HEWLETT PACKARD

An Improved SSCOP-Like Scheme for Avoiding Unnecessary Retransmissions and Achieving Ideal Throughput

Reuven Cohen, HP Israel Science Center*
HPL-95-138
December, 1995

signaling ATM, SSCOP, ideal throughput, retransmission policy

SSCOP is a new data link control protocol designed for ATM networks. To achieve high throughput, SSCOP uses the selective repeat retransmission policy, enforced using the exchange of POLL and STAT control frames between the sender and the receiver. To avoid unnecessary retransmission of information frames (I-frames), SSCOP uses the "checkpoint concept" where every POLL has a sequence number and every outstanding I-frame is associated in the sender buffers with the sequence number of the POLL sent before the last transmission of that I-frame. Using these sequence numbers, the sender knows to ignore unnecessary retransmission requests. The paper proposes an improved scheme for avoiding unnecessary retransmission of I-frames. The new scheme significantly reduces the memory needed for storing control information of the protocol at the sender interface. Compared with the SSCOP, the only potential drawback of the proposed scheme is that under certain circumstances the retransmission of lost I-frames may be delayed. However, the paper analyzes the sequences of events that must take place in such a case, and concludes that the probability for such a sequence is negligible and that the new protocol performs as well as the SSCOP.

*Technion City, Haifa 32000, Israel

1 Introduction

One of the main roles of the Data Link Control (DLC) layer is to provide an error-free logical link connection between two entities communicating over a physical or logical channel. Error control schemes append a frame check sequence and a sequence number to each frame, and provide a set of rules to let the receiver determine when a frame is lost, and to let the sender determine when a frame should be retransmitted. Collectively, these schemes are referred to as Automatic Repeat reQuest (ARQ). The most common ARQ scheme is go-back-N. Go-back-N is used by HDLC [4, 12] – the most popular DLC protocol – and by its many variations (like the LLC [10], LAP-D etc.). Simple performance analysis shows that when the sender is equipped with enough buffer space, go-back-N algorithm yields a throughput of

$$T = \frac{1-\alpha}{1+\mathrm{RTT}\alpha} \tag{1}$$

where α is the loss probability of a single Information frame (I-frame) and RTT is the Round Trip Time which consists of the propagation delay and the buffering delay in the end node and in the intermediate nodes, divided by the transmission time of a single frame. The number N of buffers needed at the sender in order to achieve this throughput depends on the transmission policy of ACK and NAK frames, but in any case must be larger than RTT.

As Eq. (1) indicates, the throughput of go-back-N decreases when RTT increases, due to the increased number of information frames needed to be retransmitted after a loss of a single frame. A large RTT can be found not only in satellite networks, but also in high-speed terrestrial networks, because of the decrease in frame transmission time and the increase in the end-to-end delay due to queuing at the intermediate switches. For such networks, the selective-repeat protocol [1, 9, 11, 14], which retransmits only lost and erroneous packets[1], is markedly superior to go-back-N. In selective-repeat, the receiver must contain enough buffer space to save out-of-order received I-frames before releasing them to the higher layer, until the missing frames are retransmitted. If the receiver window and the sender window are big enough, selective-repeat may yield the *ideal throughput*

$$T = 1-\alpha \tag{2}$$

which is the probability that an I-frame is correctly received.

However, even if the sender and the receiver are equipped with sufficient buffer space, achieving the ideal throughput is not a trivial task. An efficient and reliable mechanism is needed in order to let the sender know which I-frames should be retransmitted due to a loss of their last transmitted copy. A loss of an I-frame is usually detected by the receiver when an I-frame is received whose sequence number is by more than 1 greater than the sequence number of the last received I-frame. The receiver can inform the sender about the sequence numbers of lost I-frames by means status (STAT) frames. To avoid livelock conditions due to loss of STAT frames, the STAT frames should be sent periodically, rather than only when a new gap is detected, and should contain accumulated information. That is, every STAT notifies the sender about all the missing I-frames, including those reported by previous STATs. However, this may lead to unnecessary retransmission of I-frames, which was the reason for using selective-repeat rather than go-back-N in the first place, and to significant performance loss. (Note, however, that the reason for unnecessary retransmissions in selective-repeat is different from the reason for unnecessary retransmissions in go-back-N. In the former case

[1] Throughout the paper erroneous packets are considered as lost packets.

the I-frames are received and accepted by the receiver, but the sender thinks they have lost due to the receipt of a STAT sent by the receiver before having received the I-frames. In the latter case the frames are received but rejected by the receiver).

A selective-repeat scheme, called Checkpoint Mode (CPM), that avoids unnecessary retransmission has been presented in [3, 5, 8]. According to this scheme, the STAT (called CP in the original proposal) periodically sent by the receiver contains the sequence number of the most recently received new I-frame. For every outstanding I-frame (i.e. an I-frame for which the sender expects to receive an acknowledgment), I-frame(i) say, the sending station keeps in a table the sequence number of the next I-frame to be transmitted in the first time when the last transmission of I-frame(i) takes place. By comparing these numbers to the number included in the STAT frame, the sender can determine which of the I-frames requested by the sender should indeed be retransmitted, and which I-frames should not be retransmitted because their last copy might have been correctly received by the receiver *after* the STAT was sent. Using this scheme, unnecessary retransmission of I-frames is avoided, and for certain window sizes the ideal throughput of $(1 - \alpha)$ can be achieved [3].

Service Specific Connection Oriented Protocol (SSCOP) [13] is a new peer-to-peer protocol designed for the B-ISDN ATM Adaptation Layer 5 (AAL-5). It provides for traditional data link control functionalities, as transfer of user data with sequence integrity, error correction by retransmission, flow control and connection control, between two users communicating over an ATM Virtual Channel (VC) connection. To achieve high throughput in high-speed networks, the SSCOP retransmission scheme in the assured data transfer mode is based on selective-repeat. To avoid unnecessary retransmission of I-frames, SSCOP uses concepts adopted from the Checkpoint Mode scheme. The SSCOP sender periodically polls the receiver to determine if there is a gap in the sequence of successfully received frames. The POLL frames are sequentially numbered, independently of the I-frames. The sender uses a local table in order to associate with every I-frame the sequence number of the POLL sent before the last transmission of that I-frame. Every STAT contains the sequence number of the POLL to which it responds. By comparing the sequence number of the STAT to the sequence number of the last sent POLL associated with every I-frame for which a retransmission is requested by the STAT, the sender can determine whether to retransmit the I-frame or to ignore the request. More specifically, if the POLL sequence number associated with the last transmission of the I-frame is *larger* than the POLL sequence number to which the received STAT responds, the sender deduces that the STAT has been sent as a response to a POLL which was sent before the last transmission of the I-frame, so the retransmission request should be ignored.

This paper presents an improved scheme for achieving the SSCOP performance. The main differences between the proposed scheme and the SSCOP are that the POLL messages are associated with a binary flag, that takes the place of the 24-bit sequence number, and that the sender does not need to associate the 24-bit sequence number of the last sent POLL with every outstanding I-frame. In the new scheme, when the sender receives a STAT frame it compares the flag of the STAT with a representation of this flag in its memory. If both flags are equal, the STAT is accepted and *all* its retransmission requests are fulfilled. If, however, the two flags are different, the sender ignores all the retransmission requests because some of them *might* be unnecessary.

The main advantages of the proposed simplified scheme are as follows. First, the new scheme reduces the amount of memory needed for storing control information of the protocol. Experiences in high-speed network interface design (e.g. [2, 7]) show that in order to reduce

memory copy cost, it is necessary to distinguish between control information, needed for the execution of the protocol by the protocol processor, and the data needed to be transmitted. The control information should be stored as close to the protocol processor as possible, i.e. in the primary cache, whereas the data (I-frames) can be stored in regular memory and moved to the network by means of Direct Memory Access (DMA). In terms of the SSCOP, the control information is mainly the vector that contains the association between the sequence number of every outstanding I-frame and the sequence number of the POLL sent before the last transmission of that I-frame. The POLL sequence number is 24-bit long. As shown later, and also discussed in [3], in order to guarantee high throughput, the number of outstanding I-frames the sender should be able to accommodate with, and therefore the length of the sender vector, might be some hundreds or even thousands. This sums up to high-speed memory of several K-Bytes per *every* SSCOP connection, which is not required by the new scheme. As every interface may have to manage many SSCOP connections at any given time, the cache memory savings of the new scheme is significant. Another advantage of the new scheme is that the processing burden on the sender is reduced, and that the recovery from certain failure conditions is enhanced. These issues will be discussed in Section 3, after the new scheme is presented.

The only *potential* drawback of the new scheme compared with the SSCOP is that under certain conditions the sender may ignore relevant retransmission requests. However, the paper analyzes the probability for these conditions and shows that it is very small, and can be ignored. Simulation results approve this conclusion by showing that the SSCOP and the new scheme have the same throughput. In particular, both schemes can achieve the ideal throughput for the same window sizes.

The rest of the paper is organized as follows. Section 2 describes the SSCOP approach for avoiding unnecessary retransmissions. Section 3 presents the new scheme and discusses its advantages and its potential disadvantage. Section 4 presents an analysis of the new scheme, explaining why the new scheme performs as well as the SSCOP. Section 5 concludes the paper.

2 SSCOP Scheme for Avoiding Unnecessary Retransmissions

Service Specific Connection Oriented Protocol (SSCOP) [13] is a new peer-to-peer protocol designed for the B-ISDN ATM Adaptation Layer 5 (AAL-5). As specified in [13], the most important functions performed by the SSCOP are:

- Sequence integrity of data.
- Error correction by retransmission.
- Flow control, allowing the receiver to control the rate at which the peer sender sends information.
- Error reporting to layer management.
- Connection control, including establishment, release, resynchronizations and keeping alive of SSCOP connections.

The following section concentrates upon the selective-repeat ARQ mechanism employed by SSCOP. Since we are concerned with main concepts only, many irrelevant details are omitted and only three types of control frames are considered: (1) POLL frames, periodically sent by the sender; (2) STAT frames, sent by the receiver as a response to the received POLL frames; and (3) USTAT frames, sent by the receiver whenever a new gap in the sequence numbers of the received frames is detected. Two mechanisms are recommended for stimulating the transmission of POLL frames by the sender: POLL timer and POLL counter. Both mechanism can operate simultaneously. In what follows we assume, without loss of generality, that a POLL is sent by the sender every τ time units. Thus, if the sender is not idle, it sends τ I-frames between the transmission of two consecutive STAT frames.

The SSCOP sender manages two 24-bit sequence number counters. The first counter, referred in this paper to as *I-frame_counter*, is for the sequence numbers associated with the I-frames. The second counter, referred to as *POLL_counter,* is for the sequence numbers associated with the POLL frames. Whenever the sender window is not full, a new packet can be received from the upper layer and a new I-frame is created, to which the value of I-frame_counter incremented by 1 (modulo 2^{24}) is assigned. As long as the sender window contains I-frames which have not been transmitted in the first time, and no retransmission is requested by the receiver, the sender keeps sending I-frames in sequence.

Every τ time units, a new POLL frame is created by the sender and transmitted to the receiver. A POLL frame has a *sequence_number* field[2] to which the current value of *POLL_counter,* incremented by 1 (modulo 2^{24}) is assigned. Another field of the POLL is *last_sent_I-frame*, to which the value of *I-frame_counter* is assigned. Whenever an I-frame is transmitted, I-frame(i) say, a control vector, referred to as *last_transmission_time*[], is updated such that entry i is assigned the sequence number of the last sent POLL (*POLL_counter*).

As an example, suppose that the width of the sender's window is 12, which means that the sender may have up to 12 outstanding I-frames, and consider the frame exchange depicted in Figure 1. Figure 1(a) shows the state of the *last_transmission_time*[] vector after 10 I-frames and 2 POLL frames are sent, assuming that a POLL is sent every 4 I-frames (i.e. $\tau = 4$). Note that for the first POLL, POLL(1, 4), POLL.*sequence_number*=1 and POLL.*last_sent_I-frame*=4. When the receiver receives a POLL, it responds with a STAT frame. The purpose of the STAT is to let the sender know which of the I-frames have been correctly received and which of them have been lost and should be retransmitted. The STAT has three fields: STAT.*sequence_number*, which is copied from POLL.*sequence_number*; STAT.*last_sent_I-frame* which is copied from POLL.*last_sent_I-frame*; and STAT.*not_received_I-frames*, which contains a list of the I-frames the receiver is missing up to POLL.*last_sent_I-frame*. Figure 1(b) shows the state of the *last_transmission_time*[] vector after the first STAT is received and *processed* (namely, after the retransmission resulting from this STAT take place). When the receiver receives this STAT, it compares the values of *last_transmission_time*[2] and *last_transmission_time*[3] (both are 0) to STAT.*sequence_number* (1). The fact that the value of STAT.*sequence_number* is larger indicates that the last transmission of the lost I-frames was performed *before* the transmission of the POLL to which the received STAT responds. This tells the sender that the missing I-frames should indeed be retransmitted. After processing the STAT, the sender retransmits the missing two I-frames and updates *last_transmission_time*[2] and *last_transmission_time*[3] to *POLL_counter* (2). In addition, since this STAT serves also as a positive acknowledgment for I-frame(1), the receiver advances the sending window one step. After retransmitting I-frame(2) and I-frame(3), the

[2] Instead of adhering to the original field names, used in [13], we shall use more convenient and self-explanatory names.

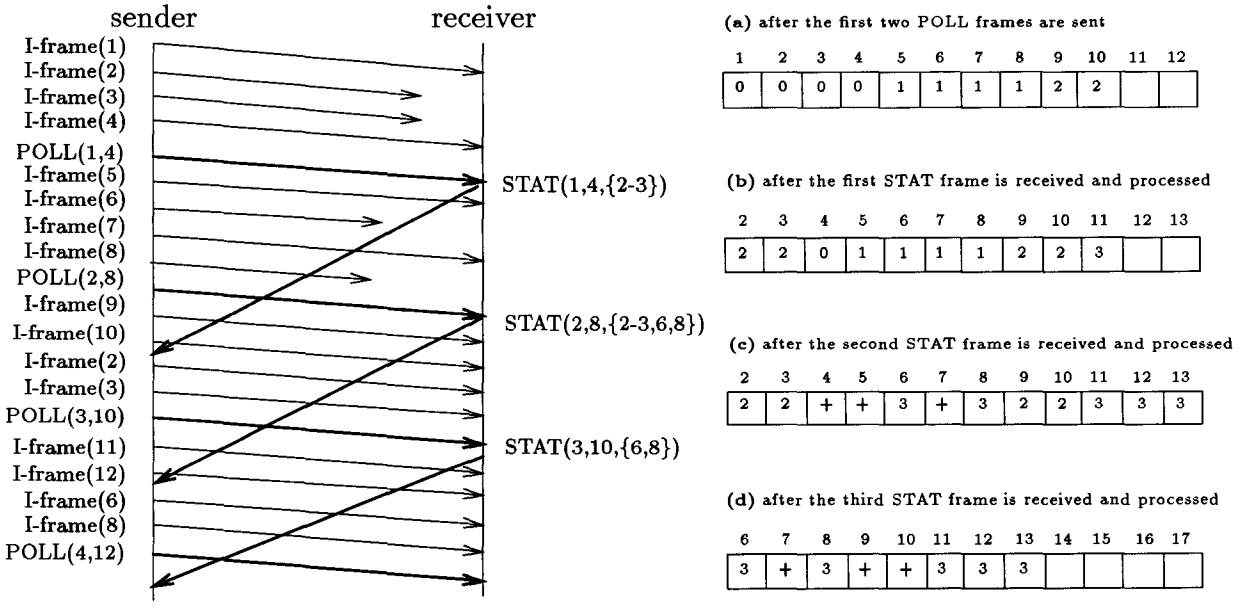

Figure 1: An SSCOP Example (without USTAT frames)

sender sends a new POLL, with *sequence_number*=3 and *last_sent_I-frame*=10. Then, I-frame(11) can be sent.

Figure 1(c) shows how SSCOP avoids unnecessary retransmission of I-frames. When the receiver receives the second POLL, it responds with STAT(2,8,{2-3,6,8}). This STAT tells the sender that I-frames 4, 5 and 7 have been correctly received by the sender (thus, entries 4, 5 and 7 in *last_transmission_time*[] change to "+"), and that I-frames 2, 3, 6 and 8 are missing. However, since *last_transmission_time*[2] and *last_transmission_time*[3] are not smaller than POLL.*sequence_number* (they all equal 2), the sender deduces that the POLL to which the received STAT responds was sent *before* the last transmission of I-frame(2) and I-frame(3). Therefore, this STAT cannot indicate whether the last transmission of these two I-frames was successful or not, and there is no need to retransmit these two I-frames. The condition of I-frame(6) and I-frame(8) is different, however, since their *last_transmission_time*[] (1) is smaller than POLL.*sequence_number* (2). Thus, the sender retransmits only these two I-frames, and changes entries 6 and 8 in *last_transmission_time*[] from 1 to 3. Since instead of performing unnecessary retransmission of I-frame(2) and I-frame(3) the sender can keep sending new I-frames from its window, the throughput of the protocol increases. Figure 1(d) concludes the example by showing the *last_transmission_time*[] vector after the third STAT is received and processed. This STAT contains an acknowledgment for I-frames 2, 3, 9 and 10, which causes the sender to advance its window four steps, and a retransmission request for I-frame(6) and I-frame(8), which is ignored.

Since the width of the sender window is bounded, it is important to reduce as much as possible the period of time elapsed between the time the receiver realizes that an I-frame is lost and the time this I-frame is retransmitted by the sender. If only POLL and STAT

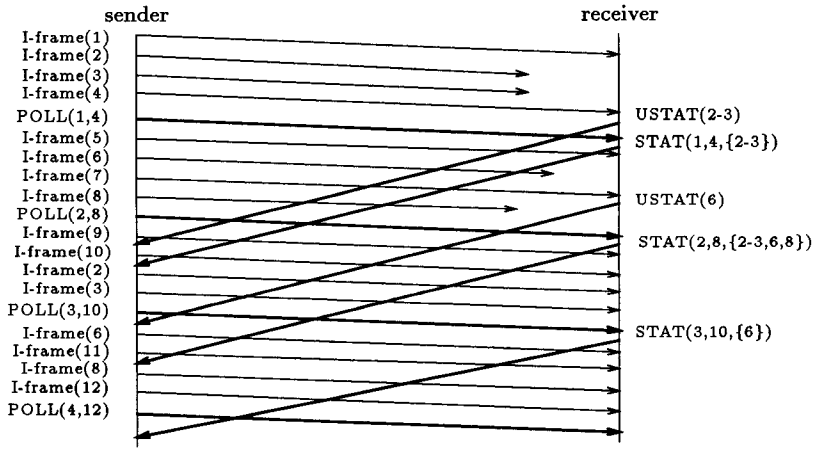

Figure 2: An SSCOP Example (With USTAT frames)

messages are used, this time is bounded by $(\tau + \text{RTT})$. However, in order to decrease the upper bound on this time to the minimum (RTT), SSCOP uses a third type of control frames, called USTAT (unsolicited STAT). Whenever the receiver detects a *new* gap in the sequence number space of the received I-frames, it informs the sender by means of a USTAT frame. The USTAT frame also acknowledges the receipt of all the I-frames until the *first* gap (which is not necessarily the *new* gap reported by the USTAT). If a new gap is detected in the first time upon the receipt of a POLL frame, in which case the responding STAT reports the gap, the receiver sends no USTAT frame.

Figure 2 is a modification of Figure 1 where USTAT frames are used. When the receiver receives I-frame(4) it realizes that I-frame(2) and I-frame(3) got lost. Thus, it immediately sends to the sender a USTAT$(1, 2-3)$ frame. In this particular case the USTAT yields no profit because immediately after receiving I-frame(4), a POLL frame is received and a STAT is sent back to the sender. Another USTAT, reporting the loss of I-frame(6), is sent by the receiver after the receipt of I-frame(7). This USTAT causes the sender to retransmit I-frame(6) two time units before the retransmission in Figure 1. Since the loss of I-frame(8) is discovered by the receiver when the POLL$(2, 8)$ is received, rather than when I-frame(9) is received, the receiver sends no USTAT for this I-frame. When the sender receives USTAT and retransmits I-frames, it updates the *last_transmission_time* vector. Hence, the USTAT frames do not lead to unnecessary retransmissions. In Figure 2, for instance, the sender does not retransmit any I-frame when a STAT is received.

SSCOP allows the receiver to send a sequence of multiple copies of the same USTAT frame whenever a new gap is detected, in order to increase the probability that at least one USTAT is received by the sender (although even if all the USTAT frames are lost, the gap will be reported and recovered using subsequent POLL/STAT handshakes). To avoid unnecessary retransmission due to the use of multiple USTAT frames, the sender uses a binary vector called *retransmitted*[]. This vector has the same role as the *last_transmission_time*[] vector, but it is effective for USTAT frames, rather than for STAT frames. This vector is initialized to 0. When some I-frame, I-frame(i) say, is retransmitted the first time, *retransmitted*[i] is set to 1. If a USTAT is received for I-frame(i) while *retransmitted*[i]=1, the USTAT is

ignored.

Definition 1 *(Unnecessary Retransmission)*

An I-frame retransmission is said to be *unnecessary* if it may be received by the sender after another copy of the same I-frame has already been received. □

Theorem 1 *The SSCOP avoids unnecessary retransmissions.*

Proof
The SSCOP sender retransmits an I-frame, I-frame(i) say, only if (1) a USTAT(i) is received while *retransmitted*[i]=0, or if (2) a STAT(i) is received whose *sequence_number* field is equal to or larger than the value of *last_transmission_time*[i] when the STAT is processed. Consider first a retransmission triggered by the receipt of a USTAT. The USTAT was sent due to the loss of the first copy of I-frame(i). The sender retransmits I-frame(i) only if *retransmitted*[i]=0, namely I-frame(i) has not been retransmitted yet. Thus, and since the channel retains the FIFO order of I-frames[3], the retransmitted copy cannot be received by the receiver after another copy of I-frame(i). Next, consider a retransmission of I-frame(i) triggered by the receipt of a STAT. I-frame(i) is retransmitted only if *last_transmission_time*[i] $\leq$ STAT.*sequence_number*. This indicates that the last copy of I-frame(i) was sent by the sender *before* the POLL to which the STAT responds[4]. Due to FIFO, if the receiver did not receive the last copy of I-frame(i) before having received the POLL, it would never receive this copy or an earlier copy. □

The simulation results in Figure 3 show the importance of avoiding unnecessary retransmissions. The graph shows the throughput vs. window size of the "original SSCOP" as described so far, compared to a "reduced SSCOP" where the mechanism for avoiding unnecessary retransmissions is not implemented (i.e. sequence numbers are not assigned to the POLL frames and the *last_transmission_time* vector is not used). The Round Trip Time (RTT) is taken to be 100 time units, and the packet loss rate (α) is 0.01. The window size is given in RTT units, ranging between 1 (100 I-frames) and 3 (300 I-fames). We assume that the sender window and the receiver window are equal and therefore do not implement any flow control mechanism. The graph also shows the performance of go-back-N and the ideal throughput as calculated according to Eq. (1) and (2) respectively.

First, note that the original SSCOP yields the ideal throughput only for window size larger than 2.5·RTT. However, even for smaller values of window size (2-2.5·RTT) it performs much better than go-back-N that requires only one buffer at the receiver. The contribution of the mechanism for avoiding unnecessary retransmissions is conspicuous for window sizes larger than 2·RTT. Note that without this mechanism the maximum throughput is 0.9 compared to 0.99 (which is the ideal throughput). The reason is that every lost I-frame is reported as lost in RTT/τ = 10 consecutive STAT messages, and therefore is retransmitted by the "reduced SSCOP" 10 times instead of only once. This has the same effect as of increasing the loss rate from 0.01 to 0.1 for which the ideal throughput is only 0.9. By reducing the value of τ,

[3] SSCOP was designed for ATM VC connections, where FIFO must be retained.

[4] Since the sequence number space of POLL frames is enormous (2^{24}), we can ignore possible effects of sequence numbers wrap around.

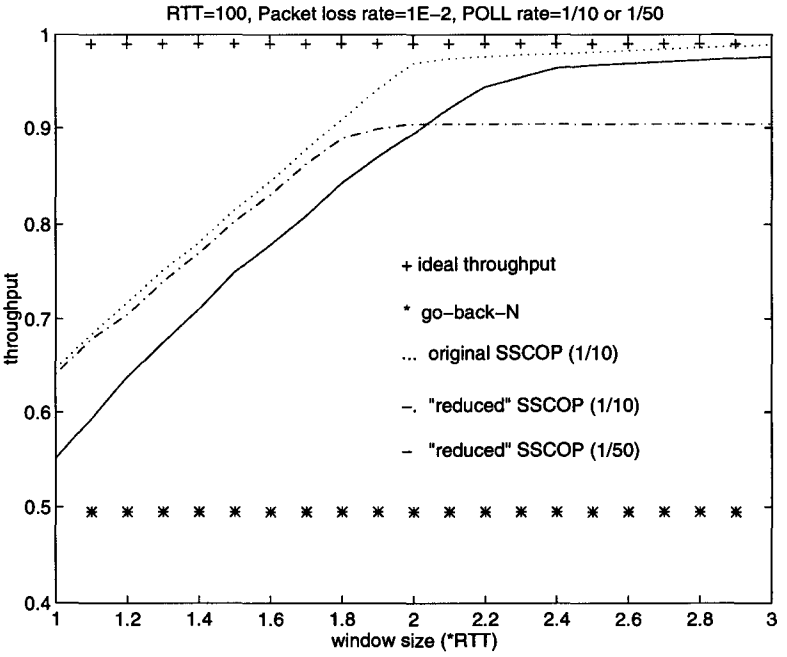

Figure 3: The Importance of Avoiding Unnecessary Retransmissions

namely the rate of the STAT frames, the number of unnecessary retransmitted copies can be reduced. However, this does not increase the performance of the protocol because the sender receives STAT frames in a lower rate, and stays idle more time. This is shown in Figure 3 by the curve marked "reduced SSCOP(50)", where a POLL is sent every 50 I-frames rather than every 10 I-frames and the mechanism for avoiding unnecessary retransmissions is not implemented. In this case the ideal throughput of 0.99 is indeed achieved for large window sizes. However, when the window size is less than 2·RTT, the "Reduced SSCOP(1/10)" that sends a POLL every 10 I-frames but does not avoid unnecessary retransmissions yields better throughput.

More simulation results for the performance of checkpoint mode selective-repeat are given and analyzed in [3].

3 The New Scheme

In the previous section it was shown that SSCOP avoids unnecessary retransmission of I-frames, and that it may therefore achieve the ideal throughput if the sender and the receiver are equipped with enough buffer space. In the following section an alternative scheme for

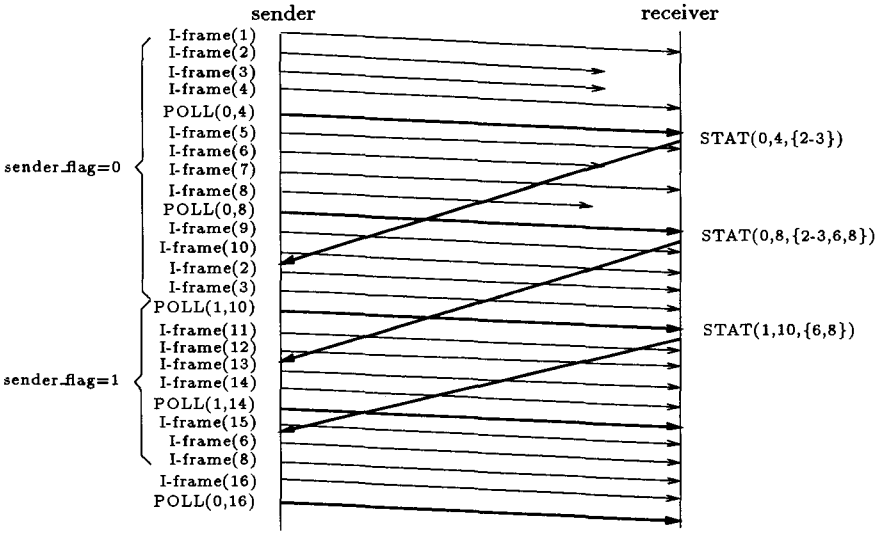

Figure 4: The New Scheme (without USTAT frames)

avoiding unnecessary retransmissions is presented and the advantages of this scheme over the SSCOP are discussed. According to the new scheme, the sender maintains a single binary flag, referred to as *sender_flag,* instead of *POLL_counter.* When a new POLL has to be sent, the sender attaches the value of the *sender_flag* to the *flag* field of the POLL. As in the SSCOP algorithm, the receiver simply copies the *flag* from the POLL to the STAT it sends as a response.

Recall that every STAT frame contains a positive acknowledgment for all the I-frames whose sequence number is $\leq$ POLL.*sequence_number* and a negative acknowledgment for all the I-frames reported as missing. As in the SSCOP protocol, the positive acknowledgment part of the STAT can be processed by the sender unconditionally. However, the negative acknowledgment is processed only if the *flag* field in the STAT frame is equal to the *sender_flag.* Otherwise, the *entire* STAT.*not_received_I-frames* list ignored by the sender. This is a major change to the SSCOP protocol, where the decision whether to retransmit an I-frame or to ignore the request is performed individually per every I-frame in the STAT.*not_received_I-frames* list. If the sender accepts a list of retransmission requests, it retransmits all the I-frames in the list and then inverts the *sender_flag.*

The mechanism as described so far, without USTAT frames, is depicted in Figure 4. The *sender_flag* is initialized to 0. Thus, when the first POLL is sent, POLL.*flag* is set to 0. When the STAT responding to the first POLL is received, *sender_flag* is still 0. Since *sender_flag* is equal to STAT.*flag*, I-frames 2 and 3 are scheduled for re-transmission. When the retransmission is completed, the *sender_flag* is set to 1. When the second STAT is received, there is no match between STAT.*flag* (0) and *sender_flag* (1). This indicates to the sender that the POLL for which the received STAT responds was sent before the last retransmission of *some* packet, and therefore *some* of the retransmissions *may* be unnecessary. Thus, the sender ignores the request for retransmission of I-frames 2, 3, 6 and 8. However, the flag field of the third STAT is 1, and therefore when it is received I-frame(6) is scheduled for retransmission.

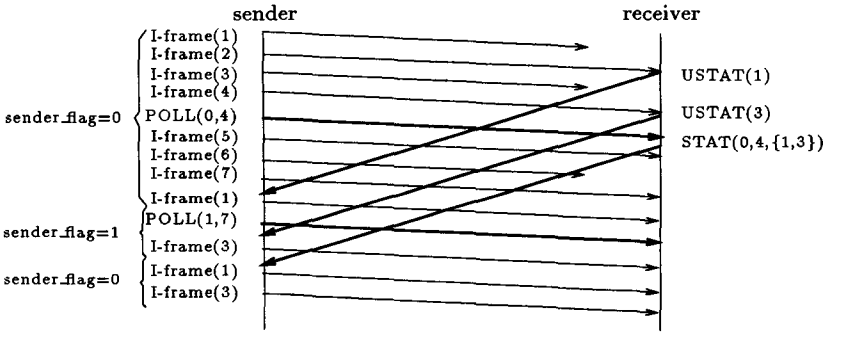

Figure 5: Unnecessary Retransmissions due to USTAT Frames

Again, when the retransmission is completed, *sender_flag* is inverted.

The idea to use a single binary flag in order to ignore unnecessary retransmission requests has already been suggested in a paper [6], co-authored by the author of this paper. However, a scheme based on the binary flag only, as the one presented in [6], has a significant drawback compared to the SSCOP[5] as follows. Since the decision whether to retransmit the list of I-frames reported as lost by the receiver is made for the entire list, rather than for each I-frame individually as in the SSCOP, the sender may ignore relevant retransmission requests. As an example, the third STAT frame in Figure 4 contains a retransmission request for I-frames 2, 3, 6 and 8. I-frames 2 and 3 should not be retransmitted because they were retransmitted after the sending time of the POLL to which the STAT responds. I-frame(6) and I-frame(8), in contrast, are reported as missing in the first time and, therefore, should be scheduled for retransmission. However, since the sender ignores the entire STAT.*not_received_I-frames* list, the retransmission of these two frames takes place only when the next STAT is received. As shown in Section 4 and summarized in the graphs in Figure 11, this drawback may cause throughput degradation of 5-20%.

However, a scheme that combines the *sender_flag* concept with Unsolicited STAT frames as in the SSCOP, would eliminate the effect of this problem on the throughput and achieve the SSCOP performance. Since a USTAT is sent for every new gap detected by the receiver, and since USTAT frames are not ignored by the sender, the first retransmission of every lost I-frame takes place regardless of the POLL/STAT handshake, and is not affected by the mechanism that tells the sender when to ignore retransmission requests in a received STAT frame. However, the USTAT frames affect the mechanism for avoiding unnecessary retransmissions. In a first glance it seems that in order to avoid unnecessary retransmissions when USTAT frames are used, it is sufficient for the sender to invert the *sender_flag* whenever a USTAT is received. However, as Figure 5 depicts, this does not work. When the first USTAT is received, the *sender_flag* is set to 1. When the second USTAT, for I-frame(3), is received, the *sender_flag* is set back to 0. Thus, the retransmission request in the first STAT is accepted, and I-frames 1 and 3 are unnecessarily retransmitted.

[5]Reference [6] does not discuss this drawback and does not present the SSCOP-like protocol, presented in this paper, which solves this drawback. This is because the binary flag in [6] is presented in the context of reliable transmission of data in networks that guarantee only *"semi-FIFO" routing*, and neither the original SSCOP nor the SSCOP-like protocol presented in this paper are applicable when only semi-FIFO, as opposed to full-FIFO, is guaranteed.

(s_1) Upon receiving STAT$^-$ with STAT.*flag* = *sender_flag*
    (s_{11}) retransmit the set (STAT.*not_received_I-frames* − *sender_set*)
    (s_{12}) *sender_set* ← ϕ
    (s_{13}) *sender_flag* ← 1 − *sender_flag*
(s_2) Upon receiving a USTAT
    (s_{21}) *sender_set* ← *sender_set* ⋃ USTAT.*not_received_I-frames*
    (s_{22}) retransmit the set USTAT.*not_received_I-frames*

Figure 6: The Sender Retransmission Algorithm According to the New Scheme

The conclusion from the example in Figure 5 is that if the new mechanism should work in conjunction with USTAT frames, the sender should be able to disregard retransmission requests for I-frames that have "recently" retransmitted following the receipt of USTAT frames. To this end, the following algorithm is proposed. In addition to the *sender_flag*, the sender maintains another data structure called *sender_set.* Whenever the *sender_flag* is converted, the *sender_set* is emptied. When a USTAT is received, all the I-frames reported as missing are retransmitted, and their sequence numbers are appended to the *sender_set,* but the *sender_flag* is not inverted. When a STAT is received and the retransmission request is accepted because STAT.*flag*=*sender_flag,* only those I-frames in the STAT.*not_received_I-frames* list which do not belong to *sender_set* are retransmitted. When the sender completes retransmitting these I-frames, it converts the *sender_flag* and empties the *sender_set.* In terms of the example in Figure 5, after the first USTAT is received *sender_set* = $\{1\}$, and after the second USTAT is received *sender_set* = $\{1,3\}$. In both cases the *sender_flag* does not change. When the first STAT is received, the retransmission request is accepted because STAT.*flag* = *sender_flag.* However, since STAT.*not_received_I-frames* = $\{1,3\}$ and *sender_set* = $\{1,3\}$, the set (STAT.*not_received_I-frames* − *sender_set*) is empty, and no I-frame is actually retransmitted. Then, the sender sets the *sender_flag* to 1 and the *sender_set* to ϕ.

A formal description of the sender retransmission algorithm according to the new scheme is given in Figure 6. In this algorithm it is assumed that the receiver sends exactly one USTAT whenever a new gap in the sequence numbers of the received I-frame is detected. If this assumption does not hold, the retransmission set in line (s_{22}) of the algorithm should contain only the I-frames in (USTAT.*not_received_I-frames* − *sender_set*) rather than those in USTAT.*not_received_I-frames*, in order to avoid unnecessary retransmissions.

In this description, as well as in the remaining discussion, we denote by STAT$^-$ a STAT frame with retransmission request; namely, a STAT whose *not_received_I-frames* field is not empty. In addition, if a received STAT$^-$ has a *flag* field equal to *sender_flag,* in which case lines (s_{11}) − (s_{13}) of the sender algorithm are performed, the STAT$^-$ frame is said to be *accepted.* Otherwise, the STAT$^-$ is said to be *rejected.*

As proven in Theorem 2 below, the new scheme avoids unnecessary retransmissions despite the use of USTAT frames. It may happen that a STAT that contains a relevant retransmission request is rejected, in which case the retransmission is deferred. However, it is shown in Section 4 that the probability for this is negligible, and therefore the performance of the new scheme is identical to the performance of the SSCOP.

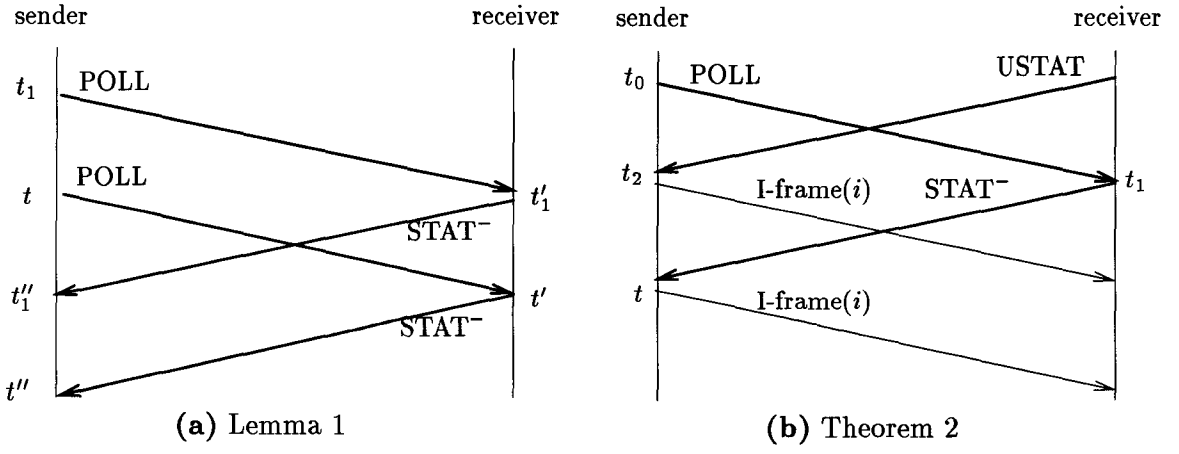

Figure 7: The Proof of Lemma 1 and Theorem 2

The following lemma is used in Theorem 2 and as in the performance analysis of the new scheme.

Lemma 1 *Suppose that the sender accepts at time t'' a STAT$^-$ frame. Let t' be the time this STAT is sent by the receiver and t be the time the POLL to which this STAT responds is sent by the sender (Figure 7(a)). Then, no STAT$^-$ is accepted by the receiver during (t, t''). (In other words, the sender may accept no more than one STAT$^-$ frame during RTT).*

Proof
Consider the first POLL/STAT handshake for which the claim is incorrect. Without loss of generality suppose that at t the value of *sender_flag* is 0. This implies that at t'' the value is 0 as well. Let t''_1 be the *last* time when the sender accepts a STAT$^-$ during (t, t''). This implies that at t''_1 the sender receives a STAT$^-$ with STAT.*flag* $= 1$ and changes the *sender_flag* to 0. Let t'_1 be the time when this STAT$^-$ is sent, and t_1 be the time when the POLL to which this STAT responds is sent (Figure 7(a)). Since at t_1 *sender_flag* is 1 whereas at t it is 0, the value of *sender_flag* changes during (t_1, t), due to the receipt of a STAT$^-$ frame. This implies that the POLL/STAT handshake initiated at t is not the first handshake for which the claim does not hold, in contradiction to our assumption. Thus, the lemma is correct. □

Theorem 2 *The new scheme avoids unnecessary retransmissions.*

Proof
First recall our assumption that the sender sends exactly one USTAT frame whenever it detects a new gap in the sequence numbers of the received I-frames. To prove the theorem, suppose it is incorrect and that I-frame(i) is unnecessarily retransmitted. Let t be the first time when the sender retransmits I-frame(i) unnecessarily. Due to FIFO, an unnecessary retransmission cannot take place following the receipt of a USTAT by the sender. This is

because if the sender is informed by a USTAT that I-frame(i) is missing, then the first copy of I-frame(i) was indeed lost and a second copy has not been sent yet.

This implies that the unnecessary retransmission of I-frame(i) at t occurs due to the receipt of a STAT$^-$ frame at t^- whose *not_received_I-frames* field contains the number i and whose *flag* field is equal to *sender_flag*. Without loss of generality, assume that both STAT.*flag* and *sender_flag* are 0. Let t_1 be the time this STAT$^-$ is sent by the receiver and t_0 be the time the sender sends the POLL to which this STAT$^-$ responds (Figure 7(b)).

Since the transmission of I-frame(i) at t^- is unnecessary, a previous copy of I-frame(i) has been received by the receiver. In fact, following our assumption that the retransmission of I-frame(i) at t is the first unnecessary retransmission, exactly one copy of I-frame(i) is received by the receiver before the copy sent at time t. Obviously, this couldn't be the first copy of I-frame(i) sent by the sender because if the first copy of any I-frame is received by the receiver, it cannot be reported as missing by any STAT frame. This implies that the first copy of I-frame(i) accepted by the receiver is sent by the sender, at time t_2 say, following either (1) the receipt and acceptance of a STAT$^-$ or (2) the receipt of a USTAT. However, since $t_2 \in (t_0, t)$ (because if $t_2 < t_0$ then the STAT sent by the receiver at t_1 cannot report that I-frame(i) is missing) then, by Lemma 1, case (1) is impossible.

Figure 7(b) depicts the second case, where the first copy of I-frame(i) is sent, at $t_2 \in (t_0, t)$, following the receipt of a USTAT. By the sender algorithm, at time t_2 the value i is appended to the *sender_set*. Since by Lemma 1 the sender does not accept a STAT during (t_0, t), and therefore does not empty the *sender_set* during this time interval, it cannot retransmit I-frame(i) following the receipt of the STAT at t, in contradiction to our assumption. □

Note that the theorem must hold even if the sender algorithm is changed such that the *sender_flag* is inverted *only* if the received STAT$^-$ indeed leads to any retransmission. Namely, if the set (STAT.*not_received_I-frames* − *sender_set*) is not empty. (Since the Theorem holds regardless of possible loss of STAT frames, it must hold even if an "intelligent channel" gets rid of any STAT that will not lead to any retransmission.) However, such a change will lead into a livelock as Figure 8 shows. In this figure, I-frame(i) is lost and USTAT(i) is sent when the receiver receives I-frame($i+1$). Upon receiving the USTAT, the sender appends i to the *sender_set* which was empty, changes the *sender_flag*, to 1 say, and retransmits the missing I-frame. The retransmitted copy of I-frame(i) is lost as well, and the sender will reject all the STAT$^-$ frames it subsequently receives as long as they require retransmission of I-frame(i) only. This livelock will be solved after another I-frame is lost, but by that time the performance is hurt.

The new scheme has two main advantages over the original SSCOP protocol or any other protocol based on the checkpoint mode presented in the past [3, 5, 8]. First, as already explained in Section 1, it reduces the amount of high-speed memory needed for storing control information of the protocol. The SSCOP requires that a 24-bit sequence number will be kept for every outstanding I-frame, whereas the new scheme requires a single *sender_flag* bit for all the frames. In addition to this single bit, the new scheme needs some buffer space in order to represent the *sender_set*. This can be done by using the SSCOP *retransmitted*[] vector, which is needed in order to avoid unnecessary retransmissions when multiple USTAT frames are sent whenever a new gap is detected. Alternatively, in order to dispense with the *retransmitted*[] vector, given that only one USTAT is sent for every new gap, the sender can dedicate two or three 24-bit words to represent the *sender_set*. This is based on the simple observation that no more than RTT time units after a sequence number is appended

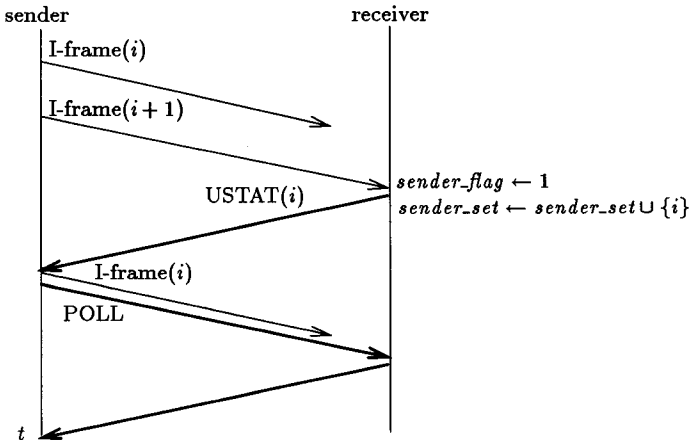

Figure 8: A Livelock Due to a Change in the Condition for Accepting a STAT$^-$

to the *sender_set,* the *sender_set* will be emptied. The expected number of I-frames for which the sender may receive a USTAT frame during this RTT time units is $\alpha \cdot \text{RTT}$, which is practically bounded by 1. The simulation results for the new scheme, presented and discussed in Section 4, that show that the new scheme has the same performance as the SSCOP, have been obtained by implementing the *sender_set* in this way.

The other important advantage of the new scheme is that it is not exposed to problems resulting from the sequence numbers associated with the SSCOP STAT frames. As an example, suppose that the entry in the *last_transmission_time*[] vector associated with a lost I-frame, I-frame(i) say, gets a wrong value due to memory failure or protocol error. Suppose also that the wrong value is much greater than the sequence number of the next POLL sent by the sender. In such a case all the retransmission requests for I-frame(i) will be rejected by the sender and the protocol will fail. The POLL/STAT handshake in the new protocol, in contrast, is a self-stabilized. If the *sender_flag* gets the wrong value, the sender will ignore relevant retransmission requests during RTT time units only, after which the protocol recovers.

4 Performance of the New Scheme

As already said, simulation results show no difference between the performance of the SSCOP and the performance of the new scheme. To understand the similarity in the performance of the two schemes, note that the only case where the two schemes work differently is when a relevant retransmission request in a STAT frame is accepted by the SSCOP but ignored by the new scheme. For the formal discussion we shall need the following definition.

Definition 2 *(Relevant Retransmission Request)*

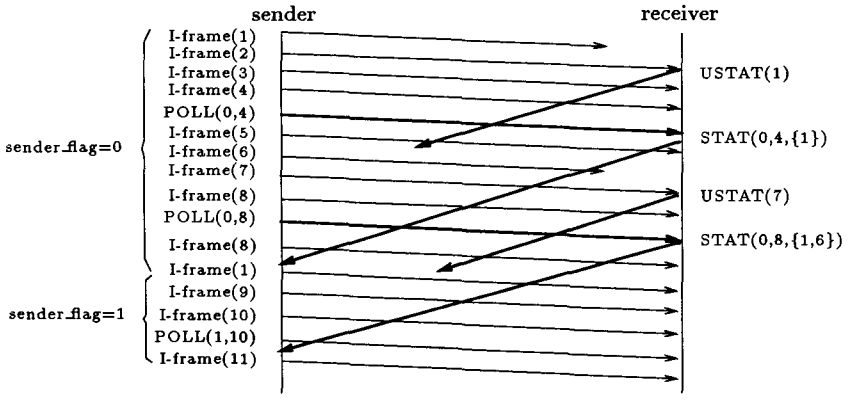

Figure 9: The Sender of The New Scheme Ignores a Relevant Retransmission Request

A retransmission request for I-frame(i) in a STAT frame is said to be relevant if and only if the sender does not send a copy of I-frame(i) during (t, t'), where t' is the time when the STAT is received by the sender and t is the time when the POLL to which the STAT responds was sent. □

Theorem 3 *In the SSCOP the sender never ignores relevant retransmission requests, whereas in the new scheme the sender may ignore relevant retransmission requests.*

Proof
The SSCOP sender ignores a retransmission request for I-frame(i) only if at the time when the STAT is received STAT.*sequence_number* $\leq$ *last_transmission_time*[i]. Since *last_transmission_time*[i] contains the *sequence_number* of the STAT sent before the last transmission of I-frame(i), the fact that STAT.*sequence_number* $\leq$ *last_transmission_time*[i] implies that I-frame(i) was transmitted after the POLL to which the considered STAT responds. Thus, the claim holds for the SSCOP. To show that a sender working according to the new scheme may ignore relevant retransmission requests, consider the scenario depicted in Figure 9. In this scenario, the first STAT is accepted by the sender and leads to retransmission of I-frame(1). Therefore, the value of the *sender_flag* is inverted to 1. The second STAT contains an irrelevant retransmission request for I-frame(1) but a relevant retransmission request for I-frame(6). The SSCOP sender would have retransmitted I-frame(6) in this case and ignored the request for I-frame(1). The sender of the new scheme, however, is not able to distinguish between the two retransmission requests. Since the STAT.*flag* is not equal to the *sender_flag,* the STAT is rejected. □

In the following analysis we shall show that the probability P_I that the sender of the new scheme ignores a relevant retransmission request for a given I-frame, I-frame(i) say, is very small, which explains the similarity in the performance of the two schemes. Recall that α is the probability that an I-frame is lost, and let β be the probability that a control frame (POLL, STAT or USTAT) is lost. We distinguish between the two parameters because usually $\alpha > \beta$.

Suppose that the sender of the new scheme ignores a STAT$^-$ frame with a relevant retransmission request for I-frame(i). Let t_6 be the time when the STAT$^-$ is received by the sender, t_3 be the time when it is sent and t_2 be the time when the POLL to which this STAT responds is sent (Figure 10(a)). By Lemma 1, a necessary condition for this STAT$^-$ to be ignored by the sender is that the sender receives during (t_2, t_6), at t_4 say, a STAT$^-$ and accepts it. Let t_1 be the transmission time of this STAT$^-$ and t_0 be the transmission time of the POLL to which this STAT responds In the rest of the analysis we shall distinguish between the following two cases: (a) the STAT$^-$ accepted by the sender at t_4 does not contain a retransmission request for I-frame(i); and (b) the STAT$^-$ accepted at t_4 contains a retransmission request for I-frame(i). We shall denote by $\mathcal{E}_1$ the event that a retransmission request for I-frame(i) is ignored due to a STAT$^-$ (the one sent at t_1) that does not contain a retransmission request for I-frame(i), and by $\mathcal{E}_2$ the event that a retransmission request for I-frame(i) is ignored due to a STAT$^-$ that contains a retransmission request for I-frame(i). Thus,

$$P_I = Prob(\mathcal{E}_1) + Prob(\mathcal{E}_2) \tag{3}$$

We shall start by finding $Prob(\mathcal{E}_1)$. This event requires that the following will hold:

- I-frame(i) is transmitted in the first time during (t_0, t_2). This is because if the first transmission time of this I-frame is before t_0 then the STAT sent at t_1 will contain a retransmission request for I-frame(i), whereas if the first transmission time is after t_2, the STAT sent at t_3 will not contain a retransmission request for I-frame(i).
- The first transmission of I-frame(i) is lost.
- The USTAT sent by the sender for I-frame(i) is lost as well. If the USTAT is not lost, then it will be received by the sender during (t_4, t_6), and will trigger a retransmission of I-frame(i) during this time interval, in which case the STAT received by the sender at t_6 cannot contain a *relevant* retransmission request for I-frame(i).

The conclusion is that

$$Prob(\mathcal{E}_1) = \alpha\beta P_1 \tag{4}$$

where P_1 is the probability that the first STAT sent after the lost USTAT is ignored (since its *flag* field is not equal to the *sender_flag*). From Figure 10(b) follows that P_1 is the probability that the sender receives and *accepts* a STAT$^-$ frame during (t_2, t_6), which is sent regardless of the loss of I-frame(i). Thus,

$$P_1 \leq Prob(\text{a STAT}^- \text{ is received and accepted by the sender during RTT}) \tag{5}$$

Recall that the sender sends a POLL frame every τ time units. Let $\gamma = \text{RTT}/\tau$ be the number of POLL frames the sender sends during RTT. Thus, it may receive no more than γ STAT frames back (some of the POLL or the STAT frames may get lost), and P_1 is the probability that one of the first $(\gamma - 1)$ frames is STAT$^-$ and is accepted by the sender (Figure 10(c)). Let $\mathcal{A}^i$ be the event that the $i'th$ STAT is a STAT$^-$ and is accepted by the sender. Thus,

$$Prob(\text{a STAT}^- \text{ is received and accepted by the sender during RTT}) = Prob(\bigcup_{1 \leq i \leq \gamma - 1} \mathcal{A}^i)$$

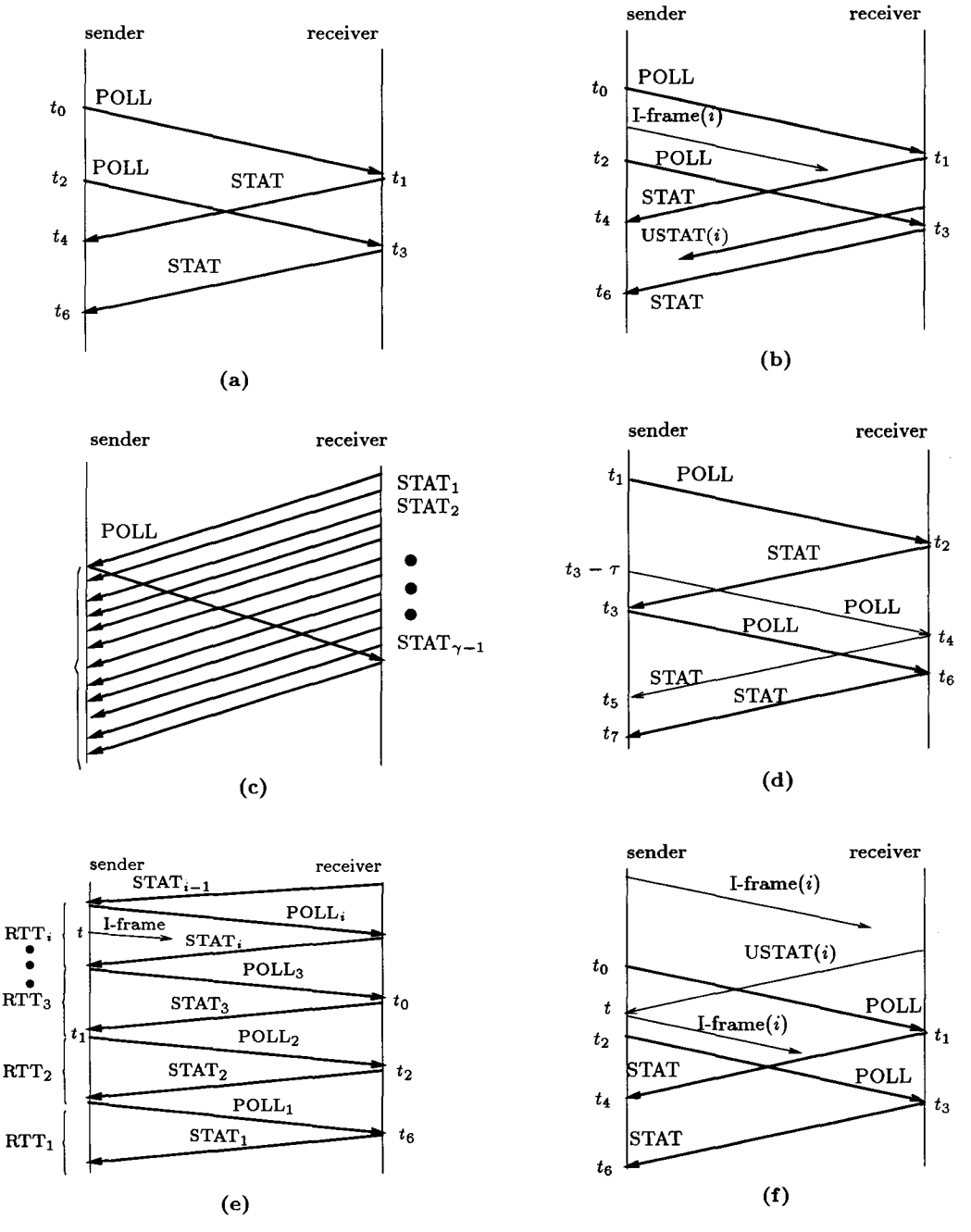

Figure 10: Analysis of the New Scheme

However, by Lemma 1 only one of these $(\gamma-1)$ STAT frames can be a STAT$^-$ accepted by the sender. Thus,

$$\forall\, i,j,\ i \neq j \quad Prob(\mathcal{A}^i \cap \mathcal{A}^j) = 0 \tag{6}$$

and therefore

$$Prob(\text{a STAT}^- \text{ is received and accepted by the sender during RTT}) = \sum_{1 \leq i \leq \gamma-1} Prob(\mathcal{A}^i) = (\gamma-1)\cdot P_2$$

where P_2 is the probability that a given STAT frame is a STAT$^-$ and is *accepted* by the sender; namely, that the STAT causes the sender to change *sender_flag*. Consequently

$$P_1 \leq (\gamma-1)P_2 \tag{7}$$

Recall that a STAT$^-$ frame has a non-empty *not_received_I-frames* list. We shall denote by STAT^{-t} a STAT$^-$ for which the I-frame with the lowest identity in this list was originally sent at t. This notation will help us in defining mutually exclusive events. For instance, the first, second and third STAT frames in the example depicted in Figure 4 are denoted STAT$^{-t'}$, STAT$^{-t'}$ and STAT$^{-t''}$, respectively, where t' is the time when the first copy of I-frame(2) is sent and t'' is the time when the first copy of I-frame(6) is sent.

In order to find P_2, and without loss of generality, we shall concentrate on the STAT sent at t_6 in Figure 10(d). Note that all the frames shown in this figure are either POLL or STAT frames, but not all the STAT/POLL frames are shown. Let $\mathcal{B}_1$ be the event that the STAT sent at t_6 is a STAT^{-t} where $t > t_3$ and it is accepted by the sender. Let $\mathcal{B}_2$ be the event that the STAT sent at t_6 is a STAT^{-t} where $t \in (t_3-\tau, t_3)$ and it is accepted by the sender. (Recall that a POLL is sent every τ time units; thus, $(t_3-\tau)$ is the last time before t_3 when a POLL is sent by the sender.) Let $\mathcal{B}_3$ be the event that the STAT sent at t_6 is a STAT^{-t} where $t \in (t_1, t_3-\tau)$ and it is accepted by the sender. Finally, let $\mathcal{B}_4$ be the event that the STAT sent at t_6 is a STAT^{-t} where $t < t_1$ and it is accepted by the sender. Since $\mathcal{B}_1 \cdots \mathcal{B}_4$ are pairwise mutually exclusive

$$P_2 = \sum_{1 \leq i \leq 4} Prob(\mathcal{B}_i) \tag{8}$$

We shall continue by calculating the probability for each of these four events. Since the considered STAT, which responds to a POLL sent at t_3, cannot report the loss of an I-frame whose first transmission time is after t_3, $Prob(\mathcal{B}_1) = 0$. Three necessary conditions for $\mathcal{B}_2$ are that at least one of the τ or less[6] I-frames transmitted by the sender in the first time during $(t_3-\tau, t_3)$ is lost, that no STAT$^-$ is accepted by the sender during (t_3, t_7) and that both the POLL sent at t_3 and the responding STAT are not lost. (Note that these conditions are insufficient, since it is also needed that all the I-frames sent by the sender before $t_3-\tau$ are received by the receiver before t_6 after either one or several transmissions.) The probability for the first condition is $1-(1-\alpha)^\tau$. By Eq. 5 and 7, the probability for the second condition is $1-(\gamma-1)P_2$. Finally, the probability for the third condition is $(1-\beta)^2$. Since these three conditions are pairwise independent, we obtain

$$Prob(\mathcal{B}_2) \leq (1-(1-\alpha)^\tau)(1-(\gamma-1)P_2)(1-\beta)^2 \tag{9}$$

[6] If the sender happens to retransmit some I-frames during $[t_3-\tau, t_3]$ then less that τ new I-frames can be transmitted during this time interval.

(the first and the second conditions are independent because the second event depends on the fate of the I-frames transmitted by the sender *before* $(t_3 - \tau)$ whereas the first event depends on the fate of the I-frames transmitted during $(t_3 - \tau, t_3)$).

The necessary conditions for $\mathcal{B}_3$ are that at least one of the (RTT $- \tau$) or less I-frames transmitted by the sender in the first time during $(t_1, t_3 - \tau)$ is lost and that all the STAT$^-$ reporting this loss and sent by the receiver before t_6 are not accepted whereas the one sent at t_6 is accepted. The probability for the first condition is $\leq 1 - (1-\alpha)^{\mathrm{RTT}-\tau}$. The second condition requires that at least one STAT$^-$ is accepted by the sender during (t_1, t_3) in order to prevent the sender from receiving the STAT$^-$ frames sent following the loss of the I-frame(s) transmitted in the first time during $(t_1, t_3 - \tau)$. On the other hand, by Lemma 1 if the sender accepts a STAT$^-$ sent after t_2 it cannot accept the STAT$^-$ sent at t_6. This leads to the conclusion that $\mathcal{B}_3$ is possible only if the particular STAT sent at t_2 is a STAT$^-$ and it is accepted by the sender. The probability for this condition, which is independent of the other condition of $\mathcal{B}_3$ is P_2. This should be multiplied by $(1-\beta)^2$, the probability that neither the POLL sent at t_3 nor the responding STAT is lost. Thus,

$$Prob(\mathcal{B}_3) \leq (1 - (1-\alpha)^{\mathrm{RTT}-\tau}) P_2 (1-\beta)^2 \tag{10}$$

To find $Prob(\mathcal{B}_4)$, namely the probability that the STAT sent at t_6 is a STAT^{-t} where $t < t_1$ and it is accepted by the sender, consider a division of the sender time axis into time intervals of RTT units, as Figure 10(e) shows. Let the last RTT be RTT_1, the previous one be RTT_2 and so forth. According to the definition of $\mathcal{B}_4$, $t \in \mathrm{RTT}_i$ where $i \geq 3$. Since all the STAT frames responding to POLL frames sent by the sender after t must be STAT$^-$, then in order for the sender to accept the STAT$^-$ sent at t_6 (STAT_1 in Figure 10(e)), it must also accept STAT_2, STAT_3, $\cdots$ STAT_i. The necessary conditions for this are (1) that all these STAT frames and the POLL frames to which they respond to are not lost, and (2) that STAT^{i-1} is a STAT$^-$ and is accepted by the sender. The probability for (1) is $(1-\beta)^{2i}$, and for (2) is P_2 independently of what happens at the channel after t. the probability Another condition is that all the $i-1$ copies of the considered I-frame, transmitted by the sender after the receipt of STAT^i, STAT^{i-1} $\cdots$ STAT^2 are also lost, which happens with probability α^{i-1}. To conclude

$$Prob(\mathcal{B}_4) \leq P_2 (1-\beta)^{2i} \alpha^{i-1}$$

and since $i \geq 3$ we obtain

$$Prob(\mathcal{B}_4) \leq P_2 (1-\beta)^6 \alpha^2 \tag{11}$$

Substituting Eq. 9, 10 and 11 into 8, while using the inequality $1-(1-\alpha)^\tau \leq \alpha\tau$, we obtain

$$P_2 \leq \alpha\tau(1-\beta)^2 + (1-\beta)^6 \alpha^2 P_2$$

Solving for P_2 yields

$$P_2 \leq \frac{\alpha\tau(1-\beta)^2}{1-(1-\beta)^6\alpha^2} \leq \alpha\tau \tag{12}$$

Substituting Eq. 12 and 7 into 4 we obtain

$$Prob(\mathcal{E}_1) = \alpha\beta P_1 \leq \alpha\beta(\gamma-1)P_2 \leq \alpha\beta\gamma P_2 \leq \alpha\beta\gamma\alpha\tau \leq \alpha^2\beta\mathrm{RTT} \tag{13}$$

Next, consider $\mathcal{E}_2$, where the STAT$^-$ accepted by the receiver at t_4 contains a retransmission request for I-frame(i). This indicates that I-frame(i) was sent in the first time before t_0.

In addition, since the STAT$^-$ received by the sender at t_6 is assumed to contain a relevant retransmission request for I-frame(i), then by Definition 2 the STAT$^-$ accepted by the sender at t_4 does not lead to a retransmission of I-frame(i). This implies that at t_4 $i \in sender\_set$, which indicates that the sender receives a USTAT(i) and retransmits I-frame(i) at $t < t_4$. In fact t is even earlier than t_2, since otherwise the retransmission request for I-frame(i) in the STAT$^-$ sent by the receiver at t_3 could not be considered relevant. Another obvious requirement for this sequence of events is that the copy of I-frame(i) sent by the sender at t is lost. This sequence of events, which requires thus far the loss of the first two copies of I-frame(i), is depicted in Figure 10(f). Consider now the size of time interval (t, t_4). If it is smaller than τ, in which case the STAT accepted at t_4 is the first STAT received after t, the number of POLL sent during (t, t_4) is either 0 or 1. This implies that the third transmission of I-frame(i) takes place at t_6 or at $t_6 + \tau$. This case can be ignored, since in the worst case (the third transmission occurs at $t_6 + \tau$), it has the same effect as of a lost STAT. The case we are interested in is that $t_4 - t > \tau$. However, in such a case the sender receives one or more STAT$^-$ frames during (t, t_4). Let the receiving time of the first of them be t'. In order for the STAT$^-$ received at t_4 to be accepted, the one received at t' should be ignored. This happens only if another STAT$^-$ is accepted during (t'', t'), where t'' is the sending time of the POLL frames to which the STAT$^-$ received at t' responds. The probability for such an event, which is independent of the loss of I-frame(i) (since the STAT frames which can satisfy this event respond to POLL sent before the first transmission of I-frame(i)), is P_1 as calculated before. To conclude,

$$Prob(\mathcal{E}_2) \leq \alpha^2 P_1 \leq \alpha^3 \text{RTT} \tag{14}$$

Substituting Eq. 13 and 14 into 3 we obtain

$$Prob(P_I) \leq \alpha^2(\alpha + \beta)\text{RTT}$$

For example, if RTT=100, $\alpha = 10^{-3}$ and $\beta = 10^{-5}$, the probability that the retransmission of a given I-frame will be delayed is $\leq 10^{-7}$, and in a session of 10^6 I-frames the probability that the retransmission of *any* I-frame will be delayed is less than 0.1. In any case only one retransmission of an I-frame can be delayed, and the delay is always $\leq RTT$.

Only when the USTAT frames are eliminated from the new scheme, the throughput of the new scheme is less than the SSCOP throughput. This is because in such a case Eq. 4 changes to $Prob(\mathcal{E}_1) = \alpha P_1$, and P_I is increased by a factor of β^{-1}. Figure 11(a) shows simulation results for RTT=100, $\alpha = 10^{-2}$ and $\beta = 10^{-4}$, assuming that a POLL is sent after every 10 I-frames (i.e. $\tau = 10$). Figure 11(b) shows simulation results for the same parameters except that RTT=1000. In both cases the upper curve represents the throughput of the new scheme with USTAT frames and of the SSCOP, whereas the lower one represents the performance of the new scheme without USTAT frames. When α is reduced to 10^{-4} then for RTT=100 the differences between the two curves disappear. When α is reduced to 10^{-5}, the differences disappear also for RTT=1000.

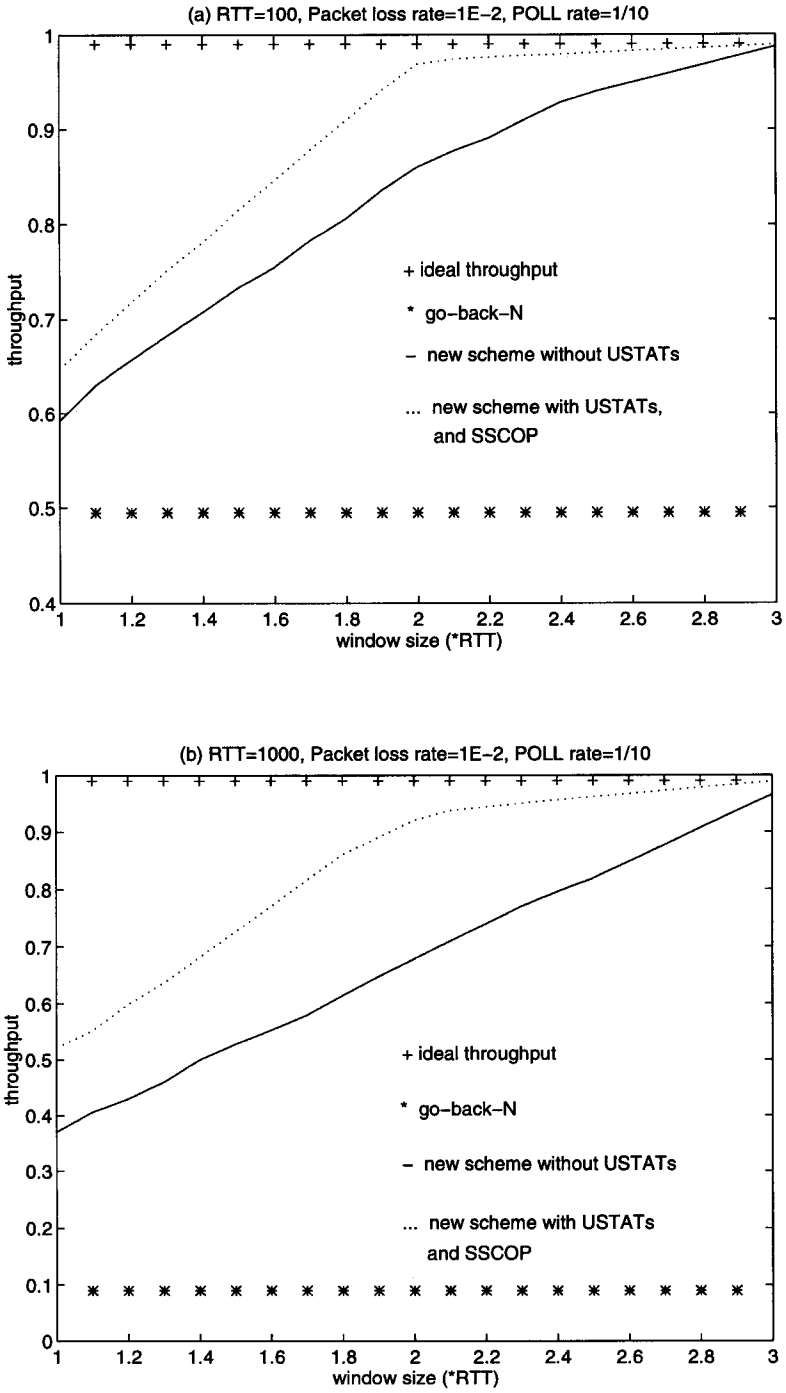

Figure 11: The Throughput of the New Scheme With and Without USTAT Frames (the throughput of the new scheme with USTAT frames is identical to the SSCOP throughput)

5 Conclusions

The paper has presented the SSCOP protocol, as defined by the ATM forum, which is a selective-repeat protocol based on the checkpoint mode. The paper has discussed the means by which the SSCOP avoids unnecessary retransmissions of I-frames by the sender. The paper has shown that without avoiding unnecessary retransmissions, the throughput of the protocol is significantly reduced. Then, the paper has presented an improved scheme for avoiding unnecessary retransmissions in checkpoint mode selective-repeat protocols. The new scheme was shown to have two main advantages over the SSCOP. Firstly, it significantly reduces the amount of high-speed memory needed for storing control information of the protocol. Secondly, it enhances the recovery from certain failure conditions. The only potential drawback of the proposed scheme compared to the SSCOP or to any other protocol presented in the past which is based on the checkpoint mode is that under certain circumstances the sender may ignore relevant retransmission requests. However, the paper has analyzed the sequence of events that must take place before the sender may ignore a relevant retransmission request and shown that the probability for such a sequence is negligible.

6 References

[1] M. Anagnostou and E. Protonotatios. Performance analysis of the selective-repeat ARQ. *IEEE Transactions on Communications*, 34:127-135, Feb. 1986.

[2] D. Banks and M. Prudence. A high-speed network architecture for a PA-RISC workstation. *IEEE Journal on Selected Areas in Communications*, 10(1):191–202, February 1993.

[3] W. Bux, P. Kermani and W. Klienoder. Performance of an improved data link control protocol. In *9'th ICCC*, 251–257, Tel-Aviv, Israel, Oct. 88.

[4] W. Bux, K. Kummerle, and H.L. Truon. Data link control performance: Result comparing HDLC operational modes. *Computer Networks and ISDN Systems*, 6(1):37–51, 1982.

[5] Brodd W. and Donnan R. Data link control improvements for satellite transmission. *Satellite and Comp. Comm.*. Elsevier Science Publishers B.V., North-Holland, Amsterdam 1983.

[6] R. Cohen and Y. Ofek. Reliable transmission of data over a semi-FIFO routing layer. *Computer Networks and ISDN Systems*, 27:1633–1649, 1995. (a shortened version of this paper was presented in Infocom'94).

[7] Davie B. The Architecture and Implementation of a High-Speed Host Interface. *IEEE Journal on Selected Areas in Communications*, 10(1):228–239, February 1993.

[8] Donnan R. Method and system for retransmitting incorrectly received numbered frames in a data transmission system. *United States Patent 4,439,859 Mar. 27 1984.*

[9] Easton M. Batch throughput efficiency of ADCCP/HDLC/SDLC selective reject protocols. *IEEE Transactions on Communications*, 28(2):187-195, Feb. 1980.

[10] Institute of Electrical and Electronics Engineers Inc. ANSI/IEEE standards 802.2-1989 (ISO 8802-2): Logical link control, 1989.

[11] Sastry A. Improving Automatic Repeat Request performance on satellite channels under high error rate conditions. *IEEE Transactions on Communications*, 23(4):436-439, April 1975.

[12] A. Shankar and S. Lam. An HDLC protocol specification and its verification using image protocols. *ACM Transactions on Computer Systems*, 1(4):331–368, November 1983.

[13] *Draft new ITU-T Recommendation Q.SAAL.1: Service Specific Connection Oriented Protocol (SSCOP) Specification.*

[14] P. Yu and S. Lin. An efficient selective repeat ARQ scheme for satellite channels and its throughput analysis. *IEEE Transactions on Communications*, 29:353-363, Mar. 1981.