On the Duality of Rate-based and Credit-based Flow Control

Hugh C. Lauer

TR95-03 December 1995

Abstract

Two forms of flow control for ATM (Asynchronous Transfer Mode) networks are examined, namely rate-based and credit-based flow control of ABR (Available Bit Rate) traffic. Under certain assumptions, these two are shown to be duals of each other. That is, the average traffic flow of a rate-based network can be achieved by controlling buffer space of a credit-based network. Similarly, the buffer requirements of a credit- based network can be achieved by controlling the rates in a rate-based network. Using the duality, it can be shown that several claimed advantages - in which some feature or attribute is claimed to be available in one form of flow control but not the other - are not advantages at all and, in fact, have corresponding implementations in the other. However, the duality is not perfect, and some asymmetries remain between these two forms of flow control. Some observations are offered to reduce the differences between these in practical networks.

ACM SIGCOMM ‘95 Conference

201 Broadway, Cambridge, Massachusetts 02139

Revision History:–

1. First version, TR 95-03, *January 26, 1995*

1.0 Introduction

ATM (*A*synchronous *T*ransfer *M*ode) network technology is rapidly gaining acceptance as the high speed network technology for a wide variety of next generation digital communication problems, ranging from traditional data processing and computer communications to multi-media, real-time control, home area networks, etc. ATM network technology provides one basic framework supporting many sizes of networks, from the wide area, to the local area, to the desk area, and even down to the backplane. ATM networks can also support more and broader classes of communication traffic than were possible with previous local area network technologies.

The connection-oriented nature of ATM networks offers a tantalizing opportunity — to be able to support several different kinds of traffic in the same network at the same time, especially real-time and non-real-time traffic. In modern industrial systems, we would like to be able to support in the same network

- “hard real-time” applications for plant control and operations,
- continuous and interactive audio and video both as sensors in the actual plant operation and for human activity such as training, monitoring, and surveillance, and
- traditional forms of network and distributed computing using protocols such as TCP/IP, Appletalk, etc.

The challenge is, on one hand, how to keep the non-real-time traffic from interfering with the real-time traffic and, on the other hand, how to let the non-real-time traffic effectively use the remaining capacity of the network after the real-time obligations are met.

In our context, the real-time traffic may be extremely bursty. For example, a motor suddenly starts and needs control information from sensors at the other end of a rolling mill. An intelligent video camera detects an anomalous situation and suddenly sends forth a burst of very high priority video. Even in a simple office workstation environment, a user may open an interactive video window and begin viewing, pausing, and scrolling, thereby taking up a significant fraction of the total network bandwidth in bursts.

Designing a network and pre-allocating resources to support the worst case bursts of real-time traffic is beyond the scope of this paper (see, for example [Ferrari90] or [Zhang93]). The problem of interest in this paper is how to use the remaining “available bit rate” of the network for the more traditional traffic between such bursts. This traditional traffic has come to be called ABR or *A*vailable *B*it *R*ate traffic. It closely resembles the kind of communication among, say, workstations and servers in a local area network or among members of the Internet. The goal of the network is to deliver the entire remaining bandwidth on demand, with no advance reservation or specification of traffic characteristics, with no holding back of network resources in anticipation of other traffic that might not materialize, but subject only to “fair” sharing among all such users and applications on a packet-by-packet or message-by-message basis. That is, the network should offer to ABR clients approximately the same kind of sharing of network resources as is already implemented by local area networks such as Ethernet or Token Ring and by the default scheduling policies of many multi-protocol routers.

The problem of supporting ABR traffic is, in fact, the problem of controlling its flow, so that it does not overflow buffers or clog network resources, especially when the “available bandwidth” suddenly changes as a result of high priority bursts. Over the past several years, two approaches to flow control have been vigorously debated in the ATM Forum, an industry-sponsored organization dedicated to the promotion of ATM network technology and to recommending standards for interoperability. One form, called *rate-based* flow control, is based on techniques used in frame relay networks and depends upon feedback from the network to each source to cause those sources to slow down or speed up [Siu95, Roberts94a, Roberts94b]. The other, called *credit-based* flow control, depends upon feedback indicating the ability of a network node or destination to accept additional data before overflow [Kung94, Hunt94]. These two approaches have been characterized as being fundamentally different in philosophy and have fundamentally different merits, advantages, and disadvantages.

The purpose of this paper is to explore the hypothesis that they are not different. Under certain assumptions, we will show that rate-based flow control can be mapped into credit-based flow control and vice versa. The behavior obtainable with one approach has a corresponding behavior in the other approach, and the buffer requirements of one are the same as those of the other. This creates a duality between them, the consequence of which is that advantages claimed for one approach turn out to be also available in the other, although not necessarily so obviously. Unfortunately, however, the duality is not perfect. There still turn out to be some differences that require discussion.

The reminder of the paper is organized as follows. Section 2 presents background and terminology. Section 3 shows that the average rate of transmission can be controlled in a credit-based system by controlling the amount of buffer space and, conversely, that the amount of buffer space required to avoid data loss in a rate-based system can be controlled by adjusting the rates of transmission. These results are applied in Section 4 to show similar behavior in both kinds of systems. Section 5 considers several areas where the duality breaks down. Finally, Section 6 presents some observations and conclusions.

2.0 Background and Terminology

In this section, we first clarify our terms, then summarize an abstraction of credit-based flow control, an abstraction of rate-based flow control, then a combined approach.

2.1 Terminology

An *ATM network* is a network of nodes and bidirectional links. Non-terminal nodes are called *switches*, and terminal nodes are called *end-systems*. The unit of transmission in an ATM network is the *cell* — i.e., a data element containing a five-byte header and a forty-eight byte payload. A *connection* is a logical channel of communication between an application at one network node and an application at another; it provides for transmission of information in *one* direction from the first to the second. This direction is called the *forward* or *downstream* direction. The opposite direction is called the *backward* or *upstream* direction. Normally, connections between applications come in pairs, one in each direction, so that the applications may engage in conversations, provide feedback to each other, etc. Logically, and for the purposes of this paper, these two connections are separate and independent. They may have quite different traffic characteristics and quality of service requirements. In this paper, we restrict our attention to one of this pair of connections, and in particular we address both data and control information flowing in the forward direction and control information flowing in the backward direction.

A *path* is the sequence of network nodes over which a connection is implemented. It is a fundamental rule in ATM networks that all cells of a connection are carried in order over the same path from start to finish. Although it is possible for cells to be lost due to transmission errors or to be discarded in response to congestion, they will not be delivered out of order. A path may be partitioned into *control segments*, each of which is a subsequence of the nodes of the path. The first node of any control segment is its *virtual source*, and the last node is its *virtual destination*. If the path comprises more than one control segment, then the virtual destination of one control segment becomes the virtual source of the next. In the following, when we say *source*, we will normally mean *virtual source* unless it is obvious from the context. The *original source* is the virtual source of the first control segment of a path. Similarly, when we say *destination*, we will normally mean *virtual destination* unless it is obvious from the context. The *final destination* is the virtual destination of the last control segment of the path.

2.2 Credit-based Flow Control

In a credit-based scheme, a source transmits data as fast as it can over a network link, but only as many cells as it knows the destination can receive. In particular:—

- Each source counts data cells transmitted, and each destination counts data cells received. In addition, each destination keeps track of the number of cells of buffer space allocated to and occupied by each connection.
- Periodically, after a destination has consumed a number of cells received on a connection (or has forwarded them to the next control segment in its path), it generates a *credit cell* for transmission to the source over the path in the backward direction. This credit cell specifies the *credit limit* indicating the highest numbered cell that will not overflow the buffer space allocated to that connection.

  I.e., if at time t_0, the destination had consumed or forwarded n_0 cells, and it has allocated a total of B cells of buffer space to the connection, it would return a credit limit of $n_0 + B$. This credit limit must accommodate cells that are already received but not yet consumed, cells in flight, and cells yet to be transmitted.[1]
- Upon receiving a credit cell, the source transmits data at any rate, but when its cell count reaches the credit limit in the most recent credit cell, it stops.

---

[1] The various credit proposals have different representations of the credit cell, but all are equivalent.

- Periodically, the source sends a control cell in the forward direction specifying the serial number of the next data cell to be transmitted on that connection. This is used by the destination to resynchronize its cell counts with the source and provides a way to recover from lost data cells.

In the most widely discussed credit-based schemes, each physical link of a path is its own control segment, so that there are no intermediate nodes between any virtual source and destination. However, this restriction can be relaxed, as will be shown below. The net effect is that an original source transmits data to a final destination in short, rapid bursts, but only as much as that final destination and each intermediate node can absorb at any one time. The claimed advantages are that credit schemes respond very rapidly to changes in the available bandwidth of the network, that they never deliberately discard data due to congestion, and that they make maximum use of the available network bandwidth. Credit-based schemes are criticized, however, for their large buffer requirements and for their complexity, especially of the perceived need for separate queues for each connection.

2.3 Rate-based Flow Control

In a simplified form of the rate-based scheme, the source determines the rate at which to send cells from the presence or absence of feedback from the destination, as follows:—

- Periodically, the source inserts a Resource Management cell (*RM* cell) into the data stream, indicating both its actual and desired rates of transmission. These *RM* cells are a form of in-band signalling and their frequency is a parameter of the connection.
- When a destination receives an *RM* cell, it replaces the desired rate field with a value representing an explicit rate at which it is able or willing to receive the data, then retransmits that *RM* cell back to the source over the same path.
- Each intermediate node in the path between the destination and the source may also reduce the value of the desired/explicit rate field to account for congestion of network resources under the management of that node.
- When the source receives an *RM* cell from downstream, it adjusts its rate according to the explicit rate and any other congestion information in the *RM* cell.
- In the absence of receiving any *RM* cells from downstream, the source it must reduce its rate of transmission according to a function determined at the time the connection is established. This a failure recovery mechanism for the case of *RM* cells being delayed or lost due, for example, to congestion somewhere downstream.

The nodes of the network maintain several other parameters for controlling the data rate of the connection. In particular, the source will never request or transmit at a rate exceeding the *Peak Cell Rate* established for the connection, and the destination and intermediate nodes will never reduce the explicit rate below a *Minimum Cell Rate* for the connection. The destination or any intermediate node may generate and transmit extra *RM* cells in the backward direction with new values of the explicit rate to respond to sudden changes in the demand for network resources. Similarly, a source may transmit extra *RM* cells in the forward direction, for example, after an idle period to request an increased rate.

The currently proposed standard for the reduction function f when no *RM* cells arrive is

$$f(R) = R * (1 - 1/RDF), \tag{1}$$

where R is the current rate and RDF is a constant parameter [Barnhart94b]. Similarly, it is proposed that a source be able to increase its rate only gradually, by adding a constant *AIR* to its current rate after receiving an *RM* cell from downstream.

The claimed advantages of rate-based schemes are that they are simpler to implement and that they require less buffer space. They are criticized for their lack of responsiveness to transients in network load and the fact that they deliberately throw away data of ABR connections in congested situations.

2.4 Combined Rate- and Credit-based Scheme

In [Zheng94], a combined scheme is described using a single *RM* cell format to carry both rate and credit information in both directions. The format of this *RM* cell is the same as the format of the *RM* cell of the rate-based method, but extended with an extra field. When the *RM* cell is travelling through the network in the forward direction, this extra field represents the serial number of the next cell to be transmitted. When it is moving in the backward direction, this field represents the credit limit for the connection.

In this scheme, a source sends *RM* cells periodically, just as in the rate-based method. When a destination receives an *RM* cell, it updates the desired rate with an explicit rate and also replaces the serial number with a credit limit reflecting the buffer space available to the connection. As this *RM* cell makes its way through the network, any switch along the path can reduce either field to take account of congestion. (Either the explicit rate or the credit limit, but not both, can be set to some value indicating that it should be ignored.)

When the source receives such an *RM* cell, it behaves exactly as a rate-based source, but stops transmitting as soon as it reaches the credit limit. Obviously, if the explicit rate field has a value of “ignore,” then the source may transmit as fast as the peak cell rate. Similarly, if the credit limit field has the value of “ignore,” it may transmit as many cells as the rate-based algorithm and parameters permit.

3.0 Duality between Rate and Credit Methods

In this section, we demonstrate that with credit-based flow control, it is possible to control the average rate of transmission by managing the size of the buffer allocated to the destination. Similarly, with rate-based flow control, it is possible to control the buffer space required for a connection at the destination by managing the explicit rates fed back to the source. Within the limits of the assumptions, this represents a duality between the two schemes. They require the same buffer space to support exactly the same average rate of data transmission across a control segment, and therefore it is possible to achieve the same behavior of an ABR connection requiring the same network resources with either scheme.

3.1 Rate Control in a Credit-based Network

Consider a control segment of a network with credit-based flow control. By the algorithm for returning credits from the destination to the source, it is clear that credits cannot flow backward any faster than the destination can consume cells (or forward them to the next control segment). Thus the ultimate control of the transmission rate of the source is bounded by the consumption rate of the destination.

However, let us consider a slightly different problem. Suppose that the destination is capable of consuming or forwarding cells as fast as they arrive, but suppose for some reason it wishes to limit the source rate to a lower value. We do this by limiting the number of cells of buffer allocated to the connection. In particular, suppose that *B* cells of buffer space have been allocated, and suppose that the round trip time for a signal on the control segment is *rtt*. Also, suppose that the destination returns one credit cell for each γ cells forward or consumed.[2] Figure 1 shows an example of part of the time line for the connection.

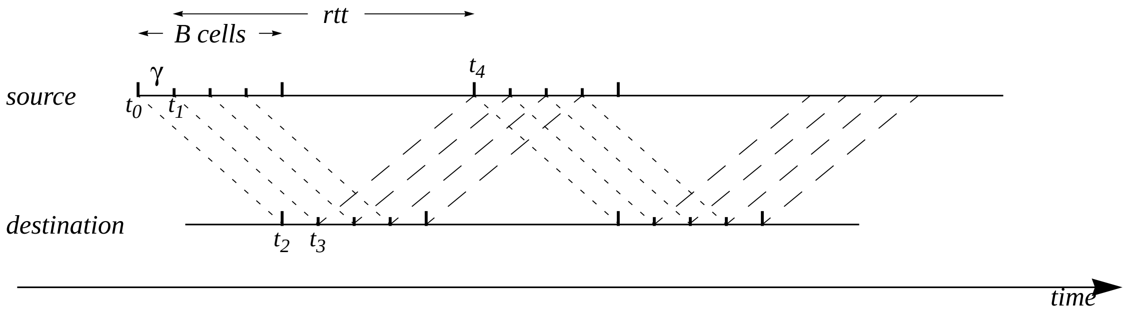

Figure 1. Time Line of a Credit Connection

At time t_0, the source has in hand enough credits to send *B* cells. It transmits cells as fast as it can, and by time t_1, it has transmitted γ cells. The first cell and the γ-th cell are consumed by the destination at times t_2 and t_3, respectively. By time t_3, the destination has consumed γ cells, and it therefore sends back a credit cell containing a new credit limit $B + \gamma$. This credit cell arrives back at the source at time t_4. Meanwhile, the source continues to transmit data cells until it runs out of its original credit limit *B*, and the destination continues to send back credit cells for every γ cells consumed or forwarded.[3]

We can now estimate a lower bound on the time t_4. Let *PCR* (peak cell rate) be the fastest possible rate of transmitting cells over this connection. *PCR* cannot exceed the bandwidth of the physical link, but the actual rate of transmitting the first γ cells is likely to be slower than this. Therefore,

$$t_1 \geq t_0 + \gamma / PCR. \qquad (2)$$

---

[2] Note that the parameter γ is the parameter *N2* in Kung's papers. Note also that *B* must be at least as large as γ so that the source does not starve for lack of credits while the destination is waiting for more cells before sending credits.

[3] Time t_4 may be before, at, or after the time that the *B*th cell following t_0 is transmitted. For clarity in the figure, it is shown as after. This does not affect the analysis.

Because the destination sends credit cells only after receiving and consuming γ data cells, it will not send a credit until at least time t_3. The source may not send its ($B + 1$)-th cell until it receives this credit. By the definition of *rtt,* we see that

$$t_4 \geq t_1 + rtt. \tag{3}$$

Combining inequalities (2) and (3), we see that the minimum time to transmit B cells of this example is

$$t_4 - t_0 \geq rtt + \gamma / PCR.$$

Thus, the rate of transmission R, averaged over the interval $[t_0, t_4]$, is

$$R = \frac{B}{t_4 - t_0} \leq \frac{B}{rtt + \gamma / \mathrm{PCR}}. \tag{4}$$

Using the same analysis, it can also be seen from Figure 1 that each time the source transmits a cell that causes a credit cell to be returned, the average rate of transmission over the next B cells is bounded by the right side of the inequality (4). This is even true if *rtt* is very small, so that it takes longer to transmit B cells than it does to get back the credit from the first group of γ cells.

Inequality (4) shows that the average rate of transmission of a source in a network with credit-based flow control is directly proportional to the amount of buffer space allocated to that connection at the destination. This means that to control the average rate, the destination need only control the amount of buffer space. In particular, to limit the rate of transmission to a maximum value of R, it is sufficient to set the buffer size to

$$B = R * (rtt + \gamma / PCR) \tag{5}$$

cells.[4] While the source may transmit in bursts, its average rate over any group of B cells can be no greater than R. Of course, if the destination cannot consume or forward cells so fast, the credit-based flow control algorithm automatically limits the actual flow of credits to the actual rate of consumption.

Moreover, if the destination wishes to constrain the source to a time-varying rate $R(t)$, then it need only adjust the buffer allocation to a time-varying amount

$$B(t) = R(t) * (rtt + \gamma / PCR). \tag{6}$$

Note that this analysis is based on, but slightly different from, Kung's analysis for adaptive buffer allocation in credit-based networks [Kung94]. In a typical local area network in which it is desired to carry ABR traffic as fast as possible after all higher priority commitments have been met, it is well known that allocating

---

[4] This is not practical for very low data rates. For example, suppose the desired data rate is 1 megabit/second. Suppose also that the round trip time in a local area network having 155 megabit/second links is 15 µseconds (about five cell times) and γ is 8. Then B would be about *16 bits*, or less than 4% the size of one ATM cell.

$$PCR * rtt + \gamma$$

cells is sufficient to guarantee that the connection will get all of the bandwidth if there is no other demand for it. In the case of long physical links that are shared among many connections, however, this would result in enormous buffer requirements. Thus in the adaptive buffer algorithm, Kung shows that the allocation to each connection can be reduced to a time-varying value

$$B(t) = R(t) * rtt + \gamma \qquad (7)$$

without slowing down any connection, *provided* that the destination can respond fast enough to changes in the aggregate transmission rate of the connections sharing the link. Observe that Equation 7 is slightly larger than Equation 6 and works for all values of R, rtt, and γ. While Equation 6 is sufficient for constraining a rate, Equation 7 is sufficient for guaranteeing one.

Note that in the analysis of this section, the round trip time of a signal *rtt* figures heavily. In order for a destination to control the rate of a source, it must "know" the value of *rtt* fairly precisely. When credit-based flow control in its native mode (i.e., when not trying to simulate rate-based flow control), *rtt* needs only to be "known" at the time the connection is established, and then only to allocate a sufficient number of buffers. From a practical point of view, it is often easier to overallocate than to try to get an accurate measure of *rtt*.

3.2 Buffer Requirements in a Rate-based Network

Suppose that a destination in a network with rate-based flow control is capable of consuming or forwarding cells of a connection at a rate $R(t)$. Suppose that *RM* cells are transmitted in the backward direction after every γ data cells received. Then when it is necessary to send an *RM* cell at time t, the destination simply sets the explicit rate field to $R(t)$.

Suppose the rate had been steady at $R(t_0)$ for at least *rtt* seconds before time t_0, and suppose that at time t_0, the rate of consumption or forwarding of cells drops drastically from $R(t_0)$ to zero. (This can happen, for example, if a high priority connection suddenly starts transmitting over the immediate downstream physical link, consuming all of the link bandwidth and thereby depriving ABR connections of any available bandwidth over a non-negligible period of time.) If it were possible to send an *RM* cell instantly to notify the source of this change in available bandwidth, a change in the source rate could not be felt at the destination at least until time $t_0 + rtt$. During this time, at least

$$rtt * R(t_0)$$

cells could arrive. Thus, there must be at least

$$B(t_0) = R(t_0) * rtt$$

cells of buffer space available in order to absorb such a sudden transient.

In practical networks, it is usually not possible to know $R(t)$ for the connection at all times, but instead it must be derived from the behavior of the network. For example, if two successive *RM* cells are received by the destination at times t_1 and t_2 after n_1 and n_2 cells had been consumed or forwarded, respectively, then $R(t_2)$ can be estimated to be

$$R(t_2) = (n_2 - n_1) / (t_2 - t_1).$$

Suppose that an *RM* cell was received at the destination and returned to the source at time t_1, just before the drastic reduction in the rate of consumption or forwarding occurred. If γ is the maximum number of data cells between *RM* cells, then as many as γ data cells would arrive at rate $R(t_1)$ before a new *RM* cell would be received at time t_2. The destination would set the explicit rate field of this new *RM* cell to zero before returning it to the source, since no data cells were consumed in $[t_1, t_2]$. However, before this rate can take effect at the source, a further $rtt * R(t_1)$ cells will arrive at the destination. Thus, the total buffer required at the destination to absorb the sudden change in rate without cell loss is

$$B(t_1) = R(t_1) * rtt + \gamma, \tag{8}$$

the same number as specified by Equation 7 and slightly more than that specified by Equation 6 in the credit-based system.

Equation 8 shows that to prevent cell loss due to transients in the network, a rate-based network requires an allocation of buffer space to each connection that is roughly proportional to the transmission rate. This is far greater than the amount of buffer space normally associated or claimed for rate-based flow control. To reduce the requirement to a particular rate, it is necessary to isolate the connection from the effects of transient behavior of other connections. This is possible in some networks, as discussed in Section 5.3. However, in others, where competing traffic is must less predictable, this is not practical.

Note that Equation 8 also describes the buffer requirement at each intermediate node of a control segment, not just the virtual destination. By the same analysis, it can be seen that if the downstream link leading from an intermediate node *I* suddenly becomes unavailable due to, for example, a high priority transmission, this node needs to be able to absorb at least $R * rtt_I + \gamma$ cells before the source can stop, where rtt_I is the round trip time for a signal between the source and node *I*. Note also that the value of *rtt* is just as important in this section as it was in Section 3.1. However, in practical rate-based networks, it is usually easier to overallocate than it is to try to measure *rtt* precisely.

3.3 Comment

The result of Section 3.2 is not as strong as we would like. Ideally, to establish a stronger duality between rate- and credit-based flow control, one should be able to show that a destination with *B* cells of buffer space can feed back rate information in such a way that the source never overflows that buffer space. If the available buffer space is reduced, then the destination should be able to send back a lower rate with confidence that the source will not only transmit more slowly but will send fewer cells as well.

This seems to depend upon *rtt*, the interval between *RM* cells γ, and the reduction function *f*. If a destination receives an *RM* cell at time t_k, and if in the interval $[t_k - rtt, t_k]$ it had previously sent back a sequence of *RM* cells with explicit rates $R_1, R_2, ..., R_{k-1}$, it can compute the maximum number of cells that it could receive from the source during the next *rtt* seconds as a result of these *RM* cells. Let this number be B_{in_flight}. Then the buffer space still available is given by

$$B_{free} = B - B_{in\_flight} - B_{occupied},$$

where B is the total buffer space allocated to the connection and $B_{occupied}$ is the buffer space occupied by cells already received but not yet forwarded or consumed.[5]

Now the trick is to determine what rate R_k to return so that the source will not send more than B_{free} cells. Note that if a source has received an explicit rate R from the destination and if it never receives another *RM* cell after that, it will send at most

$$\int_{t_k}^{\infty} f(R, \gamma) dt \tag{9}$$

cells. Thus, with an appropriate reduction function f, it is sufficient to find a value R_k so that (9) does not exceed B_{free}.[6]

While in principle, solving an integral equation provides a way for a rate-based flow control system to approximate that of a credit-based system, it is far from practical for cell-by-cell scheduling at gigabit/second speeds. Moreover, the dynamics of connections scheduled this way are far from clear. Therefore, it remains a challenge as to how strong the duality between rate-based and credit-based flow control can be.

4.0 Applications of the Duality

Despite the comments of Section 3.3, some applications of the duality are possible. In this section, we apply the results of Section 3.1 to show that credit-based flow control can support control segments with intermediate nodes and also that it is not necessary to maintain separate queues for each connection.

4.1 Control segments with Intermediate Nodes

A claimed advantage of rate-based flow control over credit-based flow control is that control segments may consist of many nodes. Much of the discussion about rate-based proposals has tended to emphasize an end-to-end philosophy in which each connection has exactly one control segment beginning at the original source and ending at the final destination. By contrast, the credit-based flow control proposals have tended to emphasize a link-by-link philosophy in which each physical link represents a separate control segment for every connection that passes over it.

---

[5] Note that this same kind of analysis was required in early versions of the credit-based proposal, in which a node had to keep track of the number of credits sent back over the previous *rtt* seconds.

[6] In the current rate-based proposal before the ATM Forum, (9) integrates to the constant *1 / RDF*.

We will use the duality to show that the same end-to-end control is possible in both rate-based and credit-based networks, with essentially the same behavior. Consider the connection *ABDE* illustrated in Figure 3, and assume that this connection has a single control segment extending from *A* to *E*. In a rate-based system, the source *A* would send an *RM* cell periodically in the forward direction. This would pass through *B* and *D*, and would be turned around by *E*, then follow the same path in the backward direction. *A* indicates the desired rate of transmission in the forward *RM* cell, then *E* sets the explicit rate field to a rate that it is willing to accept. Both *B* and *D* can reduce this rate by updating an *RM* cell in the backward direction. For example, suppose that *B* needs to divide the bandwidth of the link *B—D* "fairly" among its *n* incoming connections. It might update the explicit rate field of the backward *RM* cells of *ABDE* to a value of *1/n* of the link bandwidth. As connections come and go, it could adjust this rate accordingly.

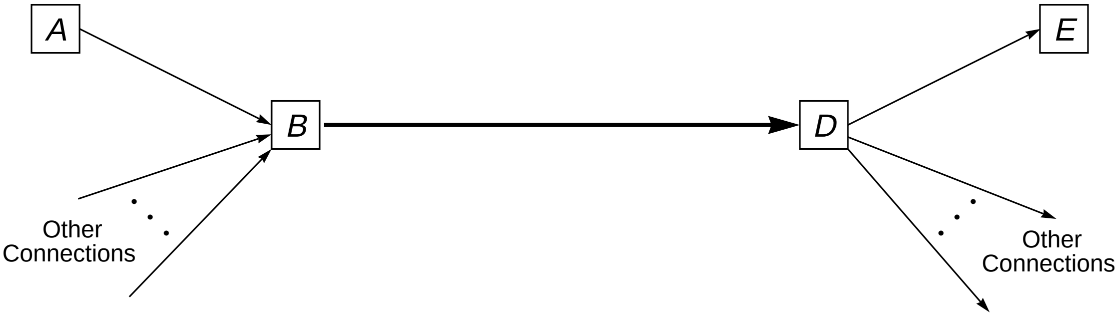

Figure 2. End-to-End Control Segment.

Note that according to Equation 8, *B* needs at least

$$R * rtt_{A-B} + \gamma$$

cells of buffer space in order to support rate *R* for the connection *ABDE*, where rtt_{A-B} is the round trip time of a signal between *A* and *B*. Without this much buffer space, a sudden burst of high priority traffic on one of the other connections sharing the link *B—D* could cause *ABDE* to overrun the available buffer space before *A* receives an *RM* cell telling it to slow down or stop.

Now let us apply the duality between rate and credit to see how a credit-based flow control would work in the same situation. For simplicity, assume the hybrid flow control method of Section 2.4. After each γ cells transmitted, *A* inserts an *RM* cell into the stream with a serial number field identifying the serial number of the next cell to be transmitted. This *RM* cell passes "in band" through *B* and *D*. That is, it is buffered, queued, and transmitted in the forward direction, just as if it were an ordinary data cell. When it arrives at *E*, the latter replaces the serial number with a credit limit indicating the highest serial numbered cell that *E* is willing to receive, then returns it to the source.

Although *B* and *D* are only intermediate nodes in the control segment, they must still maintain enough buffer space to support this and all other connections — in fact, at least as much buffer space per connection as specified by Equation 8. When an *RM* cell passes by in the backward direction, *B* or *D* adjusts the credit limit downward if necessary to reflect the amount of buffer space actually available in each. For example, suppose *B* has allocated B_{ABDE} cells of buffer to connection *ABDE*. When a credit cell arrives from downstream at time t_0, *B* must reduce the credit limit to no more than

$$n_0 + B_{ABDE} \tag{10}$$

cells, where n_0 is the number of cells that *B* has already forwarded from this connection.[7] As a result, a sudden burst of high priority traffic on some other connection sharing *B—D* may cause cells of *ABDE* to accumulate and be delayed at *B*, but (10) guarantees that there will be no buffer overflow.

4.2 Independence of Flow Control and Scheduling

Note that there is nothing in the previous section that requires *B* or *D* to be virtual sources and virtual destinations. Because *B* and *D* only reduce credits received from downstream, any cell received from the source is automatically eligible to be forwarded. Thus, there is no need for them to have the capability to dispatch cells of one connection while holding back the cells of another. In other words, this form of credit-based flow control imposes no need for separate queues for each connection. This is very different from most of the proposals for a credit-based flow control standard. It is significant because the cost of these separate queues is one of the principal arguments against credit-based flow control.

For example, suppose that *B* maintains a single queue for all arriving cells, sorted only by priority level. That is, cells of high priority connections are transmitted before cells of lower priority connections, but cells of different connections with the same priority level are commingled in the same queue and transmitted in FIFO order. In a rate-based network, *B* must keep track of comings and goings of cells of each connection so that it can adjust explicit rates, but this does not affect the actual scheduling of cells for transmission. By the duality, a credit-based *B* needs only keep track of the comings and goings of the cells of each connection so that it is aware of how much of its buffer space is occupied or committed to each connection. In fact, it appears that the complexity of the bookkeeping required by the rate- and credit-based systems is approximately the same.

The queuing and transmission of data and *RM* cells in the forward direction is a completely independent activity. Any scheduling algorithm suitable at *B* for a rate-based network is also suitable for a credit-based network. To the degree that flow control can be used to indirectly influence the scheduling by influencing rates of transmission at the sources, it can be applied equally in both kinds of networks. For example, suppose that *B* needs to divide the bandwidth of the link *B—D* “fairly” among its *n* incoming connections. In a rate-based system, it would calculate an aggregate bandwidth for all ABR connections on that link. Then, it would apportion this bandwidth equitably among the connections by setting explicit rates in their *RM* cells. Connections transmitting faster than their fair share would be forced to slow down, while slower connections would be allowed

---

[7] The only constraint is that *B* must be careful not to starve the source by reducing the credit limit so much that no future *RM* cells can get through to stimulate the transmission of more credits.

to speed up. In a credit-based system, *B* would apportion the total buffer space, instead. Faster connections would be slowed down by reduced buffer space, while slower connections could transmit faster.

4.3 Long Control Segments

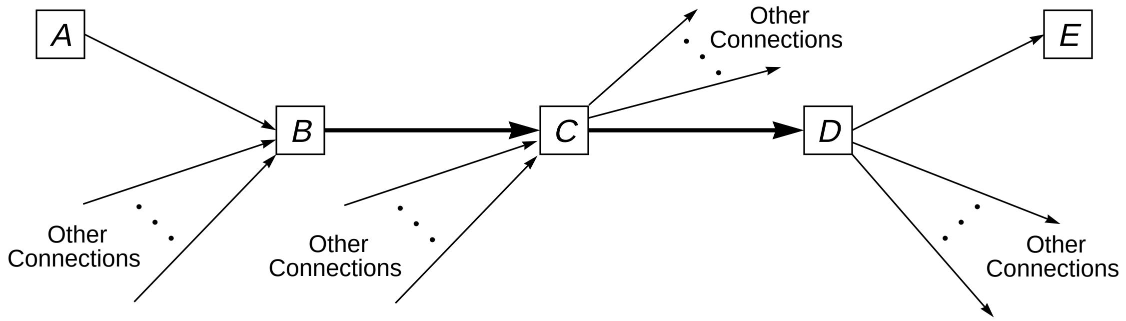

Figure 3. Multi-hop connection.

Note that rate- and credit-based networks both suffer an identical limitation of multi-hop control segments such as the one illustrated in Figure 3. Suppose that node *B* wishes to allocate link *B—C* “fairly” among *n* ABR connections, and suppose that node *C* wishes to allocate link *C—D* “fairly” among *m* ABR connections. In a rate-based network, *C* reduces the explicit rate of backward *RM* cells of connection *ABCDE* to a maximum of

$$1/m * R_{CD}$$

and *B* reduces this field further to a maximum of

$$1/n * R_{BC}.$$

Both nodes do the same for all other connections on their respective links. If

$$1/n * R_{BC} < 1/m * R_{CD}, \tag{11}$$

then the bandwidth of link *C—D* is underutilized. This is because connection ABCDE cannot keep up with the rate allocated to it by *C*, and the others were already slowed down by *C* so they cannot take advantage of the surplus.

The same happens in a credit-based network. *B* partitions its available buffer space so that *ABCDE* gets *1/n*-th of it, and *C* partitions its available buffer space by *1/m*-th. If buffer space at *C* is limited, then the other connections on *C—D* have their rates restricted, while *ABCDE* cannot take advantage of either its buffer space or the share of bandwidth that this implies.

Note that it does not help to modify the rate-based flow control algorithm to allow intermediate nodes to reduce the desired rate field in a forward *RM* cells so that downstream nodes could know that a connection could not use its full “share.” If this were the case and if inequality (11) were reversed in our example, then link *B—C* would be underutilized, instead. Similarly, in credit-based networks, it does not help to overallocate buffer space, thereby transferring the underutilization from an expensive resource (long distance link bandwidth) to a cheap resource (buffer memory). While this actually lets the connections find their own level, it also reduces the “fairness” of the bandwidth allocation, thereby defeating the original objective.

In general, the solution to this problem requires some sort of feedback loop among the nodes involved. There are many possible approaches, all involving additional complexity and all beyond the scope of this paper. For example, using intelligent marking in *RM* cells [Siu95], rates will, in favorable circumstances, converge to equitable levels. However, this is a complex control problem that can have many instabilities in different situations. Another approach is to partition the network into separate control loops and to implement the “fairness” in the scheduling of connections from their own queues. This is the implicit approach in the credit-based proposals of Kung and others. In actual practice, a lot of experimentation and operational experience is needed before a suitable solution emerges.

5.0 Limitations of the Duality

If the duality were perfect, then every behavior in one form of flow control would have an identically behaving counterpart in the other. However, this is not the case. In their present form, the rate-based and credit-based methods of flow control do produce different behavior in some circumstances. In this section, we look at three of these.

5.1 Very Low Data Rates

It was observed in Footnote 4 that it is not practical to use credit-based flow control to achieve very low data rates. For any value of γ, the lowest average rate that the network can promise for a connection is γ / rtt. Even with $\gamma = 1$, this means that the slowest connection can go as fast as $1 / rtt$ in the absence of other limiting factors.

By contrast, explicit rates can be made extremely small and are limited only by the resolution of the rate fields in *RM* cells. This means, for example, that it is possible to allocate the bandwidth of a very short, fast link “fairly” among very many connections in a rate-based network in the style of Figure 2 but not in a credit-based network. I.e., the duality breaks down because of the difference in resolution of the two control methods.

5.2 Self-limiting Connections

A credit-based network has the advantage of being “self-limiting.” That is, in the absence of positive feedback, a source is guaranteed to stop transmitting before it overruns the buffers of the destination. In rate-based networks, a source slows down gradually in the hope that there really are enough buffers to absorb all of the cells that it can transmit before stopping.

In some cases, however, it is possible to set the parameters of a rate-based connection to more closely approximate the behavior of a credit-based network. Let

- the parameter *RDF* in Equation 1 be equal to *1*,
- *MCR*, the minimum cell rate, be equal to *0*, and
- *AIR*, the additive increase parameter, be equal to *PCR*, the peak cell rate of the connection.

Then when the source receives an *RM* cell from downstream, it immediately increases its rate to the smaller of *PCR* and the explicit rate in the *RM* cell. If γ is the interval (in cells) between consecutive *RM* cells, the source will transmit γ data cells and stop. Later, when another *RM* cell arrives, it will start up again and repeat the process. By this means, we can achieve an average data rate of γ / rtt cells per second while at the same time provide a self-limiting behavior similar to that of a credit-based network.

The problem is that this does not deliver very much bandwidth to the connection for large values of *rtt* — i.e., where γ / rtt is much smaller than *PCR*. It also does not help to add more buffer space at the destination, as it would in a credit-based network. Thus, again the duality breaks down.

5.3 Very Small Buffers and Open Loop Control

The analysis of Section 3.2 shows that, in general, it is necessary to allocate

$$B(t_0) = R(t_0) * rtt + \gamma$$

cells of buffer for each connection in a network with rate-based flow control in order to avoid cell loss due to transients in the rate of forwarding or consuming cells. Thus the aggregate buffer requirement for a physical link is of the order of

$$R_{link} * rtt$$

cells. This can be very large; for example, on transcontinental links with *rtt* values of the order of 50 milliseconds and link bandwidths of 2.4 gigabits/second, the number of buffers would be nearly 300,000 cells. This is based on the assumption that the flow control must accommodate the most extreme transients.

An alternative is to avoid the transients themselves. For example, Figure 4 illustrates a number of ABR connections originating in the same local area network and competing for the bandwidth of a long distance link. These ABR connections would be flow controlled across relatively short control segments A_1–*B*, ... A_5–*B*. Thus the buffer space required at *B* would be relatively small. The control segment from *B* to *C* is, by contrast, very long. However, if the switch at *C* can be assured that there are no impediments to the flow of any connection *away* from *C*, then it does not have to maintain large buffers to protect itself from transients. I.e., if the control segments *C*–D_1, ..., *C*–D_5 can each guarantee that the data of their connections will move away at some minimum cell rate, then there is no danger that the buffers of *C* will overflow. In effect, the link *B*—*C* can operate almost in open loop mode, without much flow control at all, and *C* can operate with at most a few hundred cells of buffer.

Figure 4. Local connections to Long Distance Link

This is typical of a telecommunications network. Many ABR connections may compete for the same bandwidth resource and would be subject to tight flow control over the local loop. But the destinations are required to consume cells as fast as the network will deliver them (or suffer the consequences). In this case, a rate-based method is appropriate, not for the cell-by-cell flow control in a high transient situation, but for long term, gradual adjustments of rates at granularities of seconds, minutes, or longer.

There does not appear to be a comparable way for a credit-based system as described in Section 2.2 to provide a similar kind of control for the link *B—C*. As a consequence of Equation 4, a *necessary* condition for maintaining the link data rate in a credit-based network is a minimum buffer space of the order of

$$B \times rtt + n \times \gamma / (PCR)\ .$$

This is an enormous number of cells, and there does not seem to be any kind of open-loop control method in the credit-based network that gets around the requirement. Thus, the duality does not apply to the open-loop situation.[8]

6.0 Comments and Conclusions

The assumptions leading to the duality hypothesis of this paper are predicated on the need to address extreme transients in the available bit rate. This is exactly the problem that credit-based flow control is meant to solve. A reasonable question is, however, how realistic are such transients and whether or not they can be ignored in practical networks.

Simulations have shown that with appropriate parameters, rate-based flow control in a local area network with very small *rtt* values can adjust to transients in a few tens of milliseconds [Barnhart94a, Siu95]. This seems to be enough for typical office networks in which the distributed computing is user initiated. Indeed, the time and overhead necessary

---

[8] The response by advocates of credit-based networks to this situation is to “tunnel” credits through the long distance link. That is, a credit limit will be transmitted from some node near D_i back to some node near A_i, independent of any flow control on the link *B—D*. It becomes the responsibility of the downstream node to deal with its own transients, *not* the network’s responsibility.

to set up a new connection for, say, file transfer, is likely to be in the hundreds of milliseconds — i.e., far longer than the time needed for the rest of the ABR connections to adjust their rates to a "fair" level. More generally, although much of the ABR traffic is believed to be of a self-similar nature [Leland94] so that bursts in traffic cannot be smoothed by statistical multiplexing, nevertheless a credible argument can be made that a rate-based network could respond to bursts faster than they occur in real networks.

However, the reason we are interested in ATM networks is to support a broader class of applications in one network. In our environment, extreme transients are the rule rather than the exception. In an example application, typical of the control of a power plant or waste water treatment plant, a distributed, reflective shared memory is updated among all of the real-time computers of the plant at intervals of one-half millisecond. This means bursts of several hundred high priority data cells, each burst of which consumes a substantial part of the link bandwidth for its duration. Longer bursts of medium priority data cells occur at intervals of a few milliseconds. These demand response times from ABR traffic that are much faster than those needed in the typical office environment and much closer to the assumptions of Section 3.2.

We have shown that, under these assumptions, there appears to be a kind of duality between rate-based and credit-based flow control for ABR traffic. This duality is not perfect, but within limits it is possible to map the behavior of a network using one kind of flow control to similar behavior in a network using the other. We have used the duality to show that credit-based flow control does not have to be applied on a link-by-link basis with separate queues for each connection. We have also showed that the complexity of allocating bandwidth fairly in multi-hop control segments is essentially the same in rate-based or credit-based networks. The duality breaks down when the parameters approach the limits of resolution of the control, but in opposite directions. Credit-based flow control is not particularly good at restricting the rates of connections to very small values, and rate-based flow control is not particularly good at responding to transients at high peak rates. In addition, rate-based flow control can be used open-loop situations (or situations with very, very long time constants) with very small buffers, whereas credit-based flow control requires far more.

The purpose of this discussion is to stimulate thinking in a different way from that prevailed in the rate *versus* credit debates of the ATM Forum. We have not captured full complexity or implications of either approach, but this should serve as a starting point for further analysis.

7.0 References

[Barnhart94a]

A. W. Barnhart, "Explicit Rate Performance Evaluations," *ATM Forum Contribution 94-0938R1*, October 1994.

[Barnhart94b]
A. W. Barnhart, “Proposal for Example Pseudocode for the End-System,” *ATM Forum Contribution 94-1110*, November 1994.

[Ferrari90]
D. Ferrari and D. Verma, “Real-Time Communication in a Packet-Switching Network,” *Proc. Second International Workshop on Protocols for High-Speed Networks*, Palo Alto, November 1990.

[Hunt94]
D. Hunt, W. Andrews, J. Scott, R. Simcoe, J. Bennett, H. T. Kung, J. Howard, A. Chapman, and K. Brinkerhoff, “Credit-based FCVC Proposal for ATM Traffic Management, Revision R2,” *ATM Forum Contribution 94-0632R2*, July 1994.

[Kung94]
H. T. Kung, T. Blackwell, and A. Chapman, “Credit UPdate Protocol for Flow-Controlled ATM Networks: Statistical Multiplexing and Adaptive Credit Allocation,” *Proc. ACM SIGComm ‘94 Conference*, University College, London, August, 1994.

[Leland94]
W. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson, “On the Self-Similar Nature of Ethernet Traffic,” *IEEE/ACM Transactions on Networking*, volume 2, number 1, February, 1994.

[Roberts94a]
L. Roberts, B. Makrucki, T. Tofigh, A. W. Barnhart, B. Holden, G. Fedorkow, J. Daigle, M. Hluchyj, H. Suzuki, G. Ramamurthy, P. Newman, N. Giroux, R. Kositpaiboon, S. Sathe, G. Garg, and N. Yin, “Closed-Loop Rate-based Traffic Management,” *ATM Forum Contribution 94-0438R1*, July 1994.

[Roberts94b]
L. Roberts, “Enhanced PRCA (Proportional Rate-Control Algorithm),” *ATM Forum Contribution 94-0735R1*, September 1994.

[Siu95]
K.-Y. Siu and H.-Y. Tzeng, “Intelligent Congestion Control for ABR Service in ATM Networks,” Technical Report, University of California at Irvine, January 1995; submitted for publication.

[Zhang93]
H. Zhang and D. Ferrari, “Providing Deterministic Guarantees For Bursty Traffic,” *Workshop on the Role of Real-Time in Multimedia/Interactive Computing Systems, IEEE Real-Time Systems Symposium ’93*, Raleigh-Durham NC, Nov. 1993.

[Zheng94]
Q. Zheng, “Credit-Enhanced Rate Control: An Example Integration Scheme,” *ATM Forum Contribution 94-0942*, September 1994.