Design principles and performance analysis of SSCOP: a new ATM Adaptation Layer protocol

Thomas R. Henderson,
COMSAT Laboratories,
Clarksburg, MD 20871

Abstract-- **The Service Specific Connection Oriented Protocol (SSCOP) has been approved recently as a new B-ISDN ATM Adaptation Layer (AAL) protocol standard, initially for use in the Signaling ATM Adaptation Layer (SAAL), but also for support of certain types of user data transfer. SSCOP is a new type of protocol, embodying several design principles for high speed link and transport layer protocols. In this paper, the basic operation of SSCOP is described and the SSCOP design is compared with other similar protocols. Next, the relationships between key protocol parameters (protocol window, control message transmission interval) and maximum achievable throughput efficiency are explored. In particular, approximate performance equations are derived for predicting the maximum throughput efficiency of SSCOP based on the selected environment and parameter settings. The equations can be used to determine how much buffer capacity and/or what protocol timer settings allow SSCOP to operate at high performance. The analytical results are confirmed through comparison with simulation results. In addition, simulation results illustrate the high throughput performance achievable when using SSCOP in a highly errored or lossy environment.**

I. INTRODUCTION

In a typical data communications application providing assured data transfer, a link layer or transport layer protocol is used to ensure that lost or errored data is recovered through retransmission. A number of communications protocols, each suitable for different environments (e.g., X.25 LAPB, ISDN LAPD, TCP, OSI TP4, etc.) have been defined to provide such a service [1], [2].

However, the advent of broadband transmission networks and high speed switching technologies such as ATM has created a telecommunications network that will scale to a bandwidth many orders of magnitude greater than that of a few years ago. Along with this rise in network capacity, emerging applications are consuming more and more bandwidth. One consequence of this evolution is that some currently used data communications protocols (most notably, those based on HDLC principles [3]), when extended beyond their assumed operating environment, cause service (e.g., throughput, delay) degradation.

In particular, due to various design elements of these protocols, the achievable throughput efficiency is sensitive to increases in the bandwidth or round trip delay of the connection. For some time, satellite network operators have experienced problems with conventional protocols and have designed various changes to protocol logic in order to compensate for the added latency in their connections [4]-[6]. More recently, broadband networks have been faced with similar design issues [7]-[10].

Early in the development of ATM it was decided by ANSI Technical Subcommittee T1S1.5 that a new protocol would be needed for some ATM applications. At the time, work was already underway in defining the AAL type 3 protocol (later merged with the type 4 protocol to form AAL type 3/4), which provides error detection but not error correction. It was reasoned that error detection would be needed for most data applications but that error recovery would only be needed for signaling and on a service specific basis for data applications. Hence, the AAL was split into a "common part" and a "service specific part." In 1990-91, work was begun on the Service Specific Connection Oriented Protocol (SSCOP) [11], which incorporated many design principles for a high-speed protocol with lightweight (i.e., reduced processing) operation [12]-[14]. SSCOP has initially been specified as an ATM adaptation layer (AAL) protocol for use in signaling for ATM networks (combined with AAL type 5, it forms the signaling AAL, or "SAAL"). In particular, it provides most of the same services as provided by LAPD [15] in ISDN access signaling applications and Message Transfer Part (MTP) level 2 [16] in Signaling System No. 7.

The generic protocol engine provided by SSCOP is mapped by individual Service Specific Coordination Functions (SSCFs) into the required service for different applications, as shown in Figure 1. AAL users can

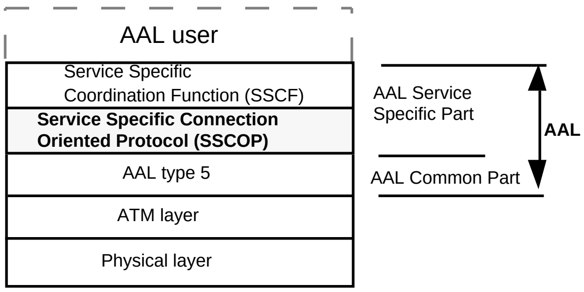

Fig. 1. SSCOP in the ATM architecture

access different services by using different SSCFs over a common SSCOP. Although the first two SSCFs defined were for support of UNI and NNI signaling [17], [18], the impetus for SSCOP's design was high speed user data transfer. Work is currently nearing completion in ITU-T Study Group 13 to define SSCFs for OSI (Open Systems Interconnection) network and transport layer service provision [19], [20]. Furthermore, SSCOP is not limited to ATM environments; it could also be used as a control protocol for Frame Relay, or in other environments with a high bandwidth-delay product or lossy channel (e.g., a recent paper [21] proposes the use of a similar protocol for wireless applications).

SSCOP is a new protocol standard for which the relationships between protocol parameters (e.g., the window size, timers) and the environment (e.g., the error/loss rate, the round trip delay) have not yet been explored in detail. This paper has three primary objectives:

i) to provide a high level description of how SSCOP works,

ii) to discuss the design features of SSCOP in the context of other similar protocols, and

iii) to explain how SSCOP protocol parameters may be selected to optimize the achievable throughput performance of the protocol in a given connection environment.

II. DESCRIPTION OF SSCOP

SSCOP belongs to a class of protocols known as synchronous bit-oriented protocols. Such protocols are responsible for flow control, sequencing and accounting of data, and actions to be taken upon error detection [2]. SSCOP transfers data to its peer in variable length protocol data units (herein referred to as "frames"). The transmitter sends frames to the receiver, and stores the frames for potential retransmission until the receiver has acknowledged them.

SSCOP's main function is to correct transmission errors or losses by selectively retransmitting the missing frames with a procedure known as automatic repeat request (ARQ). In particular, frames are numbered sequentially, and the receiver explicitly acknowledges the receipt of frames by sending an acknowledgment to the transmitter. If the receiver determines, through the examination of received sequence numbers, that a frame is missing, it explicitly requests a retransmission of the missing frame.

SSCOP relies on the selective retransmission class of continuous ARQ procedures, characterized by the receiver selectively requesting missing frames and the transmitter only retransmitting those specific frames that have been requested (i.e., there are no "time outs"). The receiver must allocate a resequencing buffer to hold correctly received frames until the missing frames are received and resequenced.

Flow control can be performed through the use of an adjustable sliding window. The receiver grants a "credit" window to the transmitter that allows it to transmit a certain number of frames. This credit value can be dynamically reduced or increased by the receiver.

There are four basic frames used for data transfer: the Sequenced Data (SD) frame for user data, and the POLL, STAT, and USTAT frames for control flow. User data is transferred in SD frames, which may be variable length up to 65535 bytes, and each of which contain a sequence number (modulo 2^{24}). Periodically, the transmitter requests, by sending a POLL frame, that the receiver respond with an update of its status. The POLL contains the sequence number of the next in-sequence (i.e., new) SD frame to be sent by the transmitter and a "poll sequence number" which essentially functions as a timestamp. This POLL can be triggered by the expiry of a timer (Timer_POLL) or, alternatively, after a certain number of frames have been sent. The receiver, upon receipt of a POLL, responds with a STAT (status) message that conveys the following information: the sequence number up to which the transmitter may transmit (i.e., the window), the number of the next in-sequence frame expected, the echoed poll sequence number, and a list of all currently outstanding SD frames, if any. The receiver knows which frames are outstanding by checking for "gaps" in its resequencing buffer and by examining the SD sequence number contained in the POLL message. The transmitter can then use the received STAT to release acknowledged frames from its retransmission buffer, to retransmit frames which the transmitter determines are lost, and to advance the transmit window as allowed by the receiver. One additional important feature of SSCOP is that when a new loss is detected, the receiver will immediately send a USTAT (unsolicited status) message that requests retransmission of only the newly detected missing frame(s). The USTAT is identical to the STAT except that it is not associated with any POLL frames and it does not include the complete list of missing frames. A retransmission can only be invoked once by a USTAT; if the frame is lost again, a new USTAT message will not be generated. The USTAT mechanism allows the transmitter to increase the interval between POLLs while still achieving rapid recovery from errors. This protocol control flow shares the same channel as the data frames, and is resilient to loss since each STAT contains the complete state information of the receiver.

The basic operation of SSCOP can best be understood by considering an example, shown in Figure 2. This example only illustrates one direction of operation, and assumes that the window offered by the receiver is large enough to keep the transmitter from idling. In the example, the transmitter sends a series of consecutively numbered frames. After frame #4 is sent, Timer_POLL expires and a POLL is sent. This POLL tells the receiver that the next message to be sent is #5, so that the receiver knows that messages 0 through 4 should have been received. In this case they have been, so the receiver returns a STAT frame acknowledging 0 through 4. The transmitter continues with frames 5 through 9 before the POLL timer expires again, but frame #7 is lost. The receiver detects this loss upon receipt of frame #8 and immediately requests retransmission of #7 with a USTAT frame. The transmitter always retransmits frames requested in a USTAT, so #7 is retransmitted (with priority over new

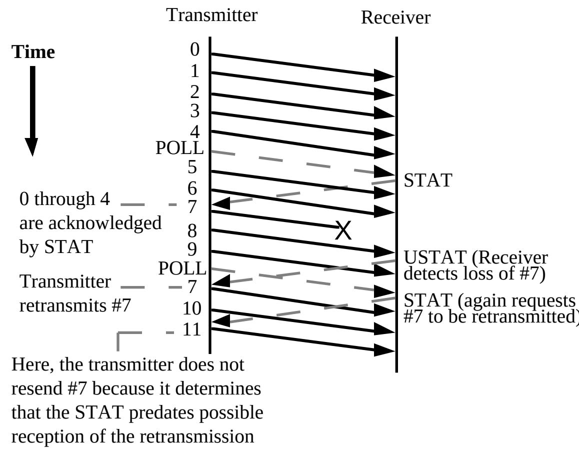

Fig. 2. Example of SSCOP operation

data). When the next POLL arrives at the receiver, #7 has not yet been received. The resulting STAT message again requests #7 for retransmission. However, the poll sequence number in the STAT message allows the transmitter to determine that the receiver has not yet received the retransmission, and an additional retransmission is inhibited. In this manner, SSCOP avoids any unnecessary retransmissions. If #7 were instead lost again, the next STAT message would have stimulated a second retransmission. Appendix II of the protocol specification [11] contains many more examples of SSCOP operation.

III. RELATIONSHIP BETWEEN SSCOP AND OTHER PROTOCOLS

When work on SSCOP was initiated by the telecommunications community, SSCOP was intended to be the generic reliable connection-oriented protocol for ATM based networks. Since that time, interest in using ATM to support TCP/IP in a distributed internet has become the driving force behind ATM deployment. In addition, TCP extensions for high bandwidth-delay networks [22] and optimization of TCP implementations (discussed in [23]) are allowing TCP to migrate towards a broadband environment. Even if closed loop, rate-based flow control and connectionless extensions to SSCOP are developed in the future, it is unlikely that SSCOP will supplant TCP as the predominant transport protocol for ATM environments. Nevertheless, SSCOP can be useful in the future for providing a high performance link layer protocol or connection-oriented network or transport layer protocol, especially for certain environments such as broadband connections requiring open-loop flow control, and mobile, wireless, and satellite networks.

Referring back to Figure 1, SSCOP was intended to be a generic protocol engine providing the function of error recovery in an efficient manner. Therefore, it does not include application specific functions (such as "link proving in," as needed for NNI signaling). The architectural model adopted was to specify a common SSCOP with basic functions, and to tailor the SSCOP service to the service required by applications through the use of the SSCF. Although SSCOP incorporated a couple of NNI signaling specific functions (such as data retrieval), it is essentially a basis from which to define application specific protocols by defining appropriate SSCFs. In this manner it resembles HDLC [3], the general procedures of which can be used as a "toolbox" to construct specific protocols.

SSCOP incorporates the following design features for high speed operation:

- the modulus is large (24 bits), and data is sequenced frame-by-frame; as a result, the sequence space should not limit the throughput of the protocol;
- the protocol is 32 bit aligned, and the protocol control information is trailer oriented in keeping with the design of AAL type 5 for reducing hardware bus cycles;
- error recovery is based only on selective retransmission;
- control and information flow is separated, reducing per-frame overhead;
- protocol logic is decoupled from timers, reducing the dependence of the implementation on timers and allowing for variations in round trip delay;
- the transmitter and receiver can be decoupled and do not share state variables, allowing for parallel implementation.

Selective retransmission error recovery for networks with a high bandwidth-delay product has been recommended for many years (e.g., [6]). Implementation experience with SNR [25] has revealed that complete state exchange coupled with implicit timers is a significant benefit in terms of protocol simplification and reduced reliance on timers. The benefits of parallel processing and insensitivity to variations in round trip delay are discussed in [8].

SSCOP was designed to work efficiently at high speeds and in lossy environments, both in terms of processing simplicity and throughput performance. During the past decade, a number of lightweight transport protocols have been developed for high speed networks; a review of many of these protocols (such as NETBLT, XTP, Datakit, and Delta-t) can be found in [1]. Of such previously developed protocols, SSCOP most closely resembles the SNR protocol [8] and the Datakit protocol [24]. The chief differences between SSCOP and SNR are that, in SNR, sequence integrity of the underlying network is not assumed, "STAT" messages are generated independently by the receiver (i.e., not stimulated by a "POLL" message), and retransmissions are based on lack of a positive acknowledgment within a specified number of received control frames. Since ATM is connection-oriented, SSCOP was able to assume underlying sequence integrity, and retransmissions could then be based explicitly upon requests from the receiver. The POLL message was added to SSCOP to allow the transmitter to control the rate of acknowledgment and to allow the protocol to escape from certain deadlock situations. The POLL mechanism makes SSCOP a "transmitter driven" protocol. This aspect of the protocol is useful in signaling, in which the POLL also functions as a "keep-alive" mechanism when no traffic is being exchanged. As for Datakit, it assumes sequence integrity of the underlying network and is also transmitter driven like SSCOP. However, Datakit differs from SSCOP in that it is based on byte streaming and retransmission is similar to Go-Back-N. SSCOP is a frame based protocol because of the underlying frame based AAL type 5.

SSCOP is presently lacking two key components that would extend its utility to more environments: rate-based flow control and operation over connectionless networks. As for flow control, the assumed environment for SSCOP has been one with open loop flow control (i.e., the available bandwidth and delay are static across the connection). SSCOP does not specify a closed loop flow control such as "TCP slow start" [26], although the same principles could be applied. Particularly if SSCOP is to be used at high speeds across different networks, the flow control must be rate-based in order to avoid overflowing buffers in hosts and gateways [7]. Use of the sliding window to approximate a rate control will likely not be sufficient. A rate-based flow control was not initially specified in order to complete the protocol specification for its use in the SAAL. However, a backward-compatible rate-based solution, possibly coupled with feedback from the ATM layer, is a future study item. As for use of SSCOP on a connectionless network, a relaxation of the rules against unnecessary retransmission of frames and a change in the retransmission procedures to account for the lack of sequence integrity among control and data frames are the necessary modifications.

IV. APPROXIMATE ANALYSIS

The exact analysis of discrete link protocols can be difficult [27]. In particular, if the system is modeled as a discrete Markov process, the number of states in the model can become intractable for high speeds or delays. Analysis of SSCOP is more difficult than that of HDLC type protocols because the control flow is periodic and detached from the data flow. Therefore, we will instead obtain approximate solutions for the protocol. The approximations are valid provided the error and loss rates are low.

We are interested in obtaining a relationship between the maximum obtainable throughput, protocol parameters (window, frame size, and POLL interval), and channel environment (bandwidth, delay, and error and loss rate). The goal is to develop relationships that will allow a user to maximize the throughput for a given environment by modifying protocol parameters.

A. Notation

Consider a channel with rate r in bits/second, and a round trip delay (*rtd*) in seconds. In the following, we will be concerned with the product of r and *rtd*, known as the bandwidth-delay product.

In this analysis, we assume a fixed frame size of s bytes (including protocol overhead). Therefore, the bandwidth-delay product can be expressed as $Tr = r(rtd)/8s$ frames worth of data. This means that there can be a maximum of Tr frames (counting partial frames) transmitted in one *rtd*.

In practice, the channel will cause frame losses due to either transmission errors or queue overflows. In the following, assume that frame losses occur randomly. This is a worst case assumption, since losses due to congestion will be correlated, allowing for a quicker protocol recovery ([28] illustrates that ARQ analysis with random errors yields a lower bound on throughput; upon inspection, this is also true for SSCOP). Let e and p be the end-to-end bit and frame error rates, where $p = 1-(1-e)^{8s}$ if the bit errors are random. The end-to-end bit error rate for digital systems based on ITU-T Recommendation G.826 will be better than 10^{-7} [29], [30], leading to frame loss rates corresponding to almost 10^{-3} for frames around 1 Kbyte. Similarly, if the frame loss rate (due to cell dropping in switch queues) is less than 10^{-3}, the analysis will still hold. Hereafter, all errors and losses are jointly referred to as "losses" from the SSCOP perspective.

A key protocol parameter is the frequency at which the transmitter sends a POLL message to request state information. This is assumed to occur regularly every *Timer_POLL* seconds. For constant frame sizes and a constant channel, *Timer_POLL* can be expressed in terms of frames (Tp) through the relationship $Tp = r(Timer\_POLL)/8s$.

Finally, the receiver grants the transmitter a window of W frames. The transmitter can have up to W (inclusive) frames outstanding in transit to the receiver . If $W = 0$, this means that the receiver has shut off the transmitter.

B. System assumptions

The following assumptions are used to simplify the analysis. They are loosely based on those found in the analysis of the SNR protocol given in [31]:

1) A discrete slotted time axis is used, in which time is divided into frame transmission opportunities of size $T = 8s/r$ seconds. Each frame takes T seconds to transmit; upon completion of a transmission, the transmitter may transmit an additional frame or must otherwise wait another T seconds until the next transmission opportunity. In the following, TP and TR denote the next highest integral values of Tp and Tr.

2) Although the protocol allows for full duplex operation, each direction of data flow is independent. Therefore, we only consider a unidirectional channel for data flow. A separate, lossless, bi-directional error-free channel is used for POLL and STAT transmissions. In practice, POLL and STAT traffic will be multiplexed with data traffic, and should be very low in volume compared with the data traffic for high speed connections. This assumption holds if the frame size is much larger than a single cell, and if the loss rate is low. In practice, the loss of a POLL or STAT message can be fully recovered by the subsequent message.

3) All retransmissions are successful. Again, as loss rates increase, this assumption becomes less valid, but this assumption should be valid for $p <= 10^{-3}$ [31].

4) All times are deterministic (although in practice, SSCOP is very tolerant of delay variations).

5) Initial recovery of each missing frame is stimulated by a USTAT message. If a frame is lost, it is assumed that the following frame is received correctly. This allows the transmission of a USTAT message immediately following the reception of the next valid frame.

6) The transmitter is saturated and always has a new frame to send.

7) Open loop flow control is assumed, with the window offered by the receiver (W) not more than the amount of receiver buffer available.

Figure 3 illustrates the model of this system, from the transmitter's perspective, assuming a slotted time axis, constant delay, and a saturated transmitter. Assume that the frame transmitted at time slot 1 is lost (denoted by "L"). The frame should have been received at time $1+Tr/2$. However, the receiver will not notice the loss until time $2+Tr/2$, when it receives the next frame. The USTAT will be immediately sent back to the transmitter, arriving (from the transmitter's perspective) at time $Tr+2$. Therefore, the retransmission can be transmitted at slot $TR+3$ (denoted by "R" in the figure). In the worst case (no further new losses), this retransmission must be acknowledged by a STAT message. A POLL will be transmitted at time $TR+3+X$, where X can vary uniformly from 0 to $TP-1$. The STAT will arrive

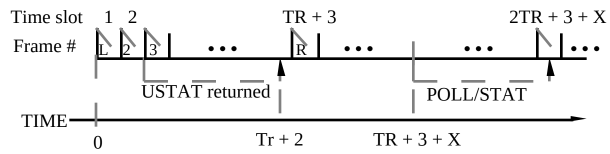

Fig. 3. Model of SSCOP operation

back at the transmitter by time slot $2TR+3+X$, at which time the retransmission will be acknowledged. Not shown in the figure are other POLLs, which are sent at times $TR+3+X-TP$, $TR+3+X+TP$, etc.

In this context, we can define the maximum throughput efficiency $\rho_{\max}$ as the number of frames that are successfully transmitted divided by the total number of frame slots available during some period of time NT (N slots of length T). An alternative measure of efficiency from the network perspective [31] is L_e, or the network link efficiency, which is defined as the fraction of frames (transmitted over the network) which the receiver can accept. We are primarily interested in $\rho_{\max}$, since L_e does not account for the forced idles that may occur due to a limited transmit window.

We now wish to examine the effect of varying the protocol parameters or the channel environment on $\rho_{\max}$. We have chosen to focus on throughput efficiency rather than the delay performance since the protocol parameter selection that maximizes throughput should also minimize average delay across a connection. The average delay is strongly influenced by the one way transmission delay and is only worsened significantly if the protocol cannot maintain a high throughput or if second order loss effects are severe.

C. *Approximate analysis*

The throughput efficiency obtainable with an SSCOP connection is dependent upon the channel environment and the protocol parameters. For a given environment (r, rtd, and e), we wish to obtain relationships between the throughput efficiency and the protocol parameters (W, s, and *Timer_POLL*). From an implementation perspective, we will want to maintain throughput efficiency while decreasing W (less receiver buffer requirements), increasing s (less channel overhead, less processing overhead), and increasing *Timer_POLL* (less channel overhead, less processing overhead). The following explores the effects of changing these protocol parameters.

In practice, the amount of receiver buffer that can be allocated to a connection will be limited. For a high speed protocol using selective retransmissions, the buffer capacity needed at the transmitter and receiver can become large. A reduction of window will not affect the link efficiency L_e, but will affect the throughput efficiency $\rho_{\max}$, since a reduced window may force a transmitter to idle. Consider again the model shown in Figure 3. In the worst case, the next POLL after the retransmission will occur $TP-1$ slots later. The STAT in response to such a worst case POLL will arrive at the transmitter by time $2TR+TP+2$. Assuming no other losses and a non-constraining window, the transmitter will be transmitting the frame numbered $2TR+TP+2$ in the next slot. The STAT will advance the window TP frames, allowing the transmitter to continue without idling even if the window is $2TR+TP+1$. Therefore, *if* $W>=2TR+TP+1$, *the transmitter will never idle due to window closure*. The throughput efficiency will be approximately $1-p$, as discussed above. If $W<=2TR+TP$, then certain loss scenarios will cause the transmitter to idle due to window closure.

Therefore, one window policy for obtaining a high throughput efficiency of SSCOP would be to simply allot the transmitter a window $W>=2TR+TP+1$ in all cases. However, in a broadband environment this window may become very large. Consequently, it is useful to explore other cases in which the window must be reduced, or in which the polling interval may be reduced to lower the required window size.

Of course, if the window is sufficiently small, the transmitter may idle even in the loss free case. In fact, regardless of the values of TP and TR, $W=TR+TP$ *is the minimal window required to keep the transmitter from idling periodically*.

Finally, we are interested in examining the window region for which $TR+TP<=W<=2TR+TP$; within this range the transmitter may or may not be forced to idle, depending on the pattern of losses. The behavior of SSCOP in this window region is different from that of HDLC based protocols, in which acknowledgments are sent more frequently. If $W>=2TR+TP+1$, the transmitter is never forced to idle. By close inspection of Figure 3, it can be seen that if W is one less than $2TR+TP+1$ (namely, $2TR+TP$), the transmitter will be forced to idle one time slot only if $X=TP-1$. If $W=2TR+TP-1$, the transmitter will be forced to idle two time slots if $X=TP-1$, and one if $X=TP-2$. In general, if $W=2TR+TP-j$, where $(0<=j<=TP-2)$, the transmitter will be forced to idle $\max[0, j+X+2-TP]$ time slots. When $W=2TR+2$, the only value of X that prevents idling is X = 0. Therefore, we have the relationship that if $2TR+2<=W<=2TR+TP$, the transmitter will idle only for certain values of X.

Furthermore, if $TP<=TR+1$ and $TR+TP<=W<=2TR+1$, the transmitter will always be forced to idle upon any loss, and the amount of time spent idling will depend upon X. Hence, *in the window region* $TR+TP<=W<=2TR+TP$, *the duration for which the transmitter is forced to idle is dependent upon the relationship between the frame loss and the POLL cycle.*

Therefore, the window W will fall into one of three categories:

1) $W>2TR+TP$; in this region, the transmitter never idles;

2) $TR+TP<=W<=2TR+TP$; in this region, the window can cause the transmitter to idle only if a loss occurs, and the duration of the idle depends upon the timing of the loss in relation to the POLL interval;

3) $W<TR+TP$; in this region, the window always limits throughput even in the loss free case.

In the following, approximate throughput relationships have been derived for the three different cases. In order to conserve space, only one case (case 2) is explained in detail; other relationships can be derived by applying similar procedures to the different cases.

Case 1: W > 2TR + TP (window does not constrain throughput)

If the transmit window W is very large (e.g., several times larger than $TP+TR$), the transmitter will never idle due to lack of window. In this case, since there are no unnecessary retransmissions for SSCOP:

$$\rho_{\max}=L_e\approx 1-p, \quad W>2TR+TP. \tag{1}$$

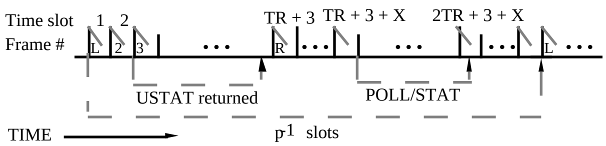

Fig. 4. Error cycle of SSCOP

This equation is equivalent to the general first order transmission efficiency equation detailed in [4]. In this case, an increase in s increases the throughput efficiency. However, it also increases the frame loss rate p.

Case 2: TR + TP <= W <= 2TR + TP (window constrains throughput only when losses occur)

In this window region, the window will only cause the transmitter to idle for certain loss scenarios. Following the approach used in [31], the analysis of regenerative cycles is applied using renewal theoretic arguments to describe the steady state performance. Again, assuming the loss rate is low (in this case, assuming that $p^{-1} > 2TR + TP$), the recovery of most losses will complete before the next loss occurs. We can then bound the cycle by beginning a new cycle upon the following event: 1) a loss occurs, and 2) the recovery from the previous loss has been completed (i.e., the protocol has returned to an equilibrium state).

Figure 4 illustrates the cycle. Suppose the frame transmitted at time slot 1 is lost. It will be retransmitted at time $TR+3$, and acknowledged by time $2TR+3+X$. The next frame loss will occur, on average, p^{-1} frames after time slot 0. This, however, will cause the end of the cycle and the beginning of the next. It is possible that the next error occurs during the recovery process (between time 1 and time $2TR+3+X$). However, such a scenario would be a superposition of regenerative cycles and can be ignored provided that $p^{-1} > 2TR + TP$; simulation will be used to check this assumption.

We will focus on counting the number of lost transmission slots that occur due to a frame loss. Each loss costs at least one slot (the retransmission occupies a slot). In addition, due to the relationship of the next POLL to the retransmission, additional slots may be unused if the transmitter is forced to idle.

For example, let the window be expressed as $W = 2TR + TP - j$, $0 <= j <= TR$. It was stated earlier that the number of time slots that the transmitter is forced to idle is $\max[0, j + X + 2 - TP]$. Since X is uniformly distributed over $[0, TP-1]$, the average number of time slots that are forced idle due to a loss can be expressed as $\frac{1}{TP}\sum_{k=0}^{TP-1}\max[0, j-k+1]$. This can be expressed in terms of W, as

$$\frac{1}{TP}\sum_{k=0}^{TP-1}\max[0, 2TR+TP-W-k+1].$$

As stated above, each cycle is started upon the occurrence of a loss. On average, there are p^{-1} time slots between transmission of lost frames. The number of successful transmissions during this time can be expressed as

$$p^{-1} - \left(1+\frac{1}{TP}\sum_{k=0}^{TP-1}\max[0, 2TR+TP-W-k+1]\right).$$

Therefore, the transmission efficiency $\rho_{\max}$ can be approximated as

$$\rho_{\max} = 1 - p\left(1+\frac{1}{TP}\sum_{k=0}^{TP-1}\max[0, 2TR+TP-W-k+1]\right),$$
$$TR + TP <= W <= 2TR + TP. \qquad (2)$$

The following observations are made about this relationship:

- as p decreases, $\rho_{\max}$ increases, as expected;
- as W increases, $\rho_{\max}$ increases, because increased window reduces idle time;
- as TR or TP increases, $\rho_{\max}$ decreases, because an increase in TR or TP increases the latency in the POLL/STAT exchange.

Equation (2) above should generally hold for low p, or more specifically, for $p^{-1} > 2TR + TP$.

Case 3: W < TR + TP (window always constrains throughput)

If $W < TR + TP$, then even when considering losses, the drop in transmission efficiency will be heavily dominated by the forced window closure (unless the losses occur more frequently than the POLL interval). Therefore, we will not consider the possibility of losses in this case.

In this window region, the behavior of the protocol is highly dependent upon the relative values of *TP* and *TR*. There are two main cases to consider: $TP >= TR$ and $TP < TR$.

a) $TP >= TR$ and $W < TR + TP$

Using a similar approach to the derivation of equation (2) above, the following relationships can be derived for the case $TP >= TR$ and $W < TR + TP$:

$$\rho_{\max} \approx \frac{W}{TP}, \ W <= TP - TR; \text{ and} \quad (3)$$

$$\rho_{\max} \approx \frac{W + TP - TR}{2TP}, \ TP - TR < W < TP + TR. \quad (4)$$

b) $TP < TR$ and $W < TR + TP$

This case is somewhat more complicated to evaluate, especially if *TR* is not a multiple of *TP*. However, we observe that:

- if *TR* is a multiple of *TP*, then *W* frames will be transmitted after every $TR + TP$ time slots. If *TR* is not a multiple of *TP*, *W* frames will still be transmitted *approximately* every $TR + TP$ time slots. Hence, an approximation for $\rho_{\max}$ is given as:

$$\rho_{\max} \approx \frac{W}{TR + TP}, \ W < TR + TP, \text{ and } TP < TR. \quad (5)$$

This equation will slightly underestimate $\rho_{\max}$ for $W > TP$.

It is observed that given insufficient window, it is advantageous to the transmitter to reduce *TP*, but the performance is bounded by W / TR. Intuitively, this is the latency limit of the channel.

A minor adjustment to equations (4) and (5) is needed to account for the change in assumptions between cases 2 and 3 (in the case 3 region, losses were disregarded since the regular forced idle dominated the throughput efficiency). For *W* just less than $TP + TR$, and for higher error rates (e.g., $p \approx 10^{-3}$), the throughput efficiency for *W* just a bit larger than $TP + TR$ (using equation (2)) is less than that for *W* just under than $TP + TR$ (using either equation (4) or (5)). This boundary condition can be accounted for when in the window region of case 3 by assigning $\rho_{\max}$ to the minimum of

- $\rho_{\max}(W = TP + TR)$, computed using equation (2), and

- $\rho_{\max}(W)$, computed using the actual window and equation (4) or (5) as appropriate.

Summary of analysis

Equations (1)-(5) above have been derived for throughput efficiency $\rho_{\max}$, depending on the relative values of *TP*, *TR*, and *W*: The equations can be used to select appropriate values of SSCOP protocol parameters (*W*, *s*, and *Timer_POLL*) on a connection by connection basis. These relationships should be valid for low loss rates. We will use simulation to examine performance at higher loss rates.

Two factors will contribute to protocol overhead in the steady state: frame overhead and POLL/STAT overhead. Each data frame has four bytes of overhead, and there is generally one POLL and one STAT (usually 20 bytes total) every *TP* time slots. To account for the frame overhead, the expressions for $\rho_{\max}$ may be multiplied by the factor $(s-4)/s$. For simplicity, we assumed that the actual data traffic dominates the POLL/STAT traffic (for low speed operation, this may not be the case).

V. SIMULATION RESULTS

Simulation was used to verify the above analysis. The data transfer procedures of SSCOP have been simulated in a Simscript simulation environment, and the analytical results from above have been compared to the simulation results. The simulation consists of a source traffic generator which, for these simulations, attempts to saturate the link; an SSCOP protocol engine at the transmitter; a full duplex data link layer protocol responsible for framing and error detection; a data channel with configurable bandwidth, static propagation delay, and random error generator; an SSCOP receiver; and a traffic sink. The POLLs and data frames share the link in the transmit direction, and the STATs and USTATs occupy the link in the reverse direction. The simulator injects bit errors into the data channel, corresponding to a specified bit error ratio. The throughput calculated in the simulation accounts for frame overhead and for POLL/STAT/USTAT overhead.

A. *Comparison with analytical results*

In the following figures, three independent simulation runs have been averaged for each set of parameters. Random bit errors occurring according to a binomial distribution were inserted to the data link, affecting both data and control frames. The length of each simulation run was adjusted such that it captured approximately 100 frame error events. Variable round-trip delays and correlated frame losses have not been examined.

Figure 5 illustrates plots of the maximum throughput versus the SSCOP window, for different values of *Timer_POLL*. The channel assumed has a link at rate 100 Mb/s with a round trip delay of 10 ms. The frame size was a constant 1024 bytes, yielding frame loss ratios of approximately 10^{-3} and 10^{-4} for the BERs of 10^{-7} and 10^{-8}, respectively.

The range of window sizes plotted is based on the range of $0.1(TR+TP)$ to $2(2TR+TP)$, for each respective value of *TP*. Recall that $W = TR + TP$ is the minimum window required to keep the transmitter from idling in the error free case. An arrow for each curve marks the point at which $W = TR + TP$ for each different value of *TP*.

While these graphs are only representative of a wide range of possible link environments and protocol configurations, they do illustrate some general performance aspects of SSCOP:

- SSCOP can offer near optimal throughput performance over a wide range of environments, provided that the window is sufficiently large. At a minimum, the window $W = TR + TP$ is needed to keep the transmitter from idling. As the error rates degrade, larger windows are needed.

- Default data link window sizes have traditionally been based on roughly twice the bandwidth-delay product in frames ($2TR$). The graphs illustrate that, if such guidelines are followed for SSCOP, insufficient window sizes could be used. This is because the window needed is based also on *Timer_POLL*, which may be significantly larger than the round trip delay.

- Performance can be relatively insensitive to the value of *Timer_POLL*. For sufficiently large window sizes, *Timer_POLL* can be extended out to 1 second without severe penalty. One penalty for running a very long *Timer_POLL* is that the maximum frame delay will be increased, due to second order effects (multiple retransmissions). At low loss rates, this should not be a significant problem.

- In Figure 5, when *Timer_POLL* is extended to 500 ms and the BER is 10^{-7}, the analysis yields lower throughput estimates than those obtained with simulation. This effect is caused by the frame error rate being higher than the POLL interval, which violates the assumption that $p^{-1} > 2TR + TP$. On average, several losses are occurring each POLL interval. Since the missing frames trigger USTAT messages that will also advance the window, the transmitter receives more continuous feedback from the receiver and is able to transmit more. In effect, a missing frame is treated almost the same as a POLL, causing a retransmission but also advancing the window. Therefore, for low window values and large POLL intervals, frame losses can actually improve the throughput efficiency by triggering USTAT messages that advance the window. In practice, it is not recommended that the POLL interval (*TP*) be selected such that $p^{-1} < 2TR + TP$.

The approximate analysis is in good agreement with the simulation results. In particular, the throughput expressions derived earlier are within a few percent provided that $p^{-1} > 2TR + TP$. It is pointed out in [31] that an approximate analysis is particularly useful when loss rates are low, since the analysis is more accurate at lower loss rates and since statistically significant simulation runs at low loss rates are very long.

B. High error or loss rates

The analytical results are valid for environments with $p <= 10^{-3}$ and with $p^{-1} > 2TR + TP$; at higher frame loss rates, the analytical results diverge from the actual performance, since some assumptions break down. However, SSCOP, because it uses selective retransmission for error recovery, can provide high throughput even in a degraded environment.

In Figure 6, simulation results are presented for a hypothetical link on a very lossy channel ($p \approx 10^{-2}$). Again, three independent simulation runs have been averaged, and the duration of each run was 5 seconds

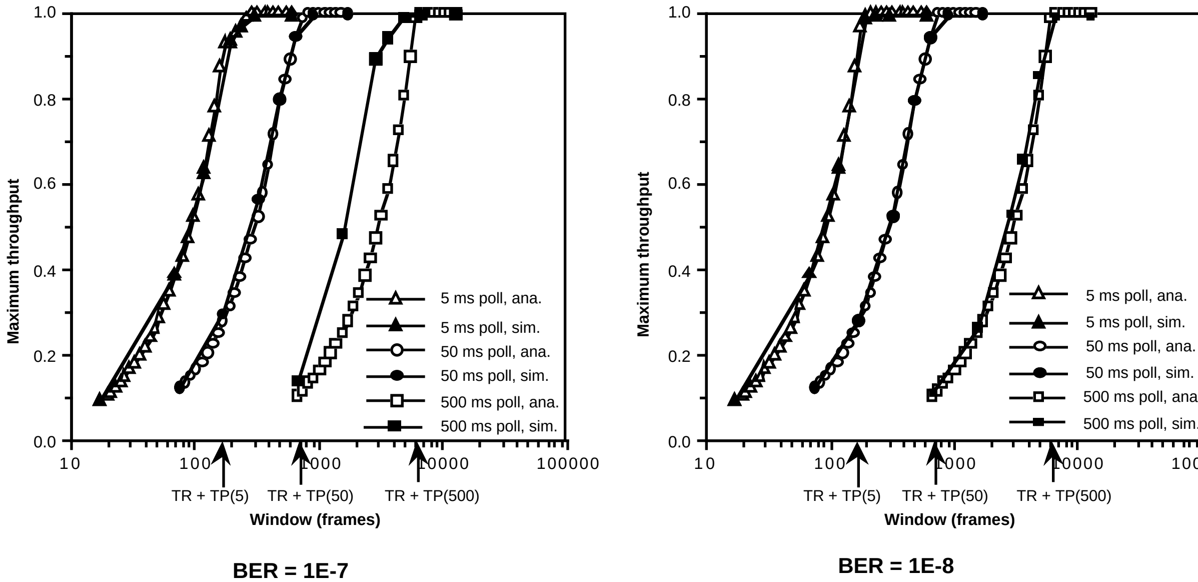

Fig. 5. SSCOP maximum throughput vs. window size (100 Mbit/s link, rtd = 10 ms, and frame size = 1024 bytes, and POLL intervals of 5, 50, and 500 ms)

(approximately 2500 frame errors). The frame window was varied while the *Timer_POLL* (2 ms) and the frame size (128 bytes) were held constant. In addition, to highlight the impact of the USTAT on protocol performance, a hypothetical modified SSCOP (labeled "SSCOP without USTAT"), in which USTAT message transmission is disabled, is compared to SSCOP. High throughput efficiency is possible in this environment, but the performance degrades more quickly when the window is reduced. In Figure 5, a high throughput (often greater than 0.8) was achievable when the window was reduced to $W = TR + TP$, but in this case, the throughput at that window size was only approximately 0.5. The USTAT mechanism allows the throughput to be maintained longer when the window is reduced. The difference between the two protocols (with or without USTAT) becomes more pronounced as the POLL interval is increased.

Although these results do illustrate that inappropriate parameter settings can cause throughput degradation, they also illustrate that the performance of SSCOP can be quite high in poor error environments or in networks that experience severe congestion losses. This performance can be largely attributed to SSCOP's selective retransmission ARQ mechanism, which significantly outperforms go-back-N in a lossy, high bandwidth-delay product environment.

VI. CONCLUSIONS AND FUTURE DIRECTIONS

SSCOP has been designed as a high speed protocol that can offer high throughput efficiency in a broadband or lossy environment. The main function of the protocol is to correct for lost or errored frames through retransmission. The protocol uses a selective retransmission strategy which offers optimal efficiency performance in the broadband environment. A number of other high speed design features were also incorporated.

SSCOP was originally intended to provide end-to-end error recovery for user data and link protection for ATM signaling. Since that time, ATM has evolved from being primarily a carrier backbone technology to being an IP subnet technology, and since SSCOP cannot operate over connectionless IP, there has been less interest in using it for user data transfer. SSCOP will see widespread use for UNI and NNI signaling. As for data transfer, SSCOP may find use in niche ATM environments and applications, particularly if connectionless extensions and rate-based flow control are added.

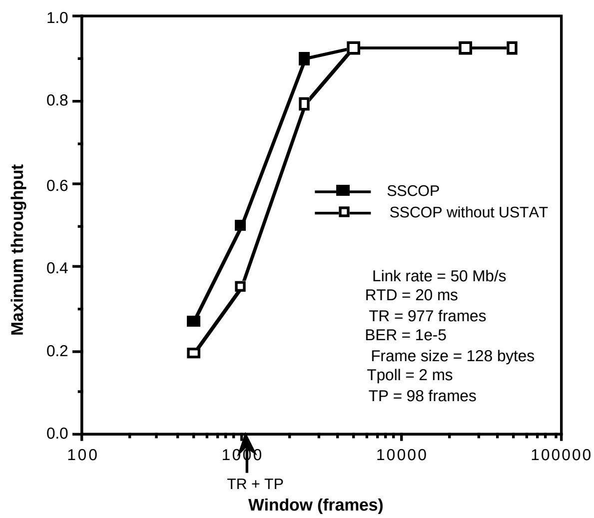

Fig. 6. Effect of different transmit windows in a degraded environment

Analytical results presented in Section IV have been shown to be accurate provided that the frame loss rate $p <= 10^{-3}$ and $p^{-1} > 2TR + TP$. At higher frame loss rates, the use of simulation was used to evaluate the achievable throughput. At low loss rates, the expressions derived herein can be used to select sufficient protocol parameters; most notably, to select a sufficiently large resequencing buffer at the receiver and a sufficiently small POLL interval at the transmitter.

The use of selective retransmission has the drawback of requiring the receiver to allocate a resequencing buffer at the receiver, in addition to the retransmission buffer at the transmitter that is inherently necessary in such a protocol. In practice, the amount of buffer available will be a limiting factor. However, the results of this paper confirm the results of [6] which illustrate that sufficient receiver buffer size is one of the key requirements for efficient operation of a selective retransmission protocol. When a satisfactory resequencing buffer is available, the protocol can offer adequate performance even in very degraded environments.

Some general results on how to configure an SSCOP connection are observed:

- to avoid constraining throughput, a resequencing buffer of at least size $W > TR + TP$ is needed at the receiver. In most environments, windows sufficiently larger than $W = 2TR + TP$ will not offer much incremental benefit.

- SSCOP is relatively insensitive to increases in the POLL interval, provided that sufficient window is available. The SSCOP transmitter, with knowledge of the round trip delay, can determine a sufficiently short POLL interval to maintain high efficiency. The POLL interval should be shorter than the frame loss interval p^{-1}. Furthermore, for applications which require a short frame delay (such as signaling), the POLL interval should be small to protect against second order loss effects.

Further study in the following areas is recommended:

- addition of rate-based flow control is one of the most important areas requiring development. In an ATM environment, SSCOP can potentially incorporate flow controls tied to ATM level traffic and congestion control mechanisms;

- with modifications to the elements of procedure pertaining to in-sequence frame reception and redundant frame retransmission, SSCOP could be modified to run on top of a connectionless service;

- very low speed operation of SSCOP (such as UNI signaling), particularly its impact on the connection control timer, requires further analysis; and

- it has been pointed out (e.g., [8] and [25]) that operating system overhead is a bottleneck in realizing the benefits of high speed transport protocols. To fully capitalize on SSCOP's reduction in front-end protocol processing, further advances are needed in reducing operating system bottlenecks.

ACKNOWLEDGMENTS

The author is indebted to A. Agarwal for developing the simulation model used herein, and to D. Chitre for helpful discussions and comments. The author would also like to acknowledge the numerous contributions from several international organizations that helped to rapidly develop and complete the SSCOP protocol specification.

REFERENCES

[1] W. Doeringer, D. Dykeman, M. Kaiserswerth, B. Meister, H. Rudin, and R. Williamson, "A Survey of Light-Weight Transport Protocols for High-Speed Networks,"

IEEE Trans. Commun., Vol. 38, No. 11, pp. 2025-2039, Nov. 1990.

[2] U. Black, Data Link Protocols, Englewood, NJ: Prentice-Hall, Inc., 1993.

[3] ISO/IEC 7809, Information technology-Telecommunications and information exchange between systems - High-level Data Link Control (HDLC) procedures - Classes of procedures, latest version approved in 1993.

[4] A. Gatfield, "Error Control on Satellite Channels using ARQ Techniques," COMSAT Technical Review, Vol. 6, No. 1, pp. 179-187, Spring 1976.

[5] A. Kaul, "Performance of High-level Data Link Control in satellite communications," COMSAT Technical Review, Vol. 8, No. 1, pp. 41-87, Spring 1978.

[6] D. Chitre, "Analysis of throughput efficiency and delay in ARQ systems," COMSAT Technical Review, Vol. 11, No. 2, pp. 345-367, Fall 1981.

[7] D. Clark, M. Lambert, L. Zhang, "NETBLT: A Bulk Data Transfer Protocol; RFC-998," Internet Request for Comments, No. 998, Network Information Center, March 1987.

[8] A. Netravali, W. Roome, and K. Sabnani, "Design and Implementation of a High-Speed Transport Protocol," IEEE Trans. Commun., Vol. 38, No. 11, pp. 2010-2023, Nov. 1990.

[9] B. T. Doshi and P. K. Johri, "Communication protocols for high speed packet networks," Computer Networks and ISDN Systems, Vol. 24, pp. 243-273, 1992.

[10] L. Kleinrock, "The Latency/Bandwidth Tradeoff in Gigabit Networks," IEEE Communications Magazine, Vol. 30, No. 4, pp. 36-40, April 1992.

[11] ITU-T Recommendation Q.2110, "B-ISDN-ATM Adaptation Layer- Service Specific Connection Oriented Protocol (SSCOP)," COM-11-R30, Geneva, Switzerland, 1994.

[12] J. Navai, "Proposed Convergence Sub-layer for VBR-AAL," T1S1.5/90-255, October 25, 1990.

[13] C. Flores and J. Swenson, "A proposed convergence protocol to serve class C and D users," T1S1.5/90-271R1, January 7, 1991.

[14] S. Jarrar and A. Agarwal, "Elements of procedure for AAL Type 4 assured operation with selective retransmission," T1S1.5/91-045, January 7, 1991.

[15] ITU-T Recommendation Q.921, "ISDN User-Network Interface- Data Link Layer Specification," Geneva, Switzerland, latest version approved in 1993.

[16] ITU-T Recommendation Q.703, "Signalling Link," Geneva, Switzerland, latest version approved in 1993.

[17] ITU-T Recommendation Q.2130, "B-ISDN-Signaling ATM Adaptation Layer- Service Specific Coordination Function (SSCF) for support of signaling at the user-to-network interface (SSCF at UNI)," COM-11-R31, Geneva, Switzerland, 1994.

[18] Draft ITU-T Recommendation Q.2140, "B-ISDN-Signaling ATM Adaptation Layer- Service Specific Coordination Function (SSCF) for signaling at the network node interface (SSCF at NNI)," COM-11-R51, Geneva, Switzerland, 1994.

[19] Draft ITU-T Recommendation I.365.2, "B-ISDN-ATM Adaptation Layer Service Specific Coordination Function to provide CONS (SSCF-CONS)," COM-13-R17, Geneva, Switzerland, 1994.

[20] Draft ITU-T Recommendation I.365.3, "B-ISDN-ATM Adaptation Layer Service Specific Coordination Function to provide COTS (SSCF-COTS)," COM-13-R17, Geneva, Switzerland, 1994.

[21] S. Nanda, R. Ejzak, and B. Doshi, "A Retransmission Scheme for Circuit-Mode Data on Wireless Links," IEEE J. Select. Areas Commun., Vol. 12, No. 8, pp. 1338-1352, October 1994.

[22] V. Jacobson, R. Braden, and D. Borman, "TCP Extensions for High Performance; RFC 1323," Internet Request for Comments, No. 1323, Network Information Center, May 1992.

[23] C. Partridge, Gigabit Networking, Addison Wesley, 1994.

[24] A. Fraser and W. Marshall, "Data transport in a byte-stream network," IEEE J. Select. Areas Commun., Vol. SAC-7, pp. 1020-1033., September 1989.

[25] G. Lundy and H. Tipici, "Specification and Analysis of the SNR High-Speed Transport Protocol," IEEE Trans. Networking, Vol. 2, No. 5, pp. 483-496, October 1994.

[26] V. Jacobson, "Congestion Avoidance and Control," Proc. ACM SIGCOMM '88, Stanford, CA, August 1988, pp. 314-329.

[27] M. Anagnostou and E. Protonotarios, "Performance Analysis of the Selective Repeat ARQ Protocol," IEEE Trans. Commun., Vol. COM-34, No. 2, pp. 127-135, Feb. 1986.

[28] S. Kim and C. Un, "Throughput Analysis for Two ARQ Schemes Using Combined Transition Matrix," IEEE Trans. Commun., Vol. 40, No. 11, pp. 1679-1683, Nov. 1992.

[29] M. Shafi and P. Smith, "The Impact of G.826," IEEE Communications Magazine, Vol. 31, No. 9, pp. 56-62, September, 1993.

[30] ITU-T Recommendation G.826, "Error Performance Parameters and Objectives for International Constant Bit Rate Digital Paths At or Above the Primary Rate," Geneva, Switzerland, latest version approved in 1993.

[31] B. Doshi, P. Johri, A. Netravali, and K. Sabnani, "Error and Flow Control Performance of a High Speed Protocol," IEEE Trans. Commun,, Vol. 41, No. 5, pp. 707-719, May 1993.