A Lossless, Minimal Latency Protocol for Gigabit ATM Networks *

Michael D. Santos, P. M. Melliar-Smith, L. E. Moser

Department of Electrical and Computer Engineering
University of California, Santa Barbara, CA 93106
E-mail: michael@alpha.ece.ucsb.edu, {pmms, moser}@ece.ucsb.edu

Abstract

Advances in fiber-optic and VLSI technology have led to the emergence of very high-speed networks based on Asynchronous Transfer Mode (ATM). The time required to transmit the data into the network at the source is small compared to the delay to propagate the data from source to destination. Cell loss is also a major concern in ATM networks because waiting for the retransmission of lost cells delays the delivery of cells and requires substantial buffer space. The Instant Start protocol eliminates the costly bandwidth reservation delay before transmission can begin. Simultaneously, it provides lossless transmission even when the network cannot handle the offered rate of transmission. Unlike other lossless protocols, Instant Start requires relatively little special control hardware or processing at each switch.

1. Introduction

Recent improvements in fiber-optic technology have provided large increases in network bandwidth, while VLSI technology has led to faster electronic switches capable of exploiting that high bandwidth. The Thunder and Lightning network project [4] at the University of California, Santa Barbara, is currently building a high-speed Asynchronous Transfer Mode (ATM) [1] switch in which each fiber-optic link operates at a rate of 40 gigabits per second (Gb/s). Technology is also being developed for 100 Gb/s links.

In [6] Kleinrock noted that the introduction of wide-area gigabit networks would cause a fundamental change in the appearance of the network to the user. In current wide-area networks (WANs), the time required to transmit data is constrained by the bandwidth of the network, and the time required to propagate data from the source to the destination is small in comparison. The delay required to establish a connection and to reserve bandwidth prior to transmission is much smaller than the connection holding time required for data transmission. For gigabit WANs of the future, the situation is reversed. The time required to propagate data from the source to the destination will remain constant because it is determined by the speed of light. With the increased transmission rate, however, the data transmission time will decrease dramatically. The propagation delay will become the dominant factor in the total connection holding time.

To maximize the benefits of gigabit networks, the protocols being used must hide the effects of latency from the user. Protocols that reserve bandwidth in advance before data transmission begins impose a minimum round-trip propagation delay. In cross-country WANs, this delay can exceed 40 ms during which time more than 200 megabytes of data can be transmitted at 40 Gb/s. Imposing a reservation delay in which no data may be transmitted will greatly reduce the effective throughput seen by the user.

Cell loss also imposes a round-trip propagation delay during which the destination must halt cell delivery as it waits for retransmitted cells to arrive. Cell loss will not only reduce the effective throughput seen by the user, but will require the destination to buffer hundreds of megabytes of data. Thus, it is imperative that cell loss be kept to a minimum.

This paper describes the Instant Start protocol, which allows a user to notify the network of the intended transmission rate and then to transmit data immediately. Bandwidth reservation is performed concurrently with the data transfer. In addition, Instant Start guarantees that no cells will be lost due to congestion (buffer overflow), thereby avoiding the lengthy retransmission delay and easing destination buffering requirements.

*This work was supported by DARPA Contract No. DABT63-93-C-0039, a National Science Foundation Graduate Research Fellowship, and a University of California Dissertation Year Fellowship.

Instant Start works most efficiently with RAM buffers, which can implement per-virtual circuit queuing. Because cells for each virtual circuit are placed in distinct logical queues within the RAM buffers, the rate of each virtual circuit can be controlled individually. However, Instant Start also supports FIFO buffering in which several virtual circuits are routed through the same physical FIFO buffer. Individual control over the rates of virtual circuits sharing the buffer is impossible because only the total output rate of the buffer can be altered. Hardware complexity is reduced at the expense of protocol complexity.

In this paper, the operation of the Instant Start protocol using per-VC queuing with RAM buffers is described. The use of the Instant Start protocol in a system with FIFO buffers is described in [13].

2. The Instant Start protocol

The Instant Start protocol is a lossless, minimal latency protocol designed as part of the Thunder and Lightning networking project. Instant Start has the following characteristics:

- Minimal virtual circuit reservation latency. Data cells must be delayed beyond call setup only long enough to guarantee that they do not overtake the setup request.
- Zero cell loss due to buffer overflow (congestion). (Cells may be lost due to hardware failure or noise, so higher level protocols must still use sequence numbers and error detection mechanisms to validate ATM Adaptation Layer frames.) A source may begin transmitting without reserving bandwidth in the network, which implies that even if there is insufficient capacity in the network to support the connection, any cells that were transmitted into the network before the call was rejected must be delivered.
- First-Come First-Served (FCFS) access to available bandwidth. Unlike ATM's ABR service class [2, 3], capacity is not equally divided among all active virtual circuits using ABR. Furthermore, a single virtual circuit must be able to exploit the full bandwidth of the network, if it is available.

2.1. Instant Start control cells

Because the Instant Start protocol was designed for the experimental Thunder and Lightning network, which does not support the ATM standard network-network interface (NNI), Instant Start control cells do not map into the standard resource management (RM) cells used by commercial ATM switches. Instead, ATM switches running the Instant Start protocol exchange a variety of protocol-specific control cells. Setup and release cells implement connection establishment. *Setup cells* are used to set up new connections and to provide the network with notification of the user's intention to begin transmission. A setup cell may also contain information about the path along which a new virtual circuit should be routed if pre-existing permanent virtual circuits are not used. *Release cells* are transmitted to discontinue an existing virtual circuit and release any resources associated with that connection.

The Instant Start protocol responds to congestion with rate adjustment and transient reduction cells. *Rate adjustment cells* effect a permanent change in the transmission rate of a source and the bandwidth allocated at each switch. In addition, rate adjustment cells trigger a flow control action at the cell receiver. *Transient reduction cells* are used to empty buffers filled due to congestion or due to complying with rate adjustment cells or other transient reduction cells.

The following sections describe the control cells used by Instant Start. Release and rate notification cells simply make adjustments to the allocation table at a switch and do not impose any flow control actions; therefore, they are omitted from the discussion.

2.1.1. Setup cells

As in most reservation-based virtual circuit protocols, a source wishing to initiate a data transfer begins by transmitting a setup cell into the network indicating the rate at which it wishes to send. Unlike standard reservation protocols, the source can begin to transmit data at the requested rate *before* it has received confirmation from the network.

Setup cell processing at each node involves software and may take 10 microseconds or more, while data cells are switched in hardware in as few as 10 nanoseconds. To prevent data cells for a virtual circuit from overtaking the setup cell establishing the connection, the source must delay transmission for at least the time to process a setup cell at each node multiplied by the number of nodes along the path. Because new connections established by the Instant Start protocol are either source-routed or use pre-established permanent virtual circuits, the source should be able to calculate an upper bound on this period of time. By delaying transmission for this interval, the setup cell will arrive at the destination followed immediately by the first data cell for the virtual circuit.

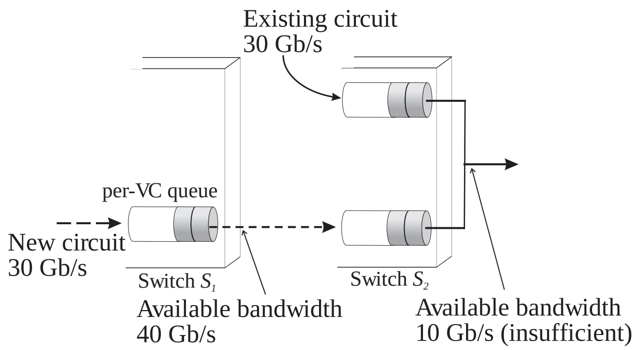

Figure 1. A new virtual circuit at a node with insufficient free bandwidth.

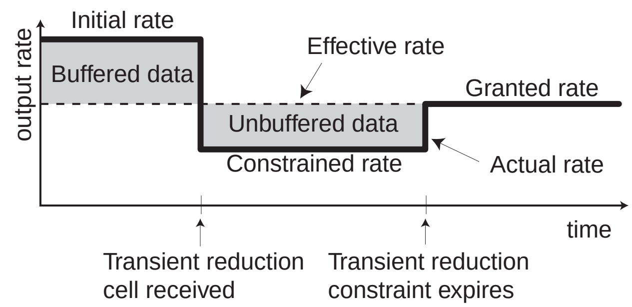

Figure 2. Per-VC queue output rate versus time.

If sufficient free capacity exists to grant the request at each node along the path, each node assigns the requested bandwidth to the new virtual circuit, decrements the available bandwidth at the output port correspondingly, and forwards the setup cell to the next switch. Transmission proceeds as if the circuit had been reserved by a standard reservation protocol, with the exception that transmission is allowed to begin without waiting for the reservation confirmation cell to return from the destination. In fact, the entire data transfer may be completed before the reservation confirmation is received at the source!

If, on the other hand, a setup cell arrives at a node with insufficient free capacity to grant the requested rate, as is the case in Figure 1, the new virtual circuit is granted the remaining free capacity. The switch modifies the requested rate field of the setup cell, to match the capacity granted, and forwards it to the next switch. The protocol must then deal with the following problems:

1. The source must be informed of the granted rate so that it will not continue to transmit at the originally requested rate.

2. To prevent buffer overflow, the upstream switch S_1 must reduce the data rate of the virtual circuit to the congested switch S_2. It cannot wait for the rate change of the source to become effective as that would require the buffering of all of the virtual circuit's excess data in the network between the source and the physical point of congestion. In the worst case, this could be the maximum data rate multiplied by the longest round-trip propagation delay. By requiring the previous switch to reduce its rate immediately, the protocol effectively distributes the buffering requirements among all switches along the path from the source to the point of congestion.

3. The buffers at the congested switch (and, in fact, all of the other switches that participated in buffering data) must be emptied; full buffers are of no use in buffering future excess data flows.

The Instant Start protocol solves problems (1) and (2) by sending a rate adjustment cell upstream containing the virtual circuit identifier and the reduced rate available to it. Each switch records the new rate for the virtual circuit in its allocation table and forwards the cell upstream toward the source.

The Instant Start protocol solves problem (3) by temporarily reducing the output rate of the previous switch below the allowed rate. The resulting output rate profile for the per-VC queue at the previous switch is shown in Figure 2. During the initial rate period, more cells are transmitted to the downstream switch than can be forwarded on the outgoing link, and the excess cells must be buffered. While the output rate of the upstream switch is constrained below the allowed rate, more cells are transmitted by the downstream switch than are received, and the buffer occupancy decreases. When data resumes arriving at the allowed rate from the upstream switch, the buffer at the downstream switch is completely empty.

This behavior is achieved by sending a transient reduction cell to the upstream switch. The congested switch calculates the excess data that must be buffered as $B = \Delta R \times 2t_p$, where t_p is the propagation delay to the previous switch and ΔR is the difference between the initial (requested) rate and the granted rate. The congested switch then chooses a reduced rate as a function of the granted rate. Given the reduced rate, the protocol calculates the time the upstream switch must maintain the rate to unbuffer B bits as $t = \frac{B}{\Delta R}$. Finally, the congested switch sends a transient reduction cell containing the virtual circuit number, restricted rate, and restriction duration to the upstream switch. The description of transient reduction cell processing is deferred until Section 2.1.3.

The function used to choose a reduced rate may be as simple as taking a strict percentage of the granted rate or may be more complex. Adjusting this function controls how quickly buffers are emptied and, in the case of FIFO buffers, the degree to which other circuits sharing the same buffer are affected. When the new virtual circuit is restricted to a small fraction of the allowed rate, the buffers will empty quickly; when the new virtual circuit is only slightly constrained, the buffers will drain gradually.

A special case occurs when a setup cell arrives at a switch with no available capacity. Although the call should be rejected to obey FCFS access to available bandwidth, the congested switch cannot simply send a rate adjustment cell with a zero granted rate back to the source. This would indeed terminate the call, but would leave data cells for the circuit trapped in buffers throughout the network.

Instead, this case is handled by granting a minimal rate to the virtual circuit to allow the buffers to empty. As usual, a transient reduction cell is sent upstream and a rate adjustment cell is sent to the source to notify it and the upstream nodes that the virtual circuit has been granted a minimal rate. In this case, the rate adjustment cell also contains a flag, which marks the virtual circuit as being rejected. When a source receives such a cell, it immediately terminates the indicated virtual circuit, but the data it sent prior to receiving the cell will eventually be delivered by the network.

To grant a minimal rate to the new virtual circuit, the same rate must be preempted from an existing virtual circuit passing through the same output port. This situation is handled as if the existing virtual circuit had sent a setup cell requesting its current rate but was granted its current rate minus the rate given to the new circuit. Both rate adjustment and transient reduction cells are sent upstream. In addition, a rate notification cell is sent downstream to inform downstream switches of the change in rate, as would have been done implicitly by a setup cell.

The preemption mechanism can easily be extended to implement virtual circuit priority. Each virtual circuit can be assigned a priority level based on the relative importance or urgency of the data being transmitted. When a setup cell arrives at a switch with insufficient free bandwidth, the new virtual circuit is allowed to preempt capacity from any existing virtual circuit of lower priority. Rate adjustment, transient reduction and rate notification cells are sent for the preempted virtual circuits as above, and the new high-priority virtual circuit proceeds as if capacity had been available.

To summarize, when congestion occurs at a switch S_2, the switch sends a rate adjustment cell upstream to inform the source and upstream switches of the rate at which the new virtual circuit may transmit. In so doing, the steady-state problem is solved. It remains, however, to solve the short-term buffering problem caused by the change in switch S_1's rate not being observed at switch S_2 until $2t_p$ seconds later. The excess data received during the delay must be buffered at switch S_2. To empty the buffers that will fill, switch S_2 sends a transient reduction cell to switch S_1, requiring it to reduce its output rate below the allowed rate temporarily. The resulting shortage of cells allows switch S_2 to empty its buffers. The data stream resumes at the allowed rate at exactly the time at which the per-VC queue in switch S_2 is emptied. Essentially, switch S_2 pushes the congestion problem one hop backward in the network. The net effect on the virtual circuit is indicated in Figure 2. The virtual circuit achieves the transfer rate it would have achieved had it known the amount of available bandwidth in advance and started transmitting at that rate originally.

2.1.2. Rate adjustment cells

Rate adjustment cells are sent upstream by a switch when a change in granted rate occurs at the switch. The change in rate may be the result of:

- A setup cell for the rate-adjusted virtual circuit that arrives at a switch with insufficient free bandwidth to grant the full requested rate,
- Bandwidth preemption due to another virtual circuit requesting bandwidth at a switch with no free bandwidth, or
- Bandwidth preemption due to a higher priority virtual circuit requesting bandwidth from a switch with insufficient available capacity to grant the full requested rate without preemption.

When a switch receives such a cell, it first forwards the cell upstream after changing the virtual circuit identifier to match that used on the upstream link. The switch then records the newly granted rate in its allocation table and reallocates the output rate of the per-VC queue accordingly.

Having done this, the switch faces an excess rate problem similar to that faced when handling setup cells: the rate of the virtual circuit entering the switch will exceed the output rate of the virtual circuit until the rate adjustment cell reaches the upstream switch and the subsequent rate change becomes effective. As when processing setup cells, the switch calculates the amount of data to be buffered as $B = \Delta R \times 2t_p$, where ΔR is now the difference between the previously recorded granted rate and the rate specified in the rate adjustment cell. Again, a reduced rate is chosen, a duration for the constraint is computed, and a transient reduction cell is sent to the previous switch.

2.1.3. Transient reduction cells

Transient reduction cells are used to empty buffers in a downstream switch. Whenever an event occurs that will cause buffering at a switch, the switch sends a transient reduction cell upstream in anticipation. Cells will need to be buffered when:

- A setup cell arrives at a switch that is unable to satisfy the full request without bandwidth preemption,
- A rate adjustment cell is received, causing the output rate of a virtual circuit at a switch to fall below the input rate temporarily (until a rate adjustment cell forwarded upstream takes effect), or
- A transient reduction cell is received, causing the output rate of a virtual circuit at a switch to fall below the input rate for a specified duration.

Upon receiving a transient reduction cell, a switch records the virtual circuit restriction and the time at which the restriction expires in its allocation table. The switch immediately adjusts the output rate of the virtual circuit by executing the rate allocation procedure described in Section 2.3. In addition, a future reallocation is scheduled to occur at the constraint expiration to reset the output rate of the virtual circuit to the granted rate.

Having adjusted its output rate to comply with the transient reduction cell, the switch then computes the effect of compliance upon the local switch buffers. The switch computes the amount of data that will be buffered as $B = \Delta R \times t_c$, where t_c is the duration of the constraint specified in the transient reduction cell. As with setup and rate adjustment cell processing, the switch arranges for the buffers to be emptied by sending a transient reduction cell upstream.

When using RAM buffers with per-VC queues, rate adjustment and transient reduction cells differ only in the duration of their effect. Rate adjustment cells cause a permanent change in the granted rate of a virtual circuit, while transient reduction cells require only temporary changes. With per-VC queuing, rate adjustment cells could be implemented as transient reduction cells with infinite duration. This is not the case with FIFO buffering because the two cell types have very different purposes. Rate adjustment cells solve the steady-state problem of bandwidth allocation. Transient reduction cells are used to empty buffers filled while a switch waits for its rate adjustment cells to take effect.

2.2. Integrating flow control actions

In previous sections we described control cell handling in isolation of other flow control actions to provide a clearer exposition. However, rate adjustment and transient reduction cells are most often sent together. For example, a setup cell arriving at a congested node causes the switch to send cells of both types to the upstream neighbor along the path of the new circuit. Furthermore, transient reduction cells will be generated by the reception of both the rate adjustment cell and the transient reduction cell at the upstream node. Thus, overlapping flow control actions are common.

Rate adjustment cells reduce the rate of a virtual circuit to that allowed by the network. Because a virtual circuit could be constrained to varying degrees at different points in the network, multiple rate adjustment cells for the virtual circuit may be received from different switches. A switch always uses the smallest value it receives, so handling multiple rate adjustment cells is trivial. If the rate indicated in the cell is smaller than the rate recorded in the allocation table at a switch, the rate adjustment cell is processed and forwarded; otherwise, it is discarded.

To keep transient reduction cell processing as simple as possible, the burden of reconciling multiple flow control actions is placed upon the sender of the control cell. For this reason, transient reduction cell processing is also simple. Each transient reduction cell supersedes any previous transient reduction cell for the same virtual circuit. As a result, the buffering imposed by the constraint in such a cell can easily be computed from the recorded granted rate for the circuit.

On the other hand, the sender of the transient reduction cell is responsible for keeping track of pending flow control actions it has requested. When a node transmits a transient reduction cell for a virtual circuit, it records in the allocation table entry corresponding to the virtual circuit the total number B_c of bits to be unbuffered by the action, the local time T_c in the transmitting node at which the action was requested, the duration t_c of the constraint, and the rate R_c at which the cells are to be unbuffered (ΔR in the original transient reduction computation). When sending

```
procedure HandleTransReductionCell(controlCell,
  allocationTable) {
  //Retrieve information from transient reduction cell
  vcOut = controlCell.virtualCircuitID
  constrainRate = controlCell.constrainedRate
  constrainDur = controlCell.constraintDuration
  //Retrieve information for vcOut in allocation table
  vcRecord = TableLookup(allocationTable, vcOut)
  vcIn = vcRecord.incomingVCID
  inPort = vcRecord.inputPort
  grantRate = vcRecord.grantedRate
  //Retrieve previous flow control information, if any
  if (vcRecord.prevFCRec != null)
    prevBits = vcRecord.prevFCRec.bitsBuffered
    prevStart = vcRecord.prevFCRec.startTime
    prevDur = vcRecord.prevFCRec.duration
    prevDrainRate = vcRecord.prevFCRec.rateReduction
  else
    prevBits = 0
  //Compute buffering requirements of the control cell
  bitsBuffered = (grantRate − constrainRate) ∗
    constrainDur
  //Add any remaining buffered bits from previous
  //control cell
  if (prevBits > 0 AND
    CurrentTime() < (prevStart + prevDur))
    bitsRemaining = prevBits − prevDrainRate ∗
       (CurrentTime() − prevStart)
    bitsBuffered = bitsBuffered + bitsRemaining
  //Select reduced rate and compute
  //duration of constraint
  newRate = ComputeReducedRate(grantRate)
  newDur = bitsBuffered / (grantRate − newRate)
  SendTransientReduction(inPort, vcIn, newRate,
    newDur)
  //Record flow control information in allocation table
  vcRecord.prevFCRec.bitsBuffered = bitsBuffered
  vcRecord.prevFCRec.startTime = CurrentTime()
  vcRecord.prevFCRec.duration = newDur
  vcRecord.prevFCRec.rateReduction =
    (grantRate − newRate)
  ReallocateOutputRate()
  ScheduleFutureReallocation(CurrentTime() +
    constrainDur)
}
```

Figure 3. Pseudocode to handle reception of transient reduction cells when using per-VC queuing.

a new transient reduction cell for the same virtual circuit, the new transient reduction cell will override the one transmitted previously. Consequently, the sender must increase the value of B_c by some value B' to compensate.

Figure 4 illustrates an example in which switch S_2 sends a transient reduction cell at time T_c and subsequently needs to send an additional transient reduction cell at time T_{now}. As shown, the new transient reduction cell will override the old cell at time $T_{now} + 2t_p$. Because the original constraint should have lasted until time $T_c + t_c + 2t_p$, part of the previously requested flow control action may be aborted. Specifically, if T_{now} is less than $T_c + t_c$, exactly $(T_c + t_c) - T_{now}$ seconds of the previous flow control action will not be completed. Because the original flow control action was unbuffering cells at a rate R_c, the amount of data that remains buffered by aborting the original flow control action can be calculated as

$$B' = B_c - R_c \times (T_{now} - T_c)$$

By adding B' to the amount of data to be unbuffered by the subsequent flow control action, the sender ensures that no cells remain in its buffers. The pseudocode for responding to a transient reduction cell is provided in Figure 3. The pseudocode for generating transient reduction cells in response to setup and rate adjustment cells is similar.

2.3. Allocation procedure

The allocation procedure for an output port is responsible for setting the service rates of the per-VC queues to obtain the desired output rates. For per-VC queuing, the allocation procedure simply sets the service rate for each queue to be the granted rate for the virtual circuit for which it provides buffering. If a circuit is currently constrained by an unexpired transient reduction, the service rate is set to the lower constrained rate.

The allocation procedure must run whenever a rate-change event occurs, though only the queues of the virtual circuits affected need to be updated. Thus, the allocation procedure runs whenever:

- Setup cells for new virtual circuits arrive,
- Virtual circuits are closed by a release cell,
- Rate adjustment cells are received,
- Transient reduction cells are received, or
- Constraints imposed by previously received transient reduction cells expire.

3. Related work

The Instant Start protocol is most closely related to protocols designed for use with the Available Bit Rate (ABR) service [2] of ATM networks. The two main flow-control mechanisms that have been proposed for use with the ABR service are credit-based flow control and rate-based flow control. Researchers have also proposed an integration of the two schemes [11] to provide

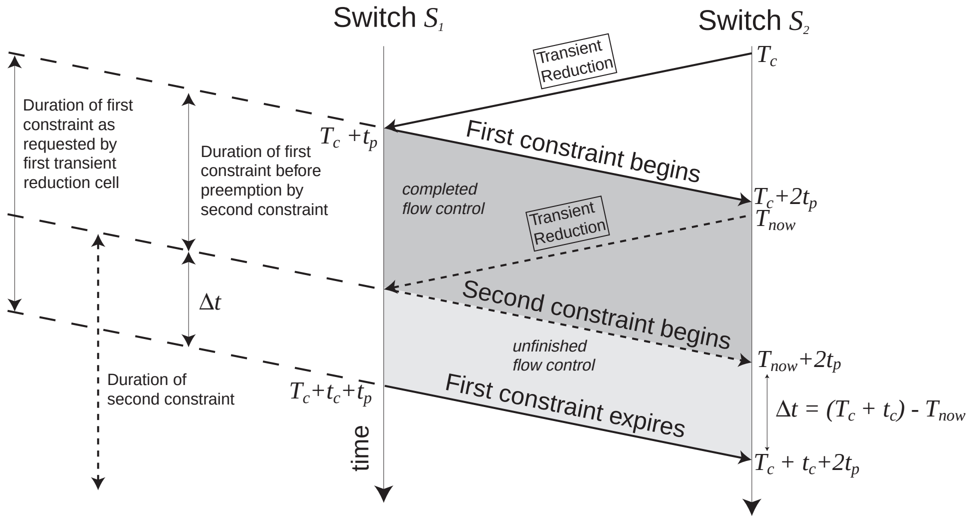

Figure 4. Overlapping transient reduction control cells.

switch designers with more flexibility in implementing flow control.

In rate-based protocols [5, 7, 10, 14], the network provides explicit feedback to the source regarding the bit rate at which it may transmit. The feedback control loop may extend from the source to the destination across the network, or individual switches may communicate with the source directly. Rate-based algorithms require little hardware support and are, thus, easy to implement, but do not provide for lossless transmission.

In credit-based protocols [8, 9], flow control is performed on a link-by-link basis. Each downstream node sends credit information to the directly connected upstream node. The credit information entitles the upstream node to send only a specified amount of data downstream. In the absence of additional credits, the upstream node must stop transmitting cells for the virtual circuit. Credit-based protocols are capable of providing lossless transmission because credits can be correlated to available buffer space downstream.

Other work has been done to reduce the inefficiencies of standard reservation protocols. Typically, capacity is reserved for a virtual circuit when the reservation request is first made, even though data will not arrive until at least an end-to-end propagation delay later. The Efficient Reserved Virtual Circuit (ERVC) protocol [15] reduces the time during which network bandwidth is reserved to exactly that used by the data transfer. This significantly increases the efficiency of using network bandwidth, but still imposes an end-to-end propagation delay before an individual virtual circuit can begin its transmission.

The Ready-to-Go Virtual Circuit (RGVC) protocol [16] eliminates the setup delay altogether. As in Instant Start, a source that wishes to begin transmission sends a setup cell into the network followed, after a short delay, by data cells. RGVC also provides lossless transmission by coupling available buffer capacity downstream with allowed transmission rate upstream. Unfortunately, the RGVC protocol requires either special hardware within the switch or a complicated list exchange procedure between switch neighbors in order to provide efficient operation.

The Instant Start protocol shares ideas with both the rate-based and the credit-based protocols. Like rate-based protocols, switches running Instant Start provide explicit feedback to sources about their allowed transmission rates in the form of rate adjustment cells. Instant Start's transient reduction cells can be interpreted as negative credit cells. In credit-based protocols, a node can transmit only when it has received credits, whereas in Instant Start a node can transmit unless it receives "anticredits" in the form of transient reduction cells.

Finally, the Instant Start protocol draws many ideas from our experience with the RGVC protocol. Indeed, Instant Start was developed after the authors first implemented a version of the RGVC protocol on both the Thunder and Lightning network simulator [12] and on a prototype of the Thunder and Lightning switch. Due to the limited processing power available within the switch, however, it was determined that the Thunder and Lightning switch was incapable of performing the required computations within the time constraints dictated by the RGVC protocol. The need for a protocol with reduced computational requirements and more efficient use of buffering capacity led to the development of the Instant Start protocol. Although Instant Start provides lossless transmission and low-latency circuit setup like RGVC, Instant Start takes advantage of information contained in the setup cell to anticipate congestion rather than react to it once it has occurred. In addition, Instant Start incorporates rate allocation to provide first-come first-served access to bandwidth that RGVC does not provide. Unlike RGVC, Instant Start does not require special hardware or complex list maintenance software.

4. Conclusion

Gigabit networks represent a fundamental change from networks of the past, where the bandwidth-limited transmission time dominated the time to complete a data transfer. Protocols that reserved bandwidth for the virtual circuit in advance worked well because the time to set up the virtual circuit was small compared to the data transfer time. In gigabit networks, however, transfers are latency limited. Consequently, the same bandwidth reservation protocols that worked well in the past will pose a costly delay for gigabit networks because the time required to reserve bandwidth may exceed the time the bandwidth is actually used.

To realize the maximum benefit from gigabit ATM networks, we must find ways of reducing the virtual circuit setup latency while simultaneously providing very low cell loss rates. In this paper we have described the Instant Start protocol, which overcomes the latency problem by overlapping the reservation and data transfer phases of communication. In addition, Instant Start provides lossless transmission even in the absence of an initial bandwidth reservation.

References

[1] ATM Forum Technical Committee. *ATM User-Network Interface Specification.* Number af-uni-0010.002. The ATM Forum, 1994.

[2] ATM Forum Technical Committee. *Traffic Management Specification.* Number af-tm-0056.000. The ATM Forum, April 1996.

[3] T. M. Chen, S. S. Liu, and V. K. Samalam. The available bit rate service for data in ATM networks. *IEEE Communications*, 34(4):56–58,63–71, May 1996.

[4] DARPA Thunder and Lightning Research Review. Technical report, University of California, Santa Barbara, March 1995.

[5] A. Dinesh and M. Moh. An efficient traffic control scheme for integrated voice, video, and data over ATM networks: Explicit Allowed Rate Algorithm (EARA). In *Proceedings of the Sixth International Conference on Computer Communications and Networks*, pages 398–403, Las Vegas, NV, September 1997.

[6] L. Kleinrock. The latency/bandwidth tradeoff in gigabit networks. *IEEE Transactions on Communications*, 30(4):36–41, April 1992.

[7] A. Koyama, L. Barolli, S. Mirza, and S. Yokoyama. An adaptive rate-based congestion control scheme for ATM networks. In *Proceedings of the 12th International Conference on Information Networking*, pages 14–19, Tokyo, Japan, January 1998.

[8] H. T. Kung, T. Blackwell, and A. Chapman. Credit-based flow control for ATM networks: Credit update protocol, adaptive credit allocation, and statistical multiplexing. *Computer Communications Review*, 24(4):101–114, October 1994.

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

[10] H. Ohsaki, M. Murata, H. Suzuki, C. Ikeda, and H. Miyahara. Rate-based congestion control for ATM networks. *Computer Communications Review*, 25(2):60–72, April 1995.

[11] K. K. Ramakrishnan and P. Newman. Integration of rate and credit schemes for ATM flow control. *IEEE Network*, 9(2):49–56, March-April 1995.

[12] M. D. Santos, P. M. Melliar-Smith, and L. E. Moser. A protocol simulator for the Thunder and Lightning ATM network. In *Proceedings of the First Annual Conference on Emerging Technologies and Applications in Communications*, pages 28–31, Portland, OR, May 1996.

[13] M. D. Santos, P. M. Melliar-Smith, and L. E. Moser. Flow control in the high-speed Thunder and Lightning ATM network. In *Proceedings of the Seventh International Conference on Computer Communications and Networks*, Lafayette, LA, October 1998.

[14] H. Y. Tzeng and K. Y. Siu. Comparison of performance among existing rate control schemes. Technical Report 94-1078, ATM Forum Contribution, November 1994.

[15] E. A. Varvarigos and V. Sharma. An efficient reservation connection control protocol for gigabit networks. In *Proceedings of the International Symposium on Information Theory*, pages 17–22, Whistler, British Columbia, Canada, September 1995.

[16] E. A. Varvarigos and V. Sharma. The Ready-to-Go Virtual Circuit protocol: A lossfree protocol for gigabit networks with FIFO buffers. *IEEE/ACM Transactions on Networking*, 5(5):705–718, October 1997.