15

Flow Control Mechanisms for SAAL Links

K. Kant
Bell Communications Research
331 Newman Springs Road, Red Bank, NJ 07701, USA
Tel: (908)758-5384, Email: kant@perf.bellcore.com

Abstract

Signalling protocols in telecommunication networks provide for link-level flow-control procedures to ensure that the signalling traffic brought in by a link does not overwhelm the processing capacity on the receive end. This paper examines the flow-control procedures available in the signalling ATM adaptation layer (SAAL). The SAAL receiver effects flow control by granting an appropriate amount of *credit* to the SAAL transmitter; however, the details of how much credit is granted and when, are not standardized and left to implementation. This paper proposes two simple credit allocation schemes and provides guidelines for their parameterization. Since the receive buffer size depends on the credit mechanism, the paper also proposes buffer sizing guidelines.

The work reported here was motivated by the need for flow control on SAAL links deployed in the current common channel signalling (CCS) networks. It is shown that flow-control becomes even more important in broadband networks, thereby requiring well-designed credit control schemes.

Keywords

Asynchronous transfer mode (ATM), Signalling ATM adaptation layer (SAAL), Service-specific connection oriented protocol (SSCOP), Flow-control, Buffer sizing, Transfer controlled (TFC) procedure, Signalling message handling (SMH) congestion.

1 INTRODUCTION

This paper examines the problem of link-level flow-control in an asynchronous transfer mode (ATM) signalling network. The purpose of signalling is to establish and tear-down end to end connections and to manage the network. In an ATM network, signalling information is carried on specially designated virtual channels in the control plane and is supported by the ATM adaptation layer 5 (AAL5). The ATM signalling suite of protocols

L. Mason et al. (eds.), *Broadband Communications*

is also known as SAAL (Signalling AAL), and includes a number of sublayers: CP (common part), SSCOP (service-specific, connection-oriented protocol), SSCF (service-specific coordination function), and LM (layer management). Of these, the transmission protocol, SSCOP, is the one that is of primary interest in this paper.

SSCOP provides a *credit* mechanism by which the receiver periodically grants credit to the transmitter for sending *user* protocol data units (PDUs)* Section 2 briefly describes SSCOP and the credit granting scheme. The SSCOP standard does not specify how the credit is to be updated, thereby allowing the flow-control scheme to be implementation dependent. In this paper, we shall propose two schemes for updating the credit, compare their performance, and study their parameterization under various scenarios.

In general, if the receiver grants credit for n additional PDUs, it can be viewed as a commitment on its part to accept them even if they all happen to arrive back to back. This commitment can always be fulfilled by using a large receive buffer, but doing so is undesirable from a delay perspective. On the other hand, dropping a solicited PDU (i.e., one for which credit was granted) because of lack of receive buffer is also undesirable because the link error monitor cannot distinguish between losses due to errors and losses due to deliberate rejections of PDUs. In this paper we shall discuss these issues and suggest a few receive buffer sizing schemes.

The work reported here was motivated by the need to enhance the capacity of common channel signalling (CCS) links used in the current narrow-band telecommunication networks. We first discuss this aspect briefly and then turn to broadband networks.

The currently deployed CCS network in the US uses "quasi-associated signalling", where each switch "homes" only on a pair of STPs (signalling transfer points), and the connectivity is provided via STPs only. The SCPs (signalling control points), that provide database capabilities, also "home" on a pair of STPs. Such a network architecture may face two limitations as the network size and traffic grow (a) inadequate number of physical link ports available on the STP, and (b) inadequate total capacity of STP–STP and STP–SCP link-sets. With a rapid deployment of advanced telecommunications services such as PCS and AIN, these limitations are already being faced by several network providers and have prompted the deployment of high speed links (HSLs) in the CCS network. For a variety of reasons, it was decided to use SAAL based HSLs running over physical 1.536 Mb/sec channels. (See GR-2878). However, this deployment results in two complications that are relevant from a flow-control perspective:

1. If the driver for converting a DS0 link-set to a HSL is the shortage of port capacity or future traffic growth, the conversion may cause capacity problems. For example, replacing 4 DS0 links by a single HSL not only frees up 3 ports, but also allows the traffic to come in at the rate of about 16 DS0 links.† Thus, during high usage intervals the receive end may become overwhelmed and cause excessive delays and buffer overflows.
2. Consider a STP serving a mix of HSLs and DS0 links. Since a HSL can bring much

*A more accurate terminology for user PDUs is sequenced data PDUs; we use user PDUs here since signalling does not use unnumbered data capability of SSCOP.

†Because of much higher padding overhead of SAAL compared to MTP2, one 1.5 Mb/sec SAAL link is equivalent to about 16 DS0 MTP2 links for current messages sizes (Kant 1995a).

more traffic than a DS0 link, a focussed overload coming from the HSL may result in *signal message handling* (SMH) congestion, i.e., congestion at level 3 of the STP. If SMH congestion control is *not* implemented, this situation may result in excessive delays, timeouts, and customer retries. Otherwise, the STP will send TFCs (transfer controlled messages) to all its traffic sources, which is also undesirable since a global action is initiated in response to overloading of a single link.

Both problems occur essentially because of coarse granularity in selecting a link speed — the link speed can be either 56 Kb/sec or 1.5 Mb/sec with no intermediate values allowed. In a broadband network, intermediate values *are* allowed; in fact, since the signalling channels in a broadband network are merely virtual circuits, it is easy to change their bandwidth as needed. Yet, it is unlikely that this flexibility will actually be used. The problem is that a number of system parameters (e.g., buffer sizes, congestion thresholds, PDU pickup policy from level 2, etc.) are tied to link speed, and it is expensive to keep track of them or change them via an operations interface. Moreover, given that a 1.5 Mb/sec signalling channel occupies 1 percent or less of the ATM "pipe" capacity, there is little to be gained by using a lower rate. Thus the problems mentioned above do not go away in broadband networks.

The large disparity between the bandwidth of the signalling channel and the ATM pipe brings another complication in broadband networks: Over short periods (defined by the burst size parameter of the policing function), the signalling PDUs can arrive at a much higher rate than the peak rate of the signalling channel.[‡] In the absence of any flow-control, such bursts could easily overwhelm processing resources at SAAL and MTP3 levels, since those processors would certainly not be engineered for any more than the peak rate of the signalling channel. This makes the flow-control even more important in broadband networks.

2 SAAL PROTOCOL AND FLOW-CONTROL MECHANISM

SSCOP belongs to the class of ARQ (automatic repeat request) algorithms with selective retransmission of errored messages along with periodic polling of the receiver by the transmitter. The full protocol may be found in (Quinn 1993). Here, we shall describe only its most essential aspects.

In SSCOP, each message, or protocol data unit (PDU), carries a sequence number (or *seqno* for short) that is used for detecting missing PDUs and for delivering them in proper order. The transmitter maintains a counter VT(S) for assigning these sequence numbers and another counter VT(A) for keeping track of the seqno up to which all PDUs have been correctly acknowledged by the receive end. The receiver maintains two main counters, VR(R) and VR(H). VR(R) indicates that all PDUs with a lower seqno have already been received correctly. VR(H) indicates that the highest seqno seen thus far is VR(H)-1. That is, VR(H)>VR(R) indicates that there is a gap in the seqnos (i.e., certain PDUs were lost). Whenever a new gap develops, the receiver sets VR(H) to 1 plus the

[‡]This assumes that the signalling traffic is not shaped to look like a constant bit-rate (CBR) traffic.

highest seqno seen thus far, and alerts the transmitter by sending a USTAT (unsolicited status) message. In response, the missing PDU is retransmitted. All correctly received PDUs are placed in a *receive buffer* by SSCOP. Of these, the PDUs with a seqno less than VR(R) (i.e., in-sequence PDUs) become available for pick up by the MTP3 layer.

The transmitter periodically sends a POLL message to the receiver to enquire its status. In reply, the receiver sends a STAT (status or solicited status) message, which contains a list of all currently existing gaps. All missing PDUs are retransmitted in response to a STAT. The poll/stat mechanism is essential for the protocol in that the USTAT is dispatched only to report the first loss of a given PDU. Poll/stats also serve the purpose of "I am alive" signal between the peers. Unlike user PDUs, POLLs, STATs, and USTATs are never retransmitted. Both STATs and USTATs bring back VR(R), so that VT(A) can be advanced.

For flow control, the SSCOP receiver maintains a counter called VR(MR), which gives the maximum seqno up to which the receiver will accept the arriving PDUs. The transmitter maintains a corresponding counter called VT(MS). Every STAT and USTAT message conveys the current VR(MR) value to the transmitter so that VT(MS) counter can be kept in sync with VR(MR). The difference $VT(MS) - VT(S)$ gives the currently available *credit* to the transmitter. If VT(MS)-VT(S)=0, the transmitter ceases transmission of *new* user PDUs. The credit granting is done by updating VR(MR), but SSCOP does not specify how or when this should be done.

When VT(S) reaches VT(MS), the transmitter starts a timer T_{nc} (with a default value of 1.5 seconds). If T_{nc} expires before the no-credit situation is remedied, SSCF takes the link out of service. In the event of a credit-rollback (i.e., when the credit is taken away for an already transmitted PDU), arriving PDUs may fall outside the credit window and will be dropped. The proposed SAAL error monitor (Kant 1995c) ignores any retransmissions while the no-credit situation is in effect. In extreme situations, this may cause it to miss out on some real error phenomenon. It follows that the credit control scheme should not lead to long or frequent zero credit periods.

3 CREDIT ASSIGNMENT SCHEMES

In this section we propose two simple schemes for advancing the VR(MR) counter (and thereby granting the credit). However, before doing so, let us motivate an important requirement that any credit granting scheme must satisfy. Toward this end, note that the receive buffer holds both in-sequence and out of sequence PDUs. Typically, the in-sequence PDUs are picked up by level 3 using a periodic circular scan of all level 2s controlled by it. Thus, in the event of link errors or slow pickup by level 3, the receive buffer may become full. If the transmitter is assigned more credit than the available buffer space, more PDUs will continue to arrive and will be dropped. The resulting retransmissions will be considered as "errors" by the SAAL error monitor. Worse yet, if the PDU with seqno of VR(R) is dropped while the receive buffer is occupied by out-of-sequence PDUs, a deadlock will result since out-of-sequence PDUs cannot be delivered to level 3. To avoid these problems, *assigned credit should never exceed the available buffer space.* The two credit allocation schemes considered here are as follows:

(a) **Moving Credit:** In this scheme, whenever a PDU is picked up by level 3 from the receive buffer, VR(MR) is incremented by one. VR(MR) is initialized to an appropriate value, say N_m, which is chosen with the objective of keeping the link busy.
(b) **Fixed Credit:** This scheme involves two parameters, say N'_f and b_f. Here, VR(MR) is set to $VR(R) + N'_f$ at the time of sending every b_fth stat. VR(MR) is initialized to N'_f. N'_f is computed from considerations of the maximum rate at which the traffic can be accepted by the receiver.

In both schemes, VT(MS) can be initialized to ∞, since the first STAT will reset VT(MS) to the initial value of VR(MR). Note that the assignment VT(MS)$\leftarrow$VR(MR) happens for every STAT even in scheme (b); thus, the loss of a POLL or STAT has similar consequences for both mechanisms. The following list compares and contrasts these two mechanisms in other respects.

1. In scheme (a), updated VR(MR) is communicated to the transmitter by every stat, whereas in scheme (b), this happens on every b_fth STAT only. Thus, with $b_f > 1$, scheme (a) needs to grant credit for traffic generated over a smaller interval, and thereby requires a smaller credit and buffer size.
2. If a user PDU is corrupted by a random error, it will be reported almost immediately via a ustat. In scheme (a), this USTAT will increase the credit by the number of PDUs that were correctly received before the error. This results in a larger room for out of sequence PDUs before the corrupted PDU is re-received correctly, without requiring additional buffer space. That is, with $b_f = 1$, scheme (a) has a slight edge over scheme (b) in error situations.
3. Scheme (a) can automatically ensure that the granted credit never exceeds the available buffer space, since if VR(MR) is initialized to the receive buffer size (in messages), $VR(MR) - VR(R)$ always gives the number of available message slots in the receive buffer. In contrast, scheme (b) must explicitly adjust N'_f to achieve this property. In particular, if N_f is the desired credit amount without considering the receive buffer occupancy, one needs to set $N'_f = \max(N_f - O, 0)$ where O is the number of occupied message slots.[§]
4. In scheme (a), if an adequate amount of credit is granted initially, that credit is maintained continuously so long as level 3 keeps up with the PDU arrivals. That is, the scheme does not provide any flow control under normal conditions. In contrast, scheme (b) explicitly grants a fixed credit periodically, and thus can effect flow control even under normal conditions.
5. Since scheme (a) provides no flow control until the receive buffer is almost full, it results in an almost on-off control that becomes more sluggish as the receive buffer size increases. In contrast, scheme (b) begins to control traffic whenever it exceeds a desired threshold, independent of the receive buffer size.
6. As shown in section 4, the credit requirements must be somewhat overestimated to account of jitter in STAT arrivals and link propagation delays. This "error" becomes

[§]Thus, the parameter N_f is really the one that this paper is attempting to estimate, whereas N'_f is a local variable that is computed for updating VR(MR).

relatively less important as the credit allocation interval increases. In this sense, scheme (b) can achieve a more accurate flow control.

Item (4.) above points to a fundamental difference between the two schemes: Suppose that VR(MR) is initialized to a higher value than the desired credit amount. Then, under scheme (a), credit window plus the number of in-sequence PDUs in the receive buffer will remain invariant, thereby retaining the excess credit. On the other hand, scheme (b) advances VR(MR) relative to the current VR(R); consequently, any unused credit during an allocation interval will not propagate into the next interval, and each interval will only get the desired credit. (This property justifies the name "fixed credit" for scheme (b).)

The next section addresses the problem of estimating the parameters of these schemes (N_m for the moving credit scheme, and N_f, b_f for the fixed credit scheme). Although the determination of N_m (or N_f) directly yields the required receive buffer size in messages, specifying the size in bytes is a bit more involved and is also addressed.

4 CREDIT REQUIREMENTS AND BUFFER SIZING

Although the SSCOP standard does not contain any specific guidelines for credit allocation, it does provide a default size for the credit window (Appendix IV in (Quinn 1993)). This default size is chosen very generously so that the credit is almost never a limiting factor. This may result in large receive buffer size requirements and consequent delays. The analysis in this section comes up with smaller, and more realistic, numbers.

As discussed in section 1, level 2 and 3 processors may be engineered assuming that the signalling link carries less than 1 Erlang load. Thus, it is necessary to examine the parameterization problem without and with the processing capacity limitations. Sections 4.1 and 4.2 study these two cases. Section 4.3 then examines some issues in sizing the receive buffer.

4.1 Basic Credit Requirements

We start with situations where there are no link errors and estimate the minimum credit needed to ensure that the transmitter can indefinitely transmit PDUs at the full link capacity. It is clear that regardless of the scheme for updating VR(MR), the updated value is reflected on the transmit side (by the assignment of VT(MS)=VR(MR)) only when a STAT arrives. Thus, it suffices to compute credit requirements only for the fixed credit scheme, since the moving credit scheme becomes a special case with $b_f = 1$. The basic idea is that when a STAT arrives and updates VT(MS), the new value of VT(MS) must be large enough to allow the transmitter to keep going until the next credit granting instant.

Let T_{poll} denote the polling interval, δ_s a suitable measure of jitter in the arrival time of successive STATs. For example, one may choose δ_s as twice the standard deviation of T_a, the interarrival time of STATs on the transmit side. Assuming, for simplicity, that the queuing/processing delays experienced by successive POLLs (and STATs) are iid (independent, identically distributed), the jitter in the interarrival time of successive b_fth STATs is $\sqrt{b_f}\delta_s$. Thus, the transmitter should have enough credit to keep transmitting for the $b_f T_{poll} + \sqrt{b_f}\delta_s$ seconds. Let r_{pdu} denote the maximum rate at which user PDUs can be transmitted, and $CT_{\min}$ the required minimum credit. Then

$$CT_{\min} = r_{pdu}(b_f T_{poll} + \sqrt{b_f}\delta_s) \tag{1}$$

Let τ_u denote the end to end transit delay of POLLs and STATs. Because of this delay, the transmitter and receiver don't have the same view of the SSCOP state. Let T_u denote the maximum number of PDUs that can be transmitted during the time τ_u. Then, at the time the receiver sends a stat, the transmitter could already be ahead by T_u, i.e., VT(S) may already be $VR(R)+T_u$. By the time the STAT reaches the transmit end, VT(S) could have advanced by another T_u. Thus, if a credit of CR PDUs is granted by the receiver when sending the stat, it will result in VT(MS)=VR(R)+CR on the transmit side. Since $VT(S) = VR(R) + 2T_u$ when the STAT arrives on the transmit side, the actual credit granted to the transmitter is only $VT(MS) - VT(S) = CR - 2T_u$. It follows that the minimum credit on the receive side, $CR_{\min}$, should be set to $CT_{\min} + 2T_u$. That is,

$$CR_{\min} = r_{pdu}(b_f T_{poll} + \sqrt{b_f}\delta_s + 2\tau_u) \tag{2}$$

Let us now consider the estimation of the variables used in the above equation. Let n_{cell} denote the average PDU size in cells at the ATM layer, and r_{cell} the link speed in cells/sec. Then r_{pdu} can be estimated as

$$r_{pdu} = [r_{cell} - 2/T_{poll}]/n_{cells} \tag{3}$$

For a 1.5 Mb/sec SAAL link with $T_{poll} = 0.1$ sec, $r_{pdu} \approx 3600/n_{cells}$. The parameter τ_u includes one-way link propagation delay (t_p), plus processing, queuing, and transmission time of management messages (t_u). That is, $\tau_u = t_p + t_u$.

Because of frequent physical link rearrangements, it is undesirable to use the actual link length in estimating t_p. In fact, current systems do not have any built-in mechanism to keep track of actual link lengths. If the chosen t_p is larger than the actual, excess credit will be allocated to short links. Similarly, if the chosen t_p is smaller than actual, long links may be starved of credit. Of these two choices, the first one appears preferable since both credit control schemes will cut-down credit if the receive buffer fills up. Thus, we propose estimating t_p using, say, 90 percentile link length for the network of interest. For example, in a regional network, a benchmark length of 2000 miles will give $t_p = 20$ ms (using 0.01 ms/mile rule).

Estimation of t_u and δ_s requires computing delays experienced by management messages. The SSCOP specification suggests priority queuing for transmission with highest priority accorded to management messages. This is desirable from a credit allocation perspective, since one doesn't need as much credit over-allocation to allow for statistical fluctuations. However, some implementations may not use priority queuing. Similarly, the delay suffered by the PDU at lower layers is implementation dependent. For example, if the ATM layer takes cells from CP in large batches rather than individually, it may add significantly to the delays. In general, analytic estimation of t_u or δ_s is intractable, and one may have to use a reasonable upper bound determined from the architectural details. (However, (Kant 1995c) provides an approximate method for estimating SSCOP level delays.)

Let us now consider the impact of link errors on the credit requirement. A single link error (which results in a USTAT being generated almost immediately) results in about $2\tau_u$ seconds delay before the retransmitted PDU will arrive again. During this time, VR(R) does not change, which can reduce the granted credit by as much as $2\tau_u r_{pdu}$ because of the reception of out-of-sequence PDUs. Thus to ensure that a single link error does not result in a zero-credit (or dead) period, it is necessary that $CT > 2\tau_u r_{pdu}$. Assuming that δ_s is small, this condition, along with equation (1) implies that $b_f T_{poll} > 2\tau_u$. With moving credit scheme, this condition may not be satisfied if the link is too long or the management messages experience significant queuing delays. The fixed credit scheme with $b_f > 1$ may be preferable in this case.

Links often suffer from severe error bursts such that during the error burst almost all PDUs are lost or corrupted. The SAAL error monitor is designed to ride over error bursts of length up to $t_b = 0.4$ secs. Thus, following the error burst, the separation between VR(R) and VR(H) could be $t_b r_{pdu}$, and grow to $(t_b + 2\tau_u)r_{pdu}$ until the first retransmission arrives. The utility of giving the transmitter enough credit to continue transmission over this entire period is questionable, particularly, since the credit granting STATs would themselves get lost. However, if the user PDUs are long, it is possible that polls/stats get through, but the user PDUs do not. The credit requirements in this case are:

$$CR_{\max} = (t_b + 4\tau_u)r_{pdu} \tag{4}$$

where the additional $2\tau_u$ factor results from the difference between CT and CR parameters. If the credit is set according to this equation, the condition $CT > 2\tau_u r_{pdu}$ implies that $t_b > 0$; i.e., the dead periods cannot occur in case of single errors.

This analysis shows that in the absence of any processor capacity limitations, the parameters N_m and N_f can be set somewhere between the limits $CR_{\min}$ and $CR_{\max}$. A larger credit requires more buffer space and results in longer delays, but does not constrain the transmitter as much under error conditions. Given that link errors occur rarely, it appears that the disadvantages of choosing a credit larger than $CR_{\min}$ outweight its advantages.

Let us now examine the choice of the allocation interval b_f for the fixed credit scheme. Clearly, it is really the duration $b_f T_{poll}$ that is of interest here. The default value of T_{poll} is 100 ms, and it has been shown in (Kant 1995b) that this value of T_{poll} is adequate even at very high link speeds. Thus, we can regard T_{poll} as a constant and regard b_f as the parameter of interest.

As b_f increases, less control is exercised on the arriving traffic over short periods. For example, with $b_f = 10$, enough credit must be granted to last for 1 second, and the receiver will be obligated to accept the incoming traffic during this time even if the conditions change and require lowering the incoming traffic rate. A large b_f also requires a large receive buffer to hold the traffic in the event of a level 3 congestion. However, a large b_f has some advantages as well. First, a larger b_f results in less reduction in credit due to a single error; in fact, for very long links, a large b_f may be necessary to satisfy the condition $b_f T_{poll} > 2\tau_u$. Second, a large b_f results in more accurate credit allocation. To see this, note that the δ_s term in equation (2) can be regarded as "error", since it merely accounts for statistical fluctuations in the arrival times of successive STATs. The same is true of the t_p and t_u terms, since, as discussed above, they need to be estimated from upper bounds rather than actual delays. Thus, the worst-case relative error, denoted $\eta_{\max}$, is given by:

$$\eta_{\max} = \frac{\delta_s \sqrt{b_f} + 2\tau_u}{b_f T_{poll}} \tag{5}$$

It is clear that $\eta_{\max}$ decreases as b_f increases, which means that credit allocation falls more in line with what it is designed for. This aspect becomes even more important if the credit must be limited due to processor capacity limitations.

In the broadband environment, the ATM policing function will ensure that the long term traffic rate on the signalling channel does not exceed its peak bandwidth; however, short-term traffic bursts will still be allowed. For example, the leaky-bucket policing mechanism has allowable burst length as one of its parameters. It may be reasonable to allow the bursts to last several hundred milliseconds for the signalling channel. However, this implies that the credit allocation duration should also be at least a few hundred milliseconds long.

From the above considerations, a b_f in the range 3-5 seems appropriate. With $\tau_u = 0.025$ secs and $\delta_s = T_{poll}/10$, the maximum error $\eta_{\max}$ is 22.4% for $b_f = 3$, and 14.5% for $b_f = 5$.

4.2 Impact of Level 3 Capacity on Credit Allocation

If the level 3 processor cannot handle the full engineered rate of the SAAL link, the allocated credit N_f must be less than $CR_{\min}$ for the fixed credit scheme. (The moving credit scheme, as stated in section 3 does not consider processing ability and thus N_m is not affected.) Let r'_{pdu} denote the actual traffic rate that level 3 can handle. Then the required credit N_f can be computed by simply substituting r'_{pdu} for r_{pdu} in equation (2). That is,

$$N_f = r'_{pdu}(b_f T_{poll} + \sqrt{b_f}\delta_s + 2\tau_u) \tag{6}$$

The appropriate technique for estimating r'_{pdu} depends on the precise details of the situation involved. Following are two simple examples illustrating how to determine r'_{pdu} for a new SAAL link:

1. Suppose that the signalling needs or processor capacity limitations indicate that only a fraction ξ of the full bandwidth need be supported. Then, $r'_{pdu} = \xi r_{pdu}$.
2. Suppose that the average occupancy of the level 3 processor without the new link is ρ, which is less than the desired engineered load ρ_e. Then, $r'_{pdu} = (\rho_e - \rho)/s_{l3}$, where s_{l3} is the average processing time of a message by level 3.

In telecommunication networks, nodes often employ active redundancy for enhanced reliability. For example, with quasi-associated signalling, STP–STP links are engineered for only 0.2 Erlang traffic under normal conditions, so that a double failure only results in 0.8 Erlang load. In such an environment, the level 3 (and other) processors will also be engineered in a similar way. Thus, in estimating r'_{pdu}, we need to specify the operating conditions of the nodes on either end of a SAAL link. If the credit allocation is done based on normal conditions, the granted credit will be inadequate under failure scenarios. On the other hand, if the credit allocation assumes a double-failure scenario, the credit granted under normal conditions will be 4-times the expected value which will allow a substantial surge in traffic before the flow-control goes into effect. Of these, the latter method seems more appropriate as it does not restrict the traffic unnecessarily. A third option is to vary the granted credit appropriately as a side effect of TFR/TFP/TFA procedures.

With limited level 3 capacity, the flow-control proposed here will prevent level 3 from going into SMH congestion; however, the transmitting end of the flow-controlled link will perhaps go into congestion as a result. Consequently, congestion control procedure will be triggered and reduce the traffic going over this link. As discussed in section 1, this is a better response than letting SMH congestion occur. Finally, neither of the proposed schemes causes a credit rollback, and within reasonable limits, they are unlikely to lead to long enough dead periods to let T_{nc} timer expire. Thus, the schemes should work well even for $r'_{pdu} << r_{pdu}$.

4.3 Impact of Message Size and Receive Buffer Sizing

SSCOP requires that the credit be specified in PDUs. However, equation (3) involves n_{cell}, the average PDU size in units of cells. Estimating n_{cells}, in turn, requires detailed knowledge of message lengths so that the overhead of padding and headers could be computed correctly (Kant 1995a). Since actual message sizes could be different than assumed, the credit allocation may be inaccurate. In particular, if the assumed message size is less (more) than actual, r_{pdu}, and hence the the granted credit is overestimated (underestimated). Of these two cases, credit overestimation is perhaps preferable as it avoids unnecessary flow control. Since a fill-up of the receive buffer will eventually cut-down the credit, the flow control will still take effect eventually, although somewhat belatedly.

Depending on the implementation details, the receive buffer may be sized either in terms of messages or bytes. With message sizing, the receive buffer size can be chosen to be equal to the credit amount (N_m or N_f, depending on the scheme). Unfortunately, message sizing necessarily implies that the buffer space will be allocated dynamically depending on the sizes of the messages to be held. In contrast, byte sizing allows static allocation of buffer space and is usually preferred. With byte sizing, there are two issues to consider:

1. Determining buffer size to be allocated (needed either at design time or when the signalling channel is set up.)
2. Determining available buffer space in terms of messages whenever VR(MR) is to be updated (needed if the receive buffer holds some in-sequence messages at this time).

Let us start with issue (1). Suppose that there are no capacity limitations at level 3. In this case, equation (2) could be used for credit allocation for both schemes. The same equation could also be used for buffer sizing, provided that we consider r_{pdu} in the units of bytes/sec. In particular, a 1.5 Mb/sec SAAL link can carry at most $3600 \times 48 = 172$ Kbytes/sec of user traffic. Thus, irrespective of the quality of estimates of δ_s and t_u, or the prevailing message sizes, the receive buffer size computed using $r_{pdu} = 172$ Kbytes/sec in equation (2) cannot fall short so long as no backlog develops in the receive buffer. Now, if the backlog does develop, both credit schemes will reduce allocated credit by the number of backloged PDUs. (In the moving credit scheme, this adjustment occurs automatically, whereas in the fixed credit scheme, the credit must be explicitly reduced.) However, if the messages transmitted during periods of backlog are longer, the link may carry more than the allowed number of bytes. To illustrate this, suppose that during one credit allocation interval, the link can carry 100 average sized messages. Now, if the receive buffer already contains 90 average sized messages, the allocated credit will be only 10. Because of their small number, it is possible that these 10 messages have a significantly higher average size. They will still be carried by the link; however, the receive buffer will run out of space for them. This problem can be addressed by a slight increase in the buffer size. In fact, since the credit allocation and buffer sizing already assumes rather extreme situation with respect to end-to-end transit delays, an increase may not even be necessary.

If level 3 capacity limitation does exist, we can use a similar approach; i.e., compute the buffer size using equation (6) with r'_{pdu} converted from messages to bytes by using some average message size, say s_m. Let s_{ma} denote the prevailing average message size in the network. Clearly, we want $s_m > s_{ma}$ to avoid buffer overflows. However, s_{ma} may increase over time but the growth may be difficult to predict. Thus, if level 3 capacity is not significantly lower than the link bandwidth, computing buffer size without regard to level 3 capacity is still the best approach. Otherwise, an automatic adjustment of credit and/or buffer size is needed in order to avoid the need for intervention from operation systems. This involves keeping track of current average message size s_{ma}, and must be implemented as an enhancement to SSCOP, since the current SSCOP standard does not keep this information.

Let us now consider issue (2). The easiest approach is to work with message units only. That is, we initialize the available buffer to the credit amount (N_f or N_m, depending on the scheme), and measure available space in the units of messages. This will work so long as the receive buffer is sized conservatively.

5 CONCLUSIONS

The paper proposed two credit control schemes for SAAL links and studied their parameterization. It is clear from this discussion that both schemes can provide effective flow control. The moving credit scheme is somewhat simpler to implement, but does not provide the rate control provided by the fixed credit scheme. Instead, as level 3 starts to fall behind, the PDUs accumulate in the receive buffer until the receive buffer is full and the credit drops to zero. The fixed credit scheme (with $b_f > 1$) also allows longer traffic bursts in a broadband environment and a smoother control over traffic if the allowed traffic is much less than the link capacity; however, this is at the cost of increased receive buffer space. Overall, it appears that the fixed credit scheme is somewhat superior.

One issue not addressed in this paper is the interaction of congestion control and flow control mechanisms. With the national congestion control option, the congestion control mechanism will selectively throttle messages based on their congestion priority. Thus, if some messages are much longer than others, the average message size may change drastically during congestion periods and cause problems of buffer overflow, overcontrol, or undercontrol of traffic. This may force a dynamic calculation of average message size for flow-control purposes, as suggested above.

ACKNOWLEDGMENTS

I am grateful to J.R. Dobbins for his help in weeding out errors in the design and analysis of the credit control schemes presented in this paper. Discussions with and comments from G. Ganguli and A. Jacob were also helpful in designing the schemes.

6 REFERENCES

Quinn, S. (ed.) (1993), "BISDN - ATM Adaptation Layer - Service Specific Connection Oriented Protocol", TD PL/11-20C Revl, 1993.

GR-2878, "Generic Requirements for CCS Nodes Supporting ATM High-Speed Signalling Links", GR-2878-CORE, Bellcore, Oct 1995.

Kant, K. (1995a), "Comparison of achievable throughputs of SAAL and MTP2 Protocols", unpublished memo.

Kant, K. (1995b), "Analysis of Delay Performance of ATM Signaling Links", Proc of INFOCOM 95, Boston, MA, April 1995.

Kant, K. (1995c), "An Error Monitoring Algorithm for ATM Signaling Links", in *Data Communications and their Performance*, edited by S. Fadida and R. Onvural. (Proc. of sixth Intlĉonfôn the Performance of Computer Networks, Oct 1995, Istanbul), pp 367-381.