Virtual-Credit: Efficient End-to-End Credit Based Flow Control

Yehuda Afek* Yishay Mansour† Zvi Ostfeld‡
Computer Science Department,
Tel-Aviv University, Israel 69978.

Abstract

A new flow control scheme, called *virtual credit* that may be considered either as a credit based flow-control algorithm without a queue per session at the ports, or as a way to modify the credit based scheme and to bring it closer to an end-to-end rate based scheme, is presented. The new scheme brings the credit based and rate based flow-control schemes closer to each other, giving an attractive compromise between the two schemes. The new scheme enjoys the main advantages of the credit based flow-control scheme while not requiring a queue per session at the switches. A possible disadvantage of the scheme is the large buffer requirement at the switches to ensure very low cell loss rate (but no more than the corresponding buffer requirements in rate based schemes).

Key words: Flow-control, Credit-based, Rate-based, ABR, Round robin.

1 Introduction

Flow control is a distributed network protocol whose goal is is to prevent and resolve data traffic congestion while ensuring high utilization and fairness among the different connections. This task is achieved by controlling the flow of traffic that enters the network, and the flow of traffic inside the network. In high speed data networks, the entire network may be clogged in a split of a second, because of the high transmission rates and the large amounts of data that may be on the lines at any point in time. Thus making the task of flow control more challenging and crucial.

Many different flow control mechanisms have been proposed over the years (see [GK80, Tan81, BG87, Jai95, Ram95] for references and survey). In recent years two leading candidates were considered by the ATM Forum, the credit based scheme [KM95], and the rate based scheme [BF95] (see [Jai95] for a survey about the different Congestion Schemes). The rate based scheme

*Telephone: +972-3-6408952,Fax: +972-3-6409357, e-mail: afek@math.tau.ac.il
†Telephone: +972-3-6408829,Fax: +972-3-6409357, e-mail: mansour@math.tau.ac.il
‡Telephone: +972-3-6407439,Fax: +972-3-6409357, e-mail: ostfeld@math.tau.ac.il

is an *end-to-end* scheme in which the sources of virtual circuits (VC) control the flow of traffic in the session. Periodically, the source sends a cell that makes a round trip to the destination and back collecting on its way the minimum rate over all the switches along the VC, at which they can support that VC. The source uses this information to adjust the rate of the VC.

Unlike rate based, credit based is a link by link flow control mechanism. That is, it uses a separate window flow control mechanism for each session on each link. The downstream switch of every link manages a buffer for each session (VC) passing through the link. Whenever a cell is sent downstream out of the buffer, a *credit* is sent upstream to inform the previous, upstream switch that it may send forward one more cell of this session. Every switch transmits cells out of the sessions buffers, that have both data to send and credits, in a round robin manner. The usage of round robin ensures max-min fair division of the network bandwidth among the different sessions [BG87, DKS89]. This basic scheme, called *static buffer allocation* has been extended optimized and tuned in [KBC94, OSV95, HNS+94, KM95] in order to overcome potential problems such as credit loss and to reduce the buffer requirements.

In the credit based scheme, congestion on one link blocks the traffic of sessions going over that link (by not giving anymore credits) and thus causing *back pressure* that stops the flow of traffic in these sessions along their VCs from the congested link to the sources. Had we used one queue to store the cells of all sessions outgoing on a port we would run into the following, "blocking head of the queue" problem: the cell at the head of the queue belongs to a blocked session (a session with zero credits) while subsequent cells in the queue belong to un-blocked sessions that could have been forwarded along the link. In more severe scenarios [Ram95] this phenomena may lead to a deadlock. Thus, as stated above, this scheme requires a queue per session at each outgoing port in the network.

The main advantage of the credit based scheme is very low cell loss rate while its main disadvantage is the implementation complexity, namely the necessity of a separate queue for each VC in each link. The main advantage of the rate based scheme is its simple switch architecture, while its main disadvantage is the higher cell loss rate (if the same amount of buffer space is assumed in both schemes). The implementation complexity of credit based flow control was one of the main reasons why the ATM Forum has prefered the rate based scheme [Jai95].

In this paper we suggest a new extension to the credit based scheme called *virtual credit* that does not require a queue per VC in each link, and yet is immune to head of the queue blocking and is deadlock-free. The crux of the new extension is to "simulate" the credit based scheme on a copy of the network, called *virtual network*, with the same traffic load as the real network and to give VC's sources in the real network the same credits as the sources in the virtual network get. However, in both the real and the virtual network the flow-control is implemented without a queue per session. In the virtual network we use two counters for each session while in the real network one queue for all the data traffic is used. Since sources send the same number of cells as they do in the mimic copy, the allocation to the sources is as in the mimicked credit based protocol. Thus, max-min fairness allocation is obtained (since the credit based scheme achieves this). The mimicking of the credit based flow control scheme utilizes only a small amount of bandwidth, and therefore does not have a significant impact on the performance.

The implementation of the virtual credit scheme is simple, scalable and robust. The virtual credit scheme has most of the advantages of the credit based method, such as simple source implementation, fast reaction to changes in the network and simple possibility for policing. It can also be smoothly integrated in current networks that implement a credit based scheme.

Though the virtual credit scheme uses a mimic of the credit based scheme, it is an end-to-end scheme, where the control on the rate of a session is done at the source and is not dependent on the scheduling of the switches.

A major difference between the virtual credit scheme and the "classic" credit-based scheme is related to the amount of buffer space. In order to avoid data loss in extreme transient states buffer space should be larger in the virtual credit scheme than in the credit based scheme. This is due to the end-to-end nature of the scheme, which may result in a large queue build-up in the bottleneck links as oppose to the classic credit-based scheme in which the congestion results in a build-up of several smaller queues in different links of the same total size as our one larger queue. Following the presentation of the basic scheme we propose some extensions to reduce the size of the required queues in the virtual credit scheme. However, as shown by our simulations, we believe that in most scenarios the queue length attains reasonable values. Moreover, it is not clear that the large queue build-up near a congested link is worse than the build up of several queues along the VCs paths (why should a congestion in one part of the network consume valuable buffer space all along the virtual circuits paths). Furthermore, the maximum length of the queue using the new scheme is likely to be less than the maximum queue if the rate based scheme is implemented and required to attain the same cell loss rate.

A few proposals to combine rate based and credit based flow control have been suggested. In [TS94] it is proposed to implement the credit based flow control with one queue for all the sessions in each port by using a window for every link (in order to avoid congestion) and a Congestion Indication bit per session (in order to achieve fairness). The solution in [TS94] is different than our solution, and includes a control at the link level. This implies that special care is necessary to avoid deadlocks and head of the queue blocking. Another suggestion in [RN95] is to use the rate scheme in WANs and to use the credit scheme in LANs. As one can see these schemes are quite different than the virtual credit scheme presented in this paper both in their goals and techniques.

In Section 2 we present the virtual credit scheme and in Section 3 some performance analysis is presented. In Section 4 several methods to reduce the maximum buffer size required by the virtual-credit scheme are suggested. In Section 5 extensions and advantages of the virtual credit scheme are discussed.

2 The virtual credit flow-control algorithm

The idea of the virtual credit scheme is as follows: The network notifies the source end stations upon their allowed transmission rate. This is an end-to-end mechanism, as there is no restriction upon the scheduling in the network nodes. We suggest a mechanism according to which the feedback received at the end stations is identical to that received in the credit-based scheme. Hence, the source end-stations behave exactly as in the credit-based scheme. The suggested mechanism uses a virtual network in which it mimics the credit-based scheme.

Hence, a mirror copy of the network is constructed, called *virtual network*, and an arbitrary correct credit (good and sophisticated) flow control algorithm [KBC94, OSV95, HNS+94, KM95] is run on the virtual network with the same traffic requirements as in the real network. Whenever a source gets a permit to send a cell in the virtual network we allow the source to send a cell in the real network. As explained below this attitude implements an end to end flow control mechanism in the real network. Hence, one can use simple scheduling methods in the network nodes such as FIFO and still obtains fairness.

To be more concrete, the virtual network implements a credit-based flow control algorithm. We use the scheduling of the source in the virtual network in order to control the flow in the real network. Note that there is no queue per session mechanism in the real network (only a single queue per link in each switch). As we show in the sequel, it is possible to simulate the credit-based flow control algorithm without consuming too many network resources, and without requiring a queue per session, thus keeping the overhead low.

Since the full fledged classical credit based algorithm achieves max-min fairness, and the virtual credit algorithm mimics it, then it also achieves max-min fairness (and most of the other advantages of the credit-based scheme). In the rest of this section we describe the virtual credit algorithm in details. In Section 3 we discuss the buffer size that is necessary for our algorithm (note that our algorithm is an end-to-end algorithm, and thus will require more buffer space than the hop by hop control of the credit based flow control).

We suggest three methods for implementing virtual credit.

1. A division of the network bandwitch to two halfs. On one half the virtual credit method is implemented using dummy cells (i.e., cells with control information but without data) while on the other half the real network is implemented.

2. Combining some control cells from the virtual network into one cell, thus reducing the amount of bandwith allocated to the virtual network.

3. Implementing the real network using the entire bandwidth while using a piggyback mechanism in order to transfer messages of the virtual network.

Subsection 2.1 presents the first method while Subsection 2.2 presents the second method.

2.1 Basic Implementation of the Virtual Network

We need to simulate the virtual network on top of the real network (see Fig. 1 and Fig. 2). The content of the cells in the virtual network have no data, and therefore we refer to them as dummy cells. There are two kinds of such cells, *dummy data cells* and *dummy credit cells*. In contrast the cells in the real network are called *data cells*.

Essentially we run two flow control algorithms in the network. The two are completely independent excpet at the source of traffic. The two algorithms are, *virtual credit*, and *simple*. At each intermediate switch in *simple* there is simply one queue per outgoing link that serves cells in first-in-first-out order. At the source node of *simple* a data cell is sent on an outgoing link only if a dummy cell may be sent in the virtual-credit algorithm. Thus, at the source (and only at the source) the two algorithms are tied together and transmit in full synchrony when both there is a credit available at the virtual-credit algorithm and there is a real data cell ready to be sent.

The effect of the above mechanism, as decribed, is that the simulated credit based algorithm (called *virtual credit*) controls the rate at which the source may transmit. Since the virtual credit algorithm "sees" the same bandwidth as the real traffic, it will enable the same rate as a real credit based algorithm would have allowed.

An underlying principle of our simulation is that on each link the bandwidth given to the real traffic is the same as that given to the virtual traffic. That is, every second opportunity to send a cell (called cell slot) is given to the virtual network and the other one to the real network. If either has nothing to send in that cell slot, that cell slot is left vacant. In this subsection we describe the implementation using the above simple rule, and in the next subsection we discuss how to improve the performance, by decreasing the ratio between dummy cells and data cells.

Since the cells in the virtual network are dummy, i.e., do not contain any data, instead of having a queue per session (in each intermediate switch) we hold a counter per session that counts the queue length in the simulated credit-based algorithm. In addition to the queue counter, there is of course the credit counter as in the credit-based algorithm [KM95]. Thus, in the implementation, one dummy data cell is sent for each forward data cell sent. However, while the forward data cell is sent out of the (single) queue of data cells, the dummy cell is "sent" using the corresponding counter, according to the Round-Robin order between sessions.

The basic implementation of the combined mechanism in the source is shown in Fig. 1. The variable *Output* maintains the number of forward data cells that are waiting to be sent from the source, while the variable *Credits* maintains the number of credits received in the virtual network. The variable *Credits* is incremented with each dummy credit cell that is received in the virtual network, and is decremented with each transmission of a dummy data cell in the virtual network (which is done together with a transmission of a real data cell in the real network). Note that this is the only place where the virtual and real networks interact. Since we do not allow *Credits* to become negative, the credits received in the virtual network control the rate at which the source can send data in the real network.

The basic implementation of the simulation in the switch is shown in Fig. 2. Half of the available cell slots are allocated to the virtual network while the other half is allocated to the real network. The virtual network does a mimick of the credit based mechanism while in the real network the real data cells are sent using a simple FIFO mechanism on each link (regardless of the session they belong to).

The use of the classical static buffer allocation in the example is for simplicity. The more sophisticated adaptive buffer allocation may be mimicked too with no additional overhead (see the works of [KBC94, OSV95, HNS+94, KM95]).

2.2 Reducing the overhead

The bandwidth overhead of the virtual credit as described in the previous subsection is half of the ABR network bandwidth, which is clearly unacceptable. To reduce the overhead of the virtual network while still allowing a good mimic we suggest to combine every $k-1$ dummy data cells to one cell and using its data content to describe the header contents of the $k-1$ cells. Specifically, every $k-1$ available cell slots (i.e., opportunities to send ABR cells in the actual network) a special cell is inserted. The special cell combines in it the $k-1$ dummy data cells that would have been sent, one for each ABR slot. Note that a slot in which the special cell can be injected is allocated every $k-1$ ABR slots regardless of the number of these slots that carried real data or that were left vacant. ($k=2$ is the basic scheme described above.) In this way, the bandwidth consumed by the mimicking process is $\frac{1}{k}$ of the network bandwidth and the amount of bandwidth available for real data is $\frac{k-1}{k}$ of the network bandwidth. In this scheme the additional requirement for buffer space (in order to prevent under-utilization) when

Variables:

parity - Indicates the parity of the time slot
Output - Number of data cells that are waiting to be sent
Credits - Number of cells that are allowed to be sent
(in both networks)

Initialization:

$parity := odd$
$Output := 0$
$Credits :=$ Initial allocation used in Credit-Based mechanism

When a new data cell arrives
$Output := Output + 1$

Every available cell slot
If ($parity = odd$) **Then**
If ($Output > 0$ and $Credits > 0$) **Then**
Send a forward data cell
$parity := even$
Else { $parity = even$ }
If ($Output > 0$ and $Credits > 0$) **Then**
Send a dummy data cell
$Output := Output - 1$
$Credits := Credits - 1$
$parity := odd$

When a new dummy credit cell is received:
$Credits := Credits + 1$

Figure 1:
Virtual Credit: Basic Pseudo Code for the Source. Half of the banwidth of the link (emenating from the source) is allocated to the virtual network while the other half is allocated to the real network.

Variables:

parity - Indicates the parity of the time slot

Output and *Credits* are two arrays where

$Output[i]$ - Number of dummy cells that are waiting to be sent by session i

$Credits[i]$ - Number of dummy cells that are allowed to be sent by session i

Initialization:

```
For each session i
        parity := odd
        Output[i] := 0
        Credits[i] := Initial allocation used in Credit-Based mechanism

Every available cell slot
    If (parity = odd) Then              { This cell slot is allocated for the virtual network }
        parity := even
        Do a Round-Robin mechanism over all sessions j such that
            Credits[j] > 0 and Output[j] > 0
        If session i is chosen by the Round-Robin mechanism Then
            Output[i] := Output[i] - 1
            Credits[i] := Credits[i] - 1
            Send a dummy data cell of session i
            Send a dummy credit cell of session i
    Else { parity = even }              { This cell slot is allocated for the real network }
        Send a real data cell (if there is such one)
        parity := odd

When a dummy data cell of session i is received
    Output[i] := Output[i] + 1

When a dummy credit cell of session i is received
    Credits[i] := Credits[i] + 1
```

Figure 2:
Virtual Credit: Pseudo Code for an Intermediate Switch. Half of the banwidth of the link is allocated to the virtual network while the other half is allocated to the real network.

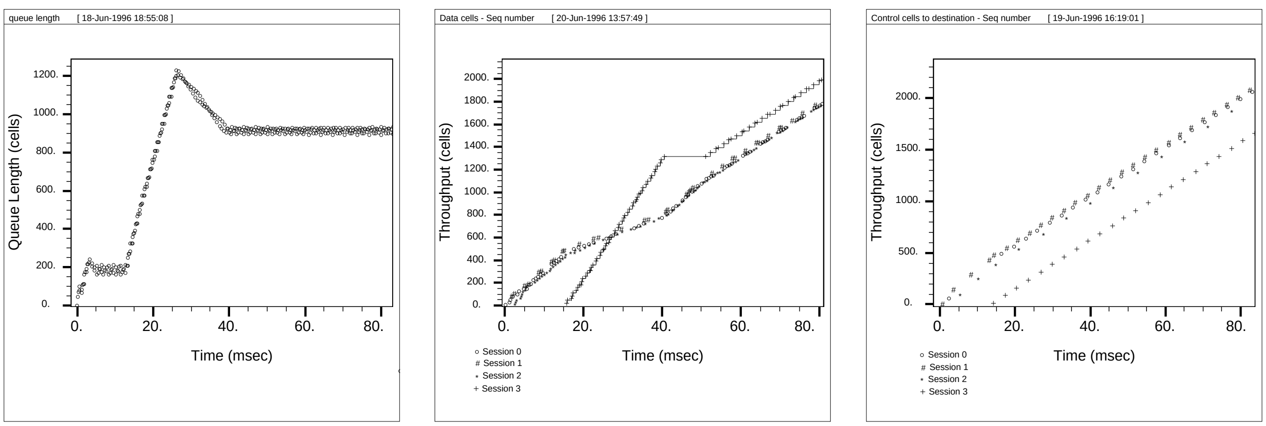
the adaptive sharing scheme is adopted is $k-1$. [1] For example, if k is set to 32 it is the same as the bandwidth overhead introduced by the forward RM cells in the rate-based scheme ([BF95]).

To show that the grouping of $k-1$ dummy data cells into one cell does not affect the fairness of the algorithm we consider a slight modification of the credit-based scheme. In this modification, the length of a basic time interval is defined to be the time slot required to transmit $k-1$ ABR cells. At the end of each such basic time interval we do the following:

1. An amount of up to $k-1$ cells is collected by running the Round-Robin procedure $k-1$ steps forward without any real transmission.

2. The collected cells are sent in the basic time interval (i.e., during the next $k-1$ available cell slots).

The modified credit-based scheme is fair by the same arguments that show that the original credit-based scheme is fair (e.g., [DKS89]). Subsequently, the fairness of the virtual credit scheme follows since it exactly simulates the modified credit-based scheme.



Queue length | Sessions' aggregate throughput | Aggregate Throughput in the virtual network

Figure 3:
Virtual-Credit - Four greedy sessions with different RTT (0.03 ms, 0.3 ms, 3 ms and 30 ms) sharing a 155 Mbs link. Except for the initial burst (when new sessions join the pool) the link throughput is fairly divided among the sessions (the rate of a session is simply the derivative of the sessions' aggregate throughput). Note that after about 50 ms, when the network becomes stable, the rates are equal.

3 Performance Analysis

3.1 Simulations Configuration

We present results of a representative set of simulations that we have carried out. The simulations are done in BONeS [ALT94]. In all the simulations the virtual network uses to combine

[1](The arguments for the additional $k-1$ cells buffer space are identical to these presented in [KM95], when referring to the use of $N2$.

the dummy cells (both the dummy data cells and the dummy credit cells) into groups of 31 (i.e., every 32 cell time, a combined dummy cell is sent). Though the virtual network is supposed to implement the re-synchronization mechanism [KM95, KBC94, OSV95, HNS+94] we omit this part from our simulations. We mimic the static buffer allocation (unless stated otherwise) in which the virtual buffer size allocated (i.e., the amount of initial credits in the upstream switch) for each session in each link (in the virtual network) is equal to 2 * (Link Bandwidth * Link Delay + 31).

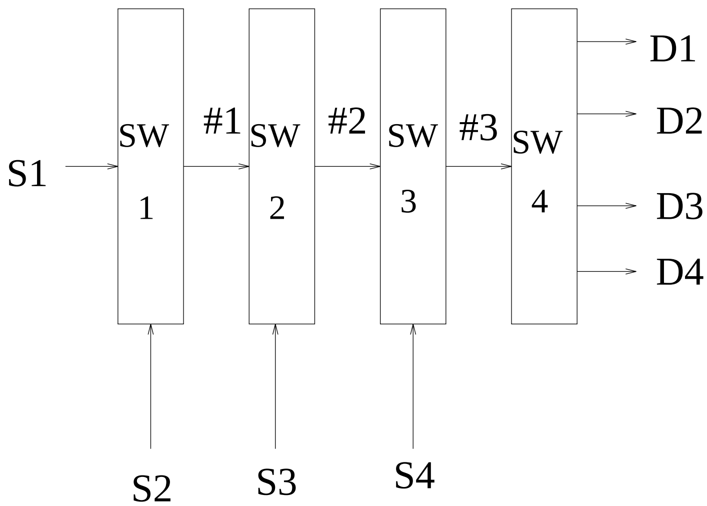
3.2 Simulations Results

In Fig. 3 four sessions with different RTT are sharing one link using the virtual credit scheme. Looking at the throughput of the sessions, one can observe that after some time (approx. 50ms in the simulation) the link bandwidth is fairly shared between the different sessions (the bandwidth of a session is simply the derivative of the sessions' aggregate throughput). The queue length is the same as the queue length obtained when using the standard credit based scheme with static buffer allocation. It is obvious that in this simple configuration, an implementation of an adaptive buffer allocation in the virtual network achieves fairness yet resulting in a smaller queue length.

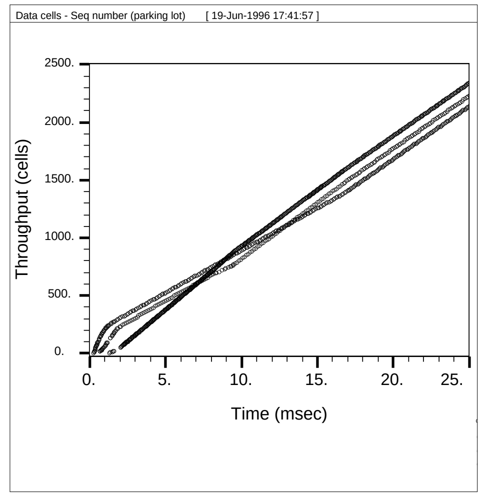
Fig. 4 illustrates the behavior of the algorithm in the parking lot configuration. The main observation here is that the virtual credit scheme achieves fairness, i.e., the throughput of the different sessions is similar. This is also a good example in which virtual credit differs from the classical credit based algorithms. The difference is in the behavior of the queue length here versus a standard credit based scheme. Where in the standard scheme the back pressure concept results in moderate queues for each session across all the links on its path, the new scheme results in a large queue on the bottleneck link (link #3), and almost no queues on the other links. As expected, the length of the large queue in the virtual credit scheme equals to the sum of the moderate queues in the classical credit-based scheme.

Consider a specific session and observe the amount of buffer space required for this session in a specifc switch when using the virtual credit. One can see that this amount is at most the accumulation of the amount of buffer space required for this session in all upstream switches on its path when using the standard credit-based scheme. More formally, assume that in a standard credit-based scheme Session i crosses link #1, link #2, link #3, etc. and that in each link $\#j$ the length of the queue required by Session i in a stable mode is $Q_{i,j}$. If link $\#k$ is a bottleneck link for Session i, then it is expected that the queue size required by Session i in the virtual credit scheme would be about $\sum_{j=1}^{k} Q_{i,j}$.

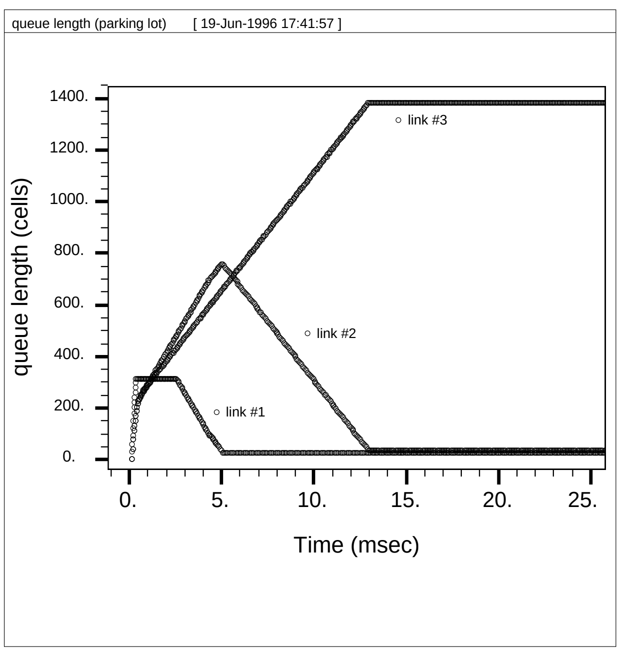
Fig. 5 illustrates the behavior of virtual credit in the presence of a greedy session and an on/off session that share a path of two links. As can be seen Fairness is achieved in that case too. The length of the queue in a link that is not a bottleneck is small. The queue length of the bottleneck link is as expected, i.e., about the sum of the lengths of the queues created by the standard credit-based scheme, for the two sessions that share it by the two links that they cross.



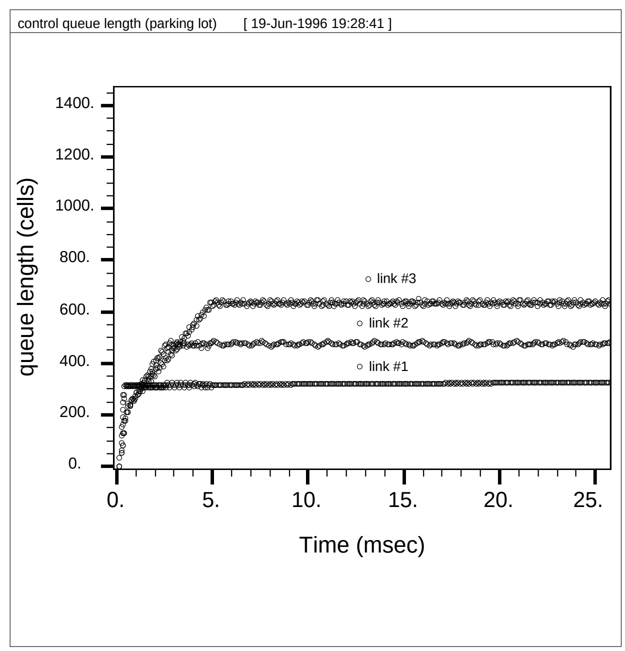
Parking lot configuration



Sessions' aggregate throughput

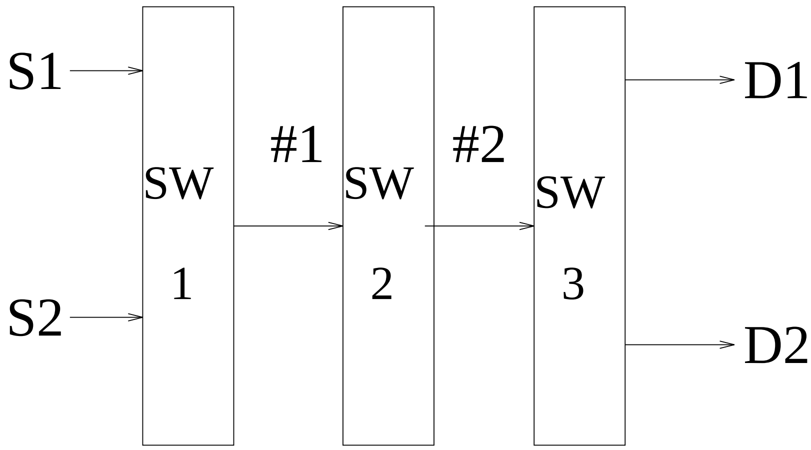


Queue length in
Virtual-Credit

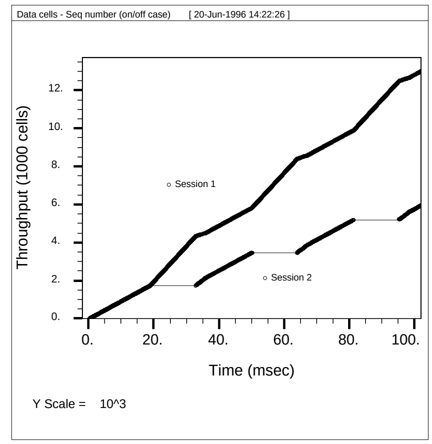


Queue length in the classical
credit-based scheme

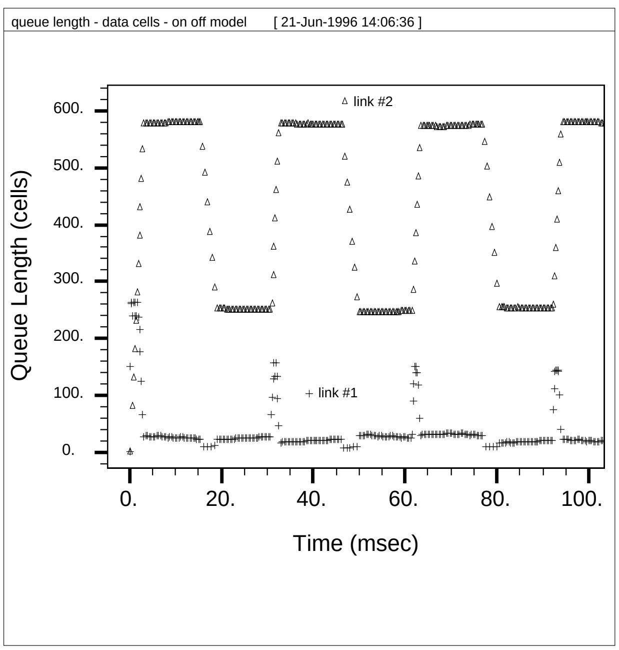
Figure 4:
Virtual Credit - Parking lot configuration. Each link has a bandwidth of 155 Mbs and a delay of 0.1 ms. The link throughput is fairly divided among the sessions (the rate of each session is the amount of cells that attains the destination during a period of time). The queues of link #1 and link #2 are drained after a short period of time while the queue of link #3 stabilizes on a large value. When using the credit-based scheme, the "queue length" (i.e., the total buffer space allocated to all the sessions) is moderate in all the three links.



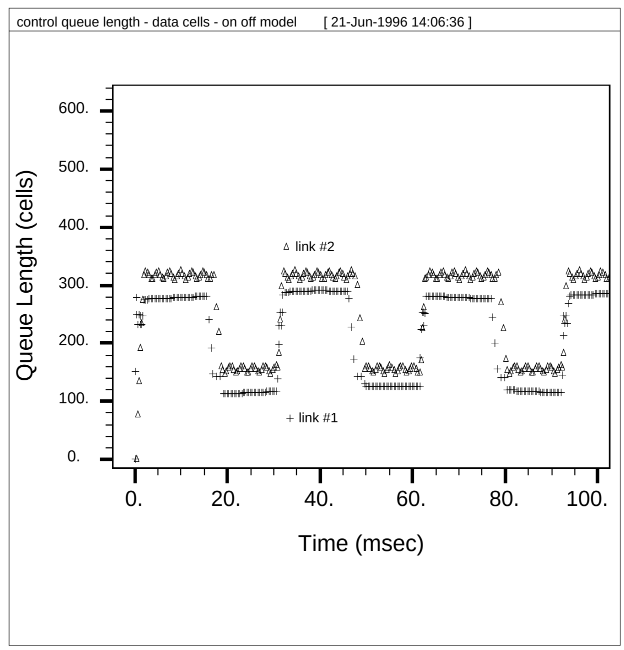
The Two links Configuration



Sessions' aggregate throughput



Queue length in Virtual-Credit



Buffer space consumed in the classical credit-based scheme

Figure 5:
Virtual-Credit - two links configuration. The first link has a bandwidth of 155 Mbs and the second link has a bandwidth of 78 Mbs. The delays of both links is 0.1 ms. One session (Session 1) is greedy during all this period of time while the other session (Session 2) is on and off for periods of time. The throughput is fairly divided among the sessions (while Session 2 is on, one can see that the derivate of the graphs are equal). One can see that in virtual credit scheme, the length of the queue close to link #2 is equal to the sum of the queues of link #1 and link #2 in the basic credit-based scheme.

4 Reducing Maximum Queue Length

As discussed before, the large size of the queues is a weakness of the virtual credit scheme as described so far. In this section we suggest three alternative ways to reduce the maximum possible queue length, which may be combined together.

In the first method the source bounds the total number of real data cells that are circulating using an end-to-end window mechanism on top of the virtual credit. In this method, when a source is prevented from sending a forward data cell (due to the window size restriction) it continues to send dummy data cells (in order to ensure that the virtual network would reach the correct allocation, and to prevent other sessions from exceeding their fair allocation). This method is natural in the sense that a maximum value end-to-end sliding window is used in most current transport layer mechanisms (TCP in particular). Hence, for reasonable values of window size there is no network underutilization while the good features of the bandwidth allocation are still in effect. A natural selection for the window size is the product of the session Round Trip Time and the total bandwidth of the smallest link on the session path.

The second method applies the adaptive buffer allocation method [KBC94, OSV95, HNS+94, KM95] in the virtual network. According to the adaptive buffer allocation method each session gets an initial small amount of credits that might be increased according to the current link utilization (while in the static allocation each session gets an initial large amount of credits). The adaptive buffer allocation method guarantees good utilization and fairness while "consuming" smaller buffer space in the virtual network. Hence, when some sessions share the same link, it is expected that in a steady state the total amount of "buffer allocated" to them in the virtual network is reduced compared to the analogous buffer used in the static allocation. As argued in Subsection 3.2 the total amount of required buffer space (in the worst case) is roughly the sum of "virtual buffer space" over all the sessions that share that link. The "virtual buffer space" of each session is the total amount of buffer space consumed by concurrent virtual queues along its path. Hence, following the notations of Subsection 3.2 the expression is at most $\sum_{i\,\in\,\gamma(e)}\sum_{j\,\in\,uplinks(i,e)} Q_{i,j}$ where $\gamma(e)$ is the set of sessions that cross link e and $uplinks(i,e)$ represents the links on the path of Session i that preceed link e. Thus the reduction in the total virtual buffers sizes reduces the real data queue length. The reduction in the queue length of the real network is more significant in cases such as in Fig. 5, where the sessions that cross a bottleneck link share several upstream links.

The third method fixes the target utilization of the virtual network to be somewhat less than 100%. This idea, common in several rate based flow control algorithms, drains the queues faster, since in normal conditions the allocation is set slightly under the capacity. One possibility is to *fix* the utilization of the virtual network to a certain value (e.g., 95%), in a similar way to CAPC [Bar94] or ERICA [JKVG95]. Fig. 6 illustrates a case in which the utilization is fixed to 95%. It is also possible to set the target utilization of the virtual network as a function of the queue length in the real network (following the idea in ERICA+ [JKG+95], where the queue length is large then the target utilization is small and vice versa). Alternatively, one can choose to have some spare bandwidth in the virtual network that is equal to the maximum rate among all the sessions that use this link. This idea is used in Phantom [AMO96] and can be implemented using an additional greedy *phantom* session that is participating in the Round-Robin mechanism (and thus the bandwidth "consumed" by this session is the spare bandwidth that is taken from the other sessions). Note that the methods that are used to drain queues are most effective in the case when a new session (i.e., a session that was recently established or a session that was previously idle and becomes active) is added gradually, as in TCP (where the rate of a session is at most doubled during every round trip time [Jac88]).

5 Discussion and Extensions

We presented a new end-to-end flow control scheme, virtual credit, that mimics the behavior of a credit based scheme without a requirement for a queue per session on every link. This is a significant advantage from the hardware implementation point of view.

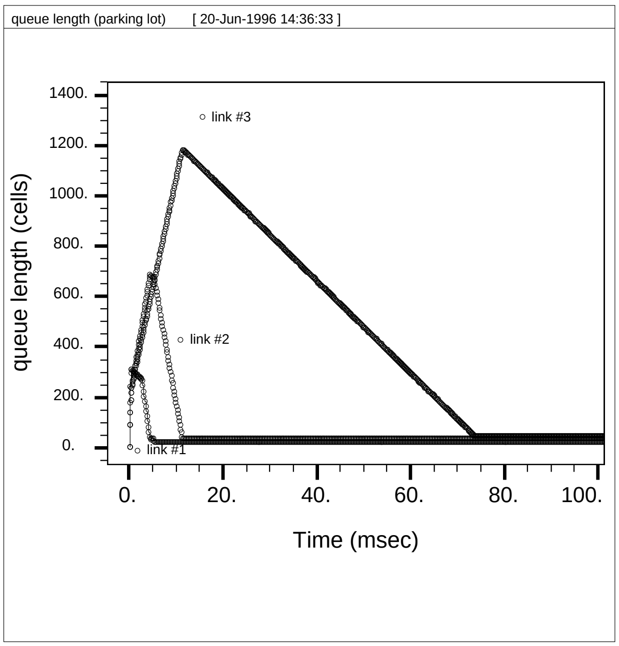
The mimicking of fair queuing in our virtual network may serve as is, to implement the "packet pair probing" technique from [Kes91] in any network. This mimicking is based upon simple Round-Robin mechanism and can be implemented with or without a queue per session mechanism. Furthermore, the virtual method easily corrects the problem of different packet lengths in fair queuing (by injecting the proportional number of dummy data cells). Thus we may use the virtual method to implement the probing and the flow-control that is based on it. A study of this direction is left for further research.

The main difference between the virtual credit and the classical credit based scheme is that in the virtual credit a larger queue may be build-up at the downstream switch of a congested link. This is not necessarily a weakness; virtual credit concentrates all the accumulation of cells at the bottleneck link, while keeping the rest of the network "free". In contrast, the standard credit based will spread the cells of the congested sessions all across the virtual circuits, allocating resources in many places.

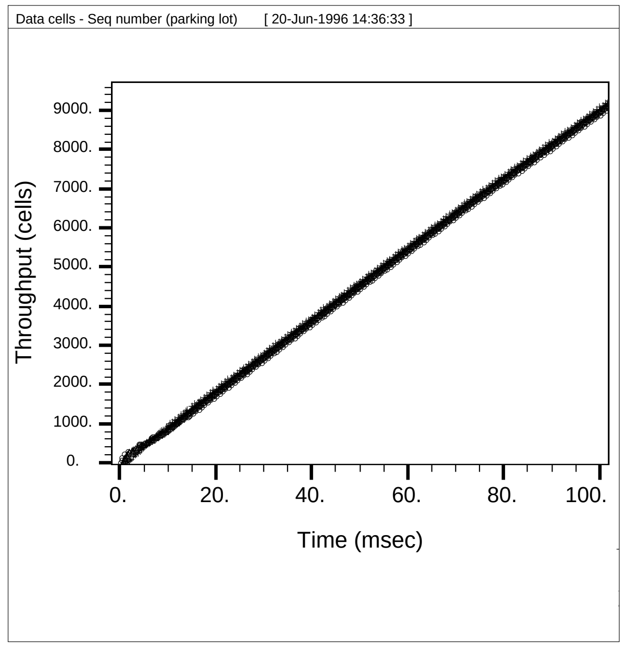
The virtual credit scheme has several advantages when compared with the rate-based scheme (most of which stem from the advantages of the classical credit based scheme [Jai95]). The first advantage is a somewhat simpler source implementation. The second advantage is the ease with which it enables the policing of the behavior of sessions by simple counting, without a queue per session mechanism. This is carried out by simply counting the number of real data cells of a session and comparing it to the total number of dummy credit cells given for that session. Since the number of cells of a well behaved session that cross a link can not exceed the number of credits supplied for this session by this link (plus some initial credits, supplied by upstream links).

The third advantage of virtual credit concerns the switch design. The simplicity of the algorithm that takes constant time for every cell that either arrives or leaves. Furthermore, the fair allocation property is straightforward. This is in contrast to several rate-based flow control algorithms that use unproven heuristics, face oscillations even in a steady states and require book-keeping operations.

Finally, the fourth advantage is that the virtual credit reacts faster than the rate based scheme. Since the rate based approach is heavily dependent on the RM cell, it has difficulty to change the rate very fast. That is, if the session has currently a "slow" rate then the time to the next RM cell may be large even if the rate may already increase. An even more severe difference is when a link on the session path becomes unavailable for *both* directions. The virtual network reacts to this change very fast (due to the mimic of the back-pressure mechanism) and avoids the entrance of new data cells to the network. In contrast, it is not clear how effective is the rate-based mechanism that is used to decrease the rate due to the absence of backward



Queue length in Virtual-Credit



Sessions' aggregate throughput

Figure 6:
Virtual-Credit - Parking lot configuration, similar to the configuration in Fig. 4. However, the utilization target of the virtual network is 95%. Note that the peak queue length is smaller than that in Fig. 4 and that the queues are drained during a stable state. The bandwidth is fairly divided among the different sessions.

RM cells [Sat96]. It is almost evident that the largest queue length is likely to show up in this scenario, and that in this extreme case, the virtual credit scheme requires less buffer space than the rate-based scheme.

On the other hand, an advantage of the rate-based scheme is that it can be implemented by algorithms that use constant space (excluding the data buffer space), i.e., the space consumed for the implementation of the control algorithm is not affected by the number of sessions that cross the link, as [Rob94, ST94, JKV94, AMO96].

It is straightforward to integrate the virtual credit scheme with credit-based networks. However, when a session enters a switch that implements the credit-based scheme after crossing several consecutive virtual credit switches it might require a larger amount of buffer space.

A more thorough examination of the behavior of this scheme is left open for further research. One issue in such examination would be to compare the performance of this scheme to both credit based and the rate based schemes within the ATM environment. Another issue would be to explore the possibilities to mimic the "packet pair probing" technique from [Kes91] in any network. In particular, this might be useful to improve both utilization and fairness in TCP networks by using relatively simple mechanisms.

References

[ALT94] ALTAGROUP. Bones designer core library reference. Technical report, December 1994.

[AMO96] Y. Afek, Y. Mansour, and Z. Ostfeld. Phantom: A simple and effective flow control scheme. In *Proc. SIGCOMM*, pages 169–182, August 1996.

[Bar94] A. W. Barnhart. Explicit rate performance evaluation. Technical Report ATM-FORUM/94-0983R1, October 1994.

[BF95] F. Bonomi and K. Fendick. The rate based flow control for available bit rate atm service. *IEEE Networks*, 9(2):24–39, March/April 1995.

[BG87] D. P. Bertsekas and R. G. Gallager. *Data Networks*. Prentice Hall, 1987.

[DKS89] A. Demers, S. Keshav, and S. Shenker. Analysis and simulation of a fair queueing algorithm. In *Proc. SIGCOMM*, pages 1 – 12, September 1989.

[GK80] M. Gerla and L. Kleinrock. Flow control: a comparative survey. *IEEE trans. on commun.*, COM-28:553–574, August 1980.

[HNS+94] D Hunt, R. Nair, J. Scott, R. Simcoe, J. C. R. Bennett, H.T. Kung, A. Chapman, and Ken Brinkerhoff. Credit-based fcvc proposal for atm traffic management. Technical Report ATM_FORUM/94-0168R1, May 1994.

[Jac88] V. Jacobson. Congestion avoidance and control. In *Proc. SIGCOMM*, pages 314–329, August 1988.

[Jai95] R. Jain. Congestion control and traffic management in atm networks: Recent advances and a survey. *Computer Networks and ISDN Systems*, February 1995.

[JKG+95] R Jain, S. Kalyanaraman, R. Goyal, Sonia Fahmy, and Fang Lu. ERICA+: Extensions to the ERICA switch algorithm. Technical Report ATM-FORUM/95-1346, The Ohio State University, October 1995.

[JKV94] R. Jain, S. Kalyanaraman, and R. Viswanathan. The OSU scheme for congestion avoidance using explicit rate indication. Technical Report ATM-FORUM/94-0883, The Ohio State University, September 1994.

[JKVG95] R Jain, S. Kalyanaraman, R. Viswanathan, and R. Goyal. A sample switch algorithm. Technical Report ATM-FORUM/95-0178R1, The Ohio State University, February 1995.

[KBC94] H.T. Kung, T. Blackwell, and A. Chapman. Credit-based flow control for atm networks: Credit update protocol, adaptive credit allocation and statistical multiplexing. In *Proc. SIGCOMM*, pages 101–114, August 1994.

[Kes91] S. Keshav. A control-theoretic approach to flow control. In *Proc. SIGCOMM*, pages 3–16, September 1991.

[KM95] H.T. Kung and R. Morris. Credit-based flow control for atm networks. *IEEE Networks*, 9(2):40–48, March/April 1995.

[OSV95] C.M. Ozveren, R. Simcoe, and G. Varghese. Reliable and efficient hop-by-hop flow control. *IEEE JSAC*, 13,4:642–650, 1995.

[Ram95] K. K. Ramakrishnan. Congestion management in high speed networks. In *ACM SIGCOMM Tutorial*, August 1995.

[RN95] K.K. Ramakrishnan and P. Newman. Integration of rate and credit schemes for atm flow control. *IEEE Networks*, 9(2):49–56, March/April 1995.

[Rob94] L. Roberts. Enhanced PRCA (proportional rate-control algorithm). Technical Report ATM-FORUM/94-0735R1, ATM Systems, August 1994.

[Sat96] S. Sathaye. Atm forum traffic management specification version 4.0. Technical Report ATM-FORUM/95-0013R10, February 1996.

[ST94] K. Y. Siu and H. Y. Tzeng. Adaptive proportional rate control (APRC) with intelligent congestion indication. Technical Report ATM_FORUM/94-0888, University of California, Irvine, September 1994.

[Tan81] A. S. Tanenbaum. *Computer Networks*. Prentice-Hall, 1981.

[TS94] H. Y. Tzeng and K. Y. Siu. Enhanced credit-based congestion notification (eccn) flow control for atm networks. Technical Report ATM_FORUM/94-0450, May 1994.