theorem are correct for the general case.

COROLLARY 1. *The expected regeneration cycle (i.e., the time between the epochs of successful reception by the receiver of retransmitted frames) is given by*

$$E(Y)=\sum_{i=1}^{\infty} p(1-p)^{i-1}Y(i,W) \tag{4}$$

Proof. This follows directly from the assumption that frames are lost independently and with probability p.

COROLLARY 2. *The total transmission rate or throughput of frames transmitted by the transmitter as determined by window size is*

$$E(TR) = \frac{1}{E(Y)}\left[\frac{1}{p} + \frac{W}{1-p}\right] \tag{5}$$

Proof. The total number of frames transmitted during a cycle in which the ith frame following the start of the cycle is in error is given by $i+W/(1-p)$. To see this, note that the number of retransmissions required before the errored frame can be successfully received is geometrically distributed. In each retransmission, a full window's worth of frames is sent. Therefore, during a regeneration cycle, the expected number of frames transmitted is

$$\sum_{i=1}^{\infty} p(1-p)^{i-1}\left[i + \frac{W}{1-p}\right] = \frac{1}{p} + \frac{W}{1-p} \tag{6}$$

Equation (6) has the following intuitive interpretation. The expected number of frames transmitted during a regeneration cycle is the expected number transmitted before the first lost frame ($1/p$) plus W times the expected number of retransmitted windows $[1/(1-p)]$.

Without loss of generality, we assume the time origin is chosen as an epoch of the end of a regeneration cycle. Now let X_k be the kth regeneration interval and let Z_k be the number of transmissions in X_k. Then (X_k, Z_k) is a renewal reward process. If $N(t)$ is the total number of renewals in $(0,t]$ and $Z(t)$ is the total number of transmissions in $(0,t]$, then

$$Z(t) = \sum_{k=1}^{N(t)} Z_k$$

and the expected transmission rate is given by

$$\lim_{t\to\infty}\frac{Z(t)}{t}$$

From a well-known result on renewal reward processes (see, for example, Reference 10, Theorem 3.16, pp. 52-53), we have

$$E(TR) = \lim_{t\to\infty}\frac{Z(t)}{t} = \frac{E[Z_k]}{E[X_k]}$$

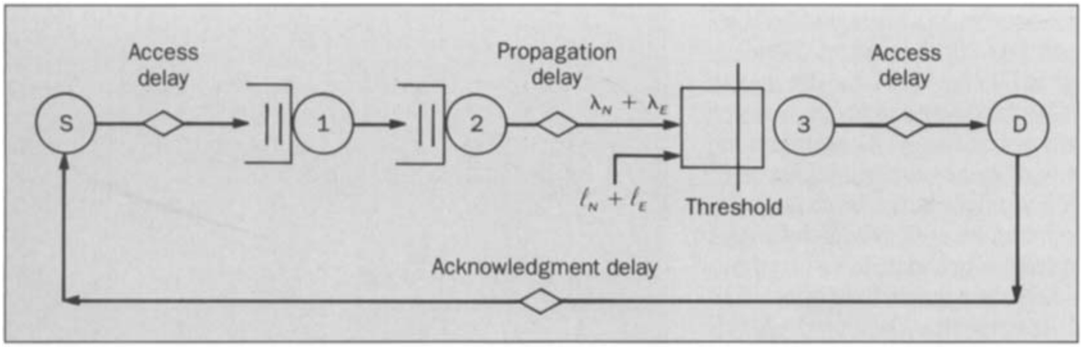


Figure 3. Reference connection for the targeted virtual circuit.

COROLLARY 3. *The goodput, E(G) (the expected throughput of successfully transmitted frames), is given by*

$$E(G) = \frac{1}{pE(Y)} \tag{7}$$

Proof. From Figure 2 we see that, during a cycle where the ith frame is lost, the number of successfully transmitted frames before the correct retransmission of the ith frame is i (remember to count the frame that started the cycle). The result follows by an argument analogous to that in Corollary 2.

Numerical Results

In this section, we show some results that indicate the performance of the window-based flow control, subject to BWM. To produce numerical results, we need a simple analytical model that captures qualitatively the desirable BWM properties discussed above. That is, it allows customers to transmit above their negotiated throughput when the network is not congested, but when congestion does occur, the excess throughput is discarded. These effects can be captured mathematically by a simple finite M/M/1 model with a threshold. Assume that the arrival streams of frames are split into two streams, a normal stream and an excess stream. The normal stream consists of all frames that were transmitted at or below negotiated throughput; the excess stream consists of all frames that were transmitted in excess of negotiated throughput. In our results, we assume a finite buffer with a capacity to store 75 frames; *excess* frames will only enter the buffer when there are less than 15 frames currently occupying the buffer.

We study the performance of a typical virtual circuit (the targeted virtual circuit) as shown in Figure 3. The average frame length is 128 bytes. The access lines are 64-kilobits-per-second (kb/s) lines and the network links are 1.5 Mb/s, so that it takes 16 milliseconds (ms) on average to transmit a frame onto the access line and the effective service rate at the network nodes is $\mu = 1500$ frames per second (frames/s). Each node has traffic from other virtual circuits as well as the one we are considering (the targeted virtual circuit). We assume that nodes 1 and 2 are lightly loaded and that node 3 is congested.

Therefore, from the targeted virtual circuit's point of view, the service rate at nodes 1 and 2 will be scaled down. We assume that this effective service rate is 600 frames/s at both nodes (which corresponds to 60-percent utilization by other traffic).

Let the total normal load from the other virtual circuits going through node 3 be ℓ_N and the total excess load from the other virtual circuits through node 3 be ℓ_E. We assume that the targeted virtual circuit always has frames to send so that the limiting factor on the transmission rate is the window (and the BWM).

Let us fix the negotiated throughput for the targeted virtual circuit at 30 frames/s. (For example, the user may expect to transmit at an average rate of 25 frames/s and may negotiate for 30 to allow for some burstiness in the traffic.) Let the *transmission rate* for the targeted virtual circuit allowed by the window be λ. This traffic will be split into λ_N and λ_E, where, if $\lambda \leqslant 30$, then $\lambda_N = \lambda$ and $\lambda_E = 0$. If $\lambda > 30$, then $\lambda_N = 30$ and $\lambda_E = \lambda - 30$.

In the finite M/M/1 model discussed above, we model the total arrival stream of frames as a Poisson process with rate $\ell_N + \ell_E + \lambda_N + \lambda_E$. Service times are assumed to be independent, and identically distributed, with exponential distribution with parameter $\mu = 1500$. With these assumptions, the number of frames in the node is a continuous-time Markov process and it is a simple matter to compute the probability of loss for both normal frames and excess frames, p_N and p_E, as well as the expected sojourns at the node, w_N and w_E, given that the frames are not dropped. (Note that normal frames can also be lost due to the finite buffer.) The average probability of frame loss and average sojourn time at the node for arbitrary frames of the targeted virtual circuit are obtained as:

$$p = \frac{\lambda_N p_N + \lambda_E p_E}{\lambda_N + \lambda_E}$$

$$\bar{w} = \frac{\lambda_N(1-p_N)w_N + \lambda_E(1-p_E)w_E}{(\lambda_N + \lambda_E)(1-p)}$$

Nodes 1 and 2 are modeled as M/M/1 queues with arrival rate λ and a service rate of 600 frames/s. The access lines add a 16-ms delay on each side so that if we assume a one-way propagation delay of 25 ms in each direction, the total average round-trip delay for the targeted virtual circuit is $\bar{x} = 0.016 + (600 - \lambda)^{-1} + (600 - \lambda)^{-1} + 0.025 + \bar{w} + 0.016 + 0.025$ seconds, or:

$$\bar{x} = 0.082 + \bar{w} + \frac{2}{600 - \lambda}$$

We now have the ingredients for our algorithm to compute the goodput of the window flow control, subject to BWM. Based on the assumption that the targeted virtual circuit always has frames to send, the final step in our analysis is to compute the realizable transmission rate and goodput. In our analysis, we must assume a transmission rate λ in the BWM model, which then leads to a frame loss probability, p, and a round-trip delay, $\bar{x}$. From Corollary 2,

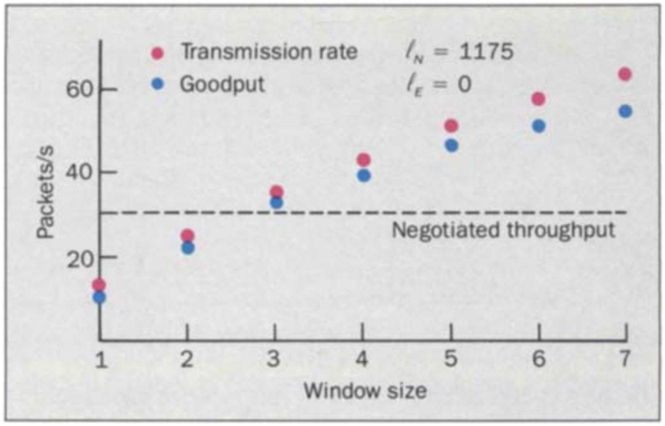


Figure 4. Throughput performance at mild network congestion.

we can compute the effective transmission rate allowed by the window, $E(TR)$. In general, this will be different from λ (because λ affects p and $\bar{x}$, which affect λ, etc.)

For consistency in our model, we need $\lambda = E(TR)$. This is accomplished by a simple iteration procedure. The obvious scheme is to pick any λ and, from the analysis, compute $E(TR)$. Now let $\lambda \leftarrow E(TR)$ [i.e., replace λ by $E(TR)$], and repeat until they converge. Unfortunately, for some window sizes, this procedure oscillates between two values of λ, one greater than 30 and one less than 30.

This behavior is intuitive because, if $\lambda < 30$, there are no excess frames. Thus, p is very small, which leads to a large $E(TR)$. If $E(TR)$ is much greater than 30, then many excess frames lead to a large p. This again leads to a small $E(TR)$.

However, it is clear that there is an equilibrium point; one way to obtain it is to damp the oscillations. For example, letting $\lambda_{new} \leftarrow 0.7\lambda_{old} + 0.3E(TR)$ worked well in our examples. This allows for a slow increase in λ and avoids the oscillation.

Figures 4 and 5 are two representative examples. For both, $T = 0.26$ seconds, which includes a 200-ms time-out interval and 60 ms for the round-trip delay of the polling message. In Figure 4, we assume that the node is mildly congested. The background parameters are

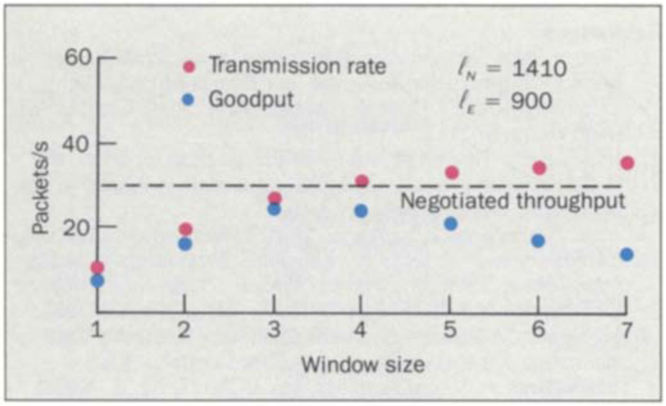


Figure 5. Throughput performance at heavy network congestion.

$\ell_N = 1175$ and $\ell_E = 0$, so that the line occupancy due to the background is 0.78 and there are no excess frames. We see that by increasing the window size, the user can steadily increase the transmission rate. The increase is not linear because the higher rate increases the round-trip delay. We note that at higher window sizes, even though the targeted circuit is generating many excess frames, there is not much loss as indicated by the goodput curve. Therefore, under these network conditions, a customer can increase the goodput by increasing the window size (up to 7, at least).

A more congested case is shown in Figure 5. Here we have $\ell_N = 1410$ and $\ell_E = 900$, so that the line occupancy due to background normal frames is 0.94 and the total offered load is 1.54. Thus, many excess frames will be dropped. In this case, we see that the targeted circuit will experience degraded performance at the larger window sizes. The difference between the transmission rate and the goodput in the figures is due to the retransmission of lost frames. In this case, the customer will optimize the goodput by using a window of 3 or 4. Note that, although the goodput for $W = 4$ is slightly higher than that for $W = 3$, there are more frame losses and retransmissions at $W = 4$. If these are undesirable, it might be preferable to operate at $W = 3$.

We also point out that the simple analytic model for BWM captures the properties desired for BWM. For example, in the situation described by Figure 5, even though the total offered load to the outgoing link is above its capacity, an end user can still achieve a favorable goodput by transmitting just below the negotiated bandwidth. This will be the case when the total negotiated traffic is less than capacity. An appropriate call acceptance/denial algorithm should be used in the network to ensure this.

Figure 6 shows the effect of BWM on performance, by showing performance in a case of heavy congestion without BWM. These results were obtained by modifying the simple queueing model of BWM so that both normal traffic and excess traffic are dropped only when the buffer is full. In the heavy congestion case, the ratio of the total offered load to the channel capacity is 1.54. Therefore, the probability of packet loss will be high (approximately 0.35) and time-out recovery will be the most common recovery mechanism. This severely limits the allowed transmission rate so that even a window size of 7 only allows a rate less than 30 packets/s. The actual goodput that can be achieved is less than 7 packets/s. Thus, we see that some form of BWM is needed to provide fairness by protecting normal traffic and to improve the overall performance of all traffic during heavy loading.

Conclusion

We have derived an algorithm for computing various performance measures for a window-based flow control, subject to bandwidth management. In particular, we have an analytic procedure for computing the transmission rate allowed by the window as well as the expected goodput (i.e., throughput minus retransmissions). We have shown that, under given network congestion conditions, a user can choose an appropriate window size to optimize the goodput.

Several minor modifications of our analysis are possible to incorporate other aspects of the protocol. For example, it is possible to incorporate some dependence in the frame loss probabilities. The first frame in a window can have a different loss probability from the others. This involves a simple adjustment to equation (1).

Also, because our intent was to show the effect of bandwidth management on window flow control, we concentrated on frame losses and ignored other types of

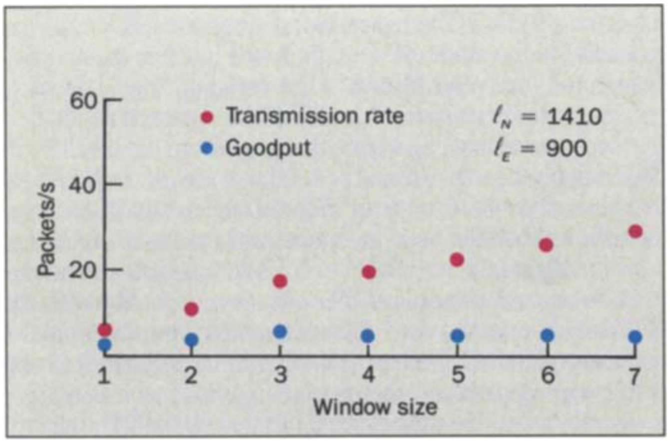


Figure 6. Throughput performance at heavy network congestion without bandwidth management.

frame errors. It is clear that the same types of regeneration arguments apply in the presence of errored frames. Computing the expected length of the regeneration cycle in this case requires the consideration of several more cases than those considered in the proof of Theorem 1.

Several modifications of our analysis make it possible to model other scenarios. It is possible to extend the idea of embedded regeneration epochs to analyze other protocols. In particular, we can analyze the multi-reject protocol studied by a simulation model in Reference 6. As shown above, for any fixed network congestion condition, a user can optimize the goodput; however, in general, congestion in the network is a transient phenomenon. It is clear that a fixed window size will not be optimal for all scenarios. Also, congestion is a function of the negotiated throughput levels and the amount of excess traffic being generated by all customers. The targeted customer will not know this.

In another paper, we analyze the performance of several simple, adaptive window-flow-control schemes.[9] In particular, we model network congestion as a random environment that randomly oscillates between mild congestion and heavy congestion. We look at several schemes in which the customer dynamically adjusts window size on the basis of perceived network congestion as indicated by, for example, lost frames.

References

1. "Frame Relay Service: A Proposed New Packet Mode Bearer Service," International Telegraph and Telephone Consultative Committee, CCITT Draft Recommendation, Study Group XVIII, Contribution No. D.717/XVIII, 1987.
2. J. S. Turner, "New directions in communications (or which way to the information age?)," *IEEE Communications Magazine*, Vol. 24, No. 10, October 1986, pp. 8-15.
3. A. E. Eckberg and P. E. Wirth, "Switch Overload and Flow Control Strategies in an ISDN Environment," *Proceedings of the 5th International Teletraffic Congress Seminar: Traffic Engineering for ISDN Design and Planning*, Lake Como, Italy, May 4-8, 1987.
4. M. Reiser, "A Queueing Network Analysis of Computer Communication Networks with Window Flow Control," *IEEE Transactions on Communications*, Vol. COM-27, No. 8, August 1979, pp. 1199-1209.
5. S. Lin et al., "Automatic-Repeat-Request Error-Control Schemes," *IEEE Communications Magazine*, Vol. 22, No. 12, December 1984, pp. 5-17.
6. P. T. Brady, "Performance Evaluation of Multi-Reject and Selective Reject Link-Level Protocol Enhancements," *Globecom '86*, pp. 3.7.1-3.7.5.
7. K. M. Rege and K. J. Chen, "A Performance Study of LAPD Frame-Relay Protocols for Packetized Data Transfer Over ISDN Networks," *Proceedings of the 5th International Teletraffic Congress Seminar: Traffic Engineering for ISDN Design and Planning*, Lake Como, Italy, May 4-8, 1987.
8. W. Bux et al., "Balanced HDLC Procedures: A Performance Analysis," *IEEE Transactions on Communications*, Vol. COM-28, November 1980, pp. 1889-1898.
9. D. T. D. Luan and D. M. Lucantoni, "Throughput analysis of an adaptive window-based flow control subject to bandwidth management," *Proceedings of the 12th International Teletraffic Congress*, Turin, Italy, 1988.
10. S. M. Ross, *Applied Probability Models with Optimization Applications*, Holden Day, Inc., San Francisco, California, 1970.

(Manuscript received August 19, 1988)