Loss-free Communication in High-Speed Networks

Emmanouel A. Varvarigos Vishal Sharma

Department of Electrical and Computer Engineering
University of California
Santa Barbara, CA 93106

Abstract

We introduce two novel connection control protocols for high-speed networks, which we call the Efficient Reservation Virtual Circuit (or ERVC) protocol and the Ready-to-Go Virtual Circuit (or RGVC) protocol. Both protocols have been developed for the 40 Gbit/s fiber-optic ATM-based Thunder and Lightning network, currently being designed and built at UCSB. The ERVC protocol, which is designed for constant-rate, delay-insensitive sessions, uses reservations and requires little buffering at intermediate nodes, while the RGVC protocol, which is designed for variable-rate and delay-sensitive sessions, uses back-pressure to control the source transmission rate and requires buffering at intermediate nodes. In this paper, we present the main features of both protocols and indicate the advantages of each.

1 Introduction

Since the advent of high-speed integrated networks in the mid 1980's, high-speed networks of different types have been designed, built, and studied by a number of researchers in the United States (see for example [1], [2], [3], and [4]), Japan, and Europe (see for example [5], [6], [7], and [8]). The important issues of call establishment, connection control, and bandwidth management in such networks have also been an active area of research within the community (see for instance [9], [10], [11], [12], [13], [14], and [15]). Partridge [16] presents an interesting discussion of the goals and challenges of gigabit networking, which is rapidly becoming a reality with the recent advances in VLSI technology and fiber-optic transmission systems.

In designing the connection and flow control protocols for the Thunder and Lightning network, our objectives were to ensure lossless transmission, efficient utilization of capacity, minimum pre-transmission delay for delay-sensitive traffic, and packet arrival in correct order. To meet these objectives, we have proposed the Efficient Reservation Virtual Circuit protocol (or ERVC) that will be used for constant-rate sessions, or for sessions whose rate has some particular smoothness properties, and the Ready-to-Go Virtual Circuit protocol (or RGVC) that will be used for delay-sensitive traffic or for traffic whose rate changes with time.

The ERVC protocol is a reservation protocol where the durations of the sessions are recorded, and every node keeps track of the *utilization profile* of each outgoing link, which describes the amount of capacity available on the link as a function of time. This feature allows capacity to be reserved only for the duration of the session, starting at the time it is actually needed. Therefore, the protocol leads to more efficient utilization of capacity and results in lower blocking probability for new sessions than in standard reservation schemes. The ERVC protocol also has the "reservation ahead" feature, which allows a node to calculate the time at which the requested capacity will be available and reserve it in advance, avoiding in this way the wasteful repetition of the call setup phase. The protocol is robust to link and node failures, and allows soft recovery from processor failures.

The RGVC protocol can be viewed as a reservation protocol where the reservation and the data transmission phases overlap, because the source need not wait for an end-to-end round-trip delay for reservations to be made before transmitting the data. Instead the data packets follow the setup packet after a short *offset-interval*, which is much smaller than the round-trip delay (see Fig. 1). As a result, the protocol does considerably better than standard reservation protocols in terms of minimizing pre-transmission delay, and is useful for connection establishment for traffic with strict delay requirements. If the setup packet is unsuccessful in reserving the required capacity, or if the rate of the session changes without there being sufficient capacity to accommodate the change, the packets are buffered at intermediate nodes and back-pressure is exercised to the upstream nodes to control the source transmission rate. The back-pressure mechanism uses the concept of *freezing* of capacity to ensure that the protocol is lossless. The RGVC protocol can operate either with RAM buffers or with FIFO buffers at the network nodes. With RAM buffers, it is possible to throttle a particular session without affecting other sessions sharing the same buffer, because a separate logical queue can be maintained for each session. With FIFO buffers, however, it is not possible to throttle an individual session, because its packets can no longer be isolated from the packets of other sessions that share the same buffer. Thus the actions that a node takes in response to the back-pressure mecha-

IEEE Catalogue No. 95TH8061
0-7803-2579-6/95/$4.00©1995IEEE

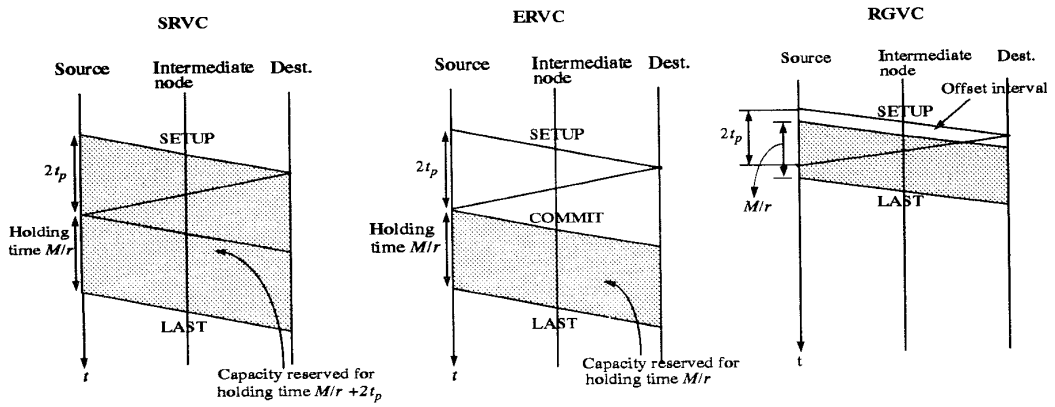

Figure 1: Illustrates the relative advantage of the SRVC, ERVC, and RGVC protocols. In the SRVC protocol, where session durations are not recorded, the capacity is blocked for duration equal to $\frac{M}{r} + 2t_p$, where t_p is the propagation delay between the source and destination. In the ERVC protocol, capacity is blocked for other sessions only for the holding time $\frac{M}{r}$. In the RGVC protocol also, capacity is occupied for time $\frac{M}{r}$ plus the time offset between the transmission of the setup packet and the first data packet of the session. The above illustrations correspond to the case where the setup packet is successful in making the appropriate reservations.

nism are more complex in the FIFO case. For brevity and ease of understanding, in the present paper, we will only focus on the operation of the RGVC protocol with RAM buffers. A discussion of the difficulties posed by the use of FIFO buffers and a detailed description of the operation of the protocol with both RAM and FIFO buffers can be found in [17].

The remainder of the paper is organized as follows. In Section 2 we explain the switch architecture that we assume. In Section 3 we present the ERVC protocol. In particular, in Subsection 3.1 we discuss why the ERVC protocol is useful for high-speed networks, and in Subsection 3.2 we outline its operation. We present the RGVC protocol in Section 4. In particular, in Subsection 4.1 we discuss the advantages of the RGVC protocol and in Subsection 4.2 we outline its operation. Concluding remarks follow in Section 5.

2 Switch architecture

In this section, we describe the switch architecture that we will refer to in our description of the protocols in the following sections.

A network switch has k bidirectional ports, each of which corresponds to an incoming and an outgoing link. Each port has a processor, called switch port processor (or SPP), which is responsible for processing the control packets flowing through the outgoing link of that port (see Fig. 2). An outgoing link transmits packets from k buffers, which can either be RAM buffers or FIFO buffers. Of these buffers, $k-1$ buffers, called *data buffers*, are used by the packets arriving on the other incoming links and intended for transmission via this link, while the k^{th} buffer, called *control buffer*, is used by control packets. The switching hardware handles the movement of the data packets through the switch without in-

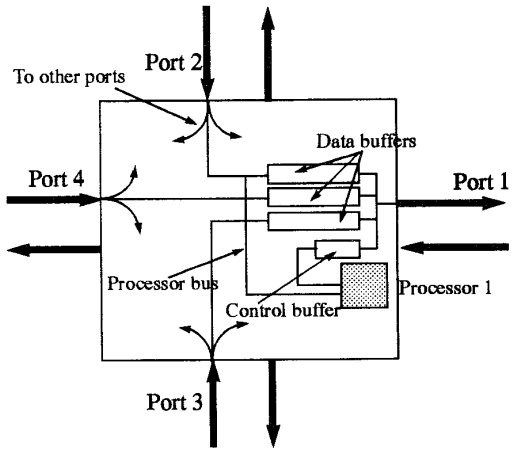

Figure 2: Illustrating the architecture of a network switch. In the above figure $k = 4$, which corresponds to the switch in the Thunder and Lightning network. Only the details of port 1 are shown.

volving the processor. The control buffer has priority over the data buffers, so that the control packets are transmitted without being affected by the data packets. If a node is a source for a session using a port j, one of the data buffers of port j is connected to it. A data buffer n at node i is denoted by $\mathcal{Q}_i(n)$. We use the notation $\mathcal{Q}_i(\mathcal{S})$ both for the buffer used by a session $\mathcal{S}$ at node i, and for the set of sessions that share that buffer.

In the Thunder and Lightning network, each switch has $k = 4$ ports, and uses FIFO buffers. Although the use of FIFO buffers poses some difficulties in the design of the RGVC protocol that are not present in the RAM case, this is the price paid for the ease of a hardware FIFO buffer implementation at speeds of 40 Gbit/s, at which RAM buffers do not

remain feasible.

3 ERVC protocol

In this section we first discuss how the ERVC protocol helps to overcome the drawbacks of standard reservation schemes and then present the main features of its operation.

3.1 Why another reservation protocol?

In standard reservation schemes the capacity required by a session at an intermediate node is reserved starting at the time the setup packet arrives at the node. This is inefficient, however, since the capacity reserved can be used only one round-trip delay after the arrival of the setup packet at the node. This is because the setup packet must travel from the intermediate node to the destination, an acknowledgement must be sent by the destination to the source, and the first data packet of the session must then travel from the source to the intermediate node (see Fig. 1). Over long transmission distances, the round-trip propagation delay may be comparable to, or even larger than, the holding time of a session. In particular, if a typical session requests capacity r bits/sec, and transfers a total of M bits over a distance of L kilometers, then the maximum percentage of time that the capacity is efficiently used in a SRVC protocol is

$$e = \frac{\frac{M}{r}}{\frac{2Lc}{\eta} + \frac{M}{r}}, \qquad (1)$$

where $c/\eta = 5\,\mu$s/km is the speed of light in fiber. Typical values of the above parameters for the Thunder and Lightning network are $r = 10$ Gbit/s, $M = 0.5$ Gbit, and $L = 3000$ km (coast-to-coast communication), which yields $e = 0.625$. In other words, for the above parameters, the capacity reserved by a typical session with a SRVC protocol stays idle for a round-trip delay of 30 ms, and is used for time equal to the holding time of a session, which is equal to 50 ms. This efficiency factor e becomes even smaller as r or L increase, or M decreases. In contrast, the efficiency factor e for the ERVC protocol can be as large as $e = 1$, independently of the parameters r, L, and M.

Consider now the situation where a setup packet requests 10 Gbit/s of capacity on an outgoing link l that has only 5 Gbit/s of capacity available at that time (see Fig. 2). If a SRVC protocol is used, such a call will be blocked. However, as shown in Fig. 2, reservations on link l are such that 10 Gbit/s of capacity will become available after 14 ms. Since the first data packets of the new session will arrive at the link after at least 30 ms (a round-trip delay), by which time the requested capacity will be available, the session should be accepted. If the ERVC protocol is used, such a call will be accepted, because each node records the session durations and can calculate

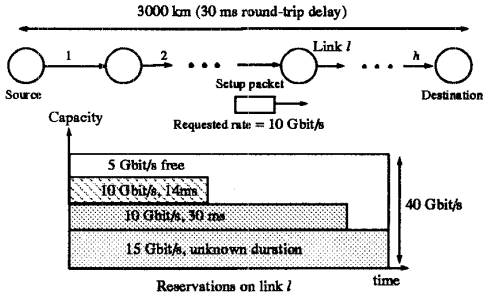

Figure 3: Disadvantage of standard reservation schemes. The 10 Gbit/s of capacity requested by the setup packet arriving at a link l will be used at least 30 ms later. If, as is the case for standard reservation protocols, no information on session durations is recorded at a node, the session will have to be rejected, because at the time the setup packet arrives the available capacity on l is only 5 Gbit/s. If, on the other hand, session durations were recorded, the session would be accepted. (In the above figure, the numbers above the links represent the hop at which the setup packet crosses that link.)

the first time at which the requested capacity will become available. Therefore, in the above situation, the setup packet will be able to reserve 10 Gbit/s of capacity starting at a time 30 ms after its arrival at the node.

The ERVC protocol also has the "reservation ahead" feature, which allows sessions to reserve capacity ahead in time and avoids repetition of the call setup phase. To see this, assume that the round-trip delay of the setup packet shown in Fig. 2 is only 10 ms, while the acceptable delay of the session is 18 ms. The setup packet now requires 10 Gbit/s of capacity on link l starting at a time 10 ms after its arrival at the intermediate node. Since 10 Gbit/s of capacity is available on link l after 14 ms and this time is within the delay that the session can tolerate, the call will be accepted on its first attempt. Thus, the reservation ahead feature avoids unnecessarily prolonged call setup phases, reduces a session's susceptibility to blocking, and leads to efficient utilization of the available capacity.

3.2 Operation of the ERVC protocol

In this section, we describe the call setup mechanism in the ERVC protocol, and describe how the reservations are made. We assume the availability of local clocks, but do not assume that the clocks are synchronized. The effects of timing uncertainties and node or link failures are discussed in [18]. We also assume, as is the case for the Thunder and Lightning network, that error control and retransmission are performed at the transport layer and are not part of the connection control protocol. New sessions are generated at each source with a specified destination, duration, and bandwidth requirement. The path followed by a session is computed by the source based on the topology and link utilization information that it has at that time.

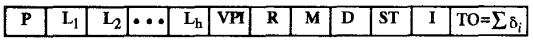

Figure 4: The SETUP packet with its fields. The first field P is the parity bit. The path of the SETUP packet is specified as a sequence of link identifiers $L_1, L_2, \ldots, L_h$, corresponding to the links that the packet must traverse. The virtual path identifier field VPI, is a 12 bit field that specifies the virtual path number of each session for a particular hop along its path. The requested rate field R is the rate required by the session. The minimum rate field M is the minimum rate with which the session will be satisfied; in general, we may have $M \neq R$. The D field is used by the intermediate nodes to decide whether the session can be accepted or not. If the intermediate node has sufficient residual capacity available starting at a time earlier than the maximum delay, the session is accepted, otherwise it is rejected.

A SETUP packet contains several fields as shown in Fig. 4. The start time field ST of a SETUP packet is initially set equal to the round-trip propagation delay $2t_p$ between the source and the destination, and is updated at each node in a manner to be described shortly. The value of the ST field of the SETUP packet when it leaves the i^{th} intermediate node is denoted by ST_i. Upon reception of the SETUP packet at an intermediate node, the ST field specifies the time (relative to the present time) at which the reservation of capacity at its outgoing link should begin. The maximum delay field D specifies the maximum delay that the session can tolerate. It is the time, relative to the time the SETUP packet starts transmission at the source, within which the source should either transmit its first data packet or learn that the session cannot be served because the delay or the capacity requirements could not be met. Since the ERVC protocol requires a pre-transmission delay equal to at least $2t_p$, it can be used only when $D \geq 2t_p$ (otherwise, the RGVC protocol is used for the session). The information field I specifies the amount of information that will be transmitted during the holding time of the session, if known. Finally, the time-offset field TO contains the time, following the reception of the acknowledgement packet at the source, after which the source should start transmission.

When the i^{th} intermediate node s_i receives a SETUP packet, it finds the first time $t \geq ST_{i-1}$ at which enough residual capacity is available to accommodate the call. If $t \geq D$ (that is, the delay until the capacity becomes available is unacceptable), the session is blocked. If $t \leq D$, the intermediate node reserves the capacity and updates the ST field to $ST_i := t$ (see Fig. 4). The node updates the TO field by adding to it the time offset $\delta_i = t - ST_{i-1}$ introduced at s_i, and forwards the SETUP packet to the next node s_{i+1}. The destination node also uses a similar algorithm to process the SETUP packet, except that instead of checking for the availability of adequate capacity, it checks for the availability of adequate memory to store the I information bits of the session. The destination acknowledges the SETUP packet by sending an ACK packet back to the source, which contains the rate finally allocated to the call

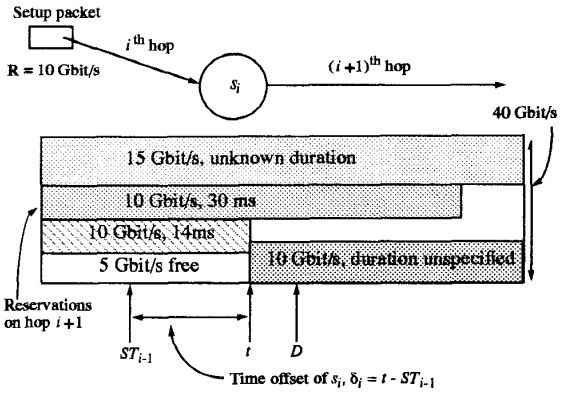

Figure 5: Illustrates the advantage of the ERVC protocol. Here the session requesting 10 Gbit/s that arrives at intermediate node s_i, is not rejected. Instead a reservation for it is made at the first time t, where $ST_{i-1} \leq t \leq D$, at which enough capacity is available. At the same time the time offset (TO) field of the SETUP packet is incremented by $\delta_i = t - ST_{i-1}$.

and the time offset $TO = \sum_{i=0}^{h} \delta_i$. After receiving the ACK, the source waits for time $\sum_{i=0}^{h} \delta_i$ and transmits a COMMIT(A, H) packet, containing the allocated rate A and the expected session holding time $H = I/A$. If the reservation is unsuccessful, the node where the session is blocked returns a REJ. If nothing is received within a time T, which is an appropriate function of the round-trip delay, REJ is assumed.

To illustrate the operation of the ERVC protocol, consider the scenario illustrated in Fig. 5. The SETUP packet is transmitted from the source s with start time field $ST_0 := 2t_p$. At each node s_i on the path, the ST field is incremented by the time offset δ_i introduced by s_i, and capacity is reserved starting at time ST_i relative to the arrival time of the SETUP packet at s_i. Therefore, in Fig. 5, link (s, s_1) is reserved starting at a time ST_0 after the SETUP packet is sent from s, link (s_1, s_2) is reserved starting at a time ST_1 after the SETUP packet arrives at node s_1, and link (s_2, t) is reserved starting at a time ST_2 after the SETUP packet arrives at node s_2. When the packet arrives at the destination t, the ST field is $ST_2 = 2t_p + \delta_1 + \delta_2$. After receiving the ACK, the source waits for $\delta_1 + \delta_2$ time units before transmitting the COMMIT packet and the data packets. This ensures that when the data packets of the session arrive at an intermediate node, capacity is available for them. In Fig. 5 above, the COMMIT packet arrives at node s_1, δ_2 time units after the starting time of the reservation on link (s_1, s_2), and it arrives at node s_2 exactly at the starting time of the reservation on link (s_2, t).

4 RGVC protocol

In this section we first discuss the advantages of the RGVC protocol and then present its main features.

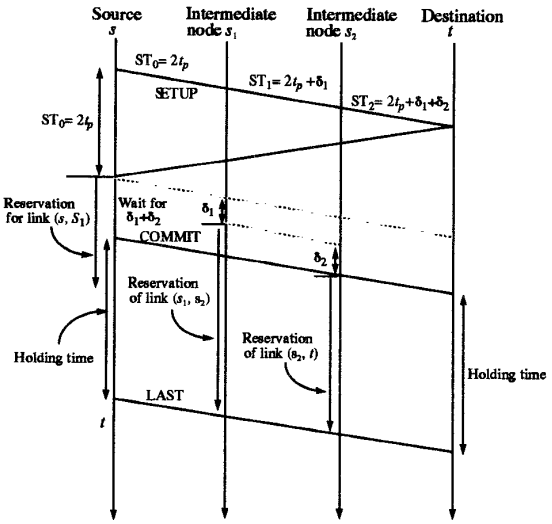

Figure 6: Illustrates the timing considerations for the ERVC protocol. When the first packet arrives at an intermediate link, the capacity for the session will be there provided that the source starts transmission after time $2t_p$. The figure above assumes that there are no timing uncertainties.

4.1 Advantages of the RGVC protocol

In the RGVC protocol, since the data packets follow the setup packet after a short *offset-interval* (see Fig. 1), a pipelining is achieved between the setup phase and the data-transmission phase, which reduces the pre-transmission delay to the minimum possible. This differs from standard reservation virtual circuit (SRVC) protocols, where a reservation phase with duration at least equal to one round-trip propagation delay between the source and the destination is needed before the data transmission phase can begin (this can be as large as 30 ms for coast-to-coast communication).

If the setup packet is successful in reserving the required capacity and the session rate remains constant, the RGVC protocol resembles a usual reservation protocol, with the added advantage that the capacity is blocked for other sessions for a much smaller time than in SRVC protocols, because capacity is reserved only for the duration of the session plus the duration of the short offset-interval. If the residual capacity of a link on the path is less than the requested rate, or if the rate of the session changes with time, the session is granted the maximum permissible rate and back-pressure (the details of which we provide in Subection 4.2) is exercised to control the source transmission rate. Therefore, even if the required rate is unavailable, the connection is not refused, but is established at a lower rate, which is gradually increased to the requested rate as capacity becomes available. Back-pressure is exercised by buffering the excess packets at the intermediate node and transmitting to the previous nodes a throttle packet that asks them to reduce their rate appropriately.

4.2 Operation of the RGVC protocol

As for the ERVC protocol, new sessions arrive at the source with a specified destination and bandwidth requirement, and based on the network information that it has available at that time, the source decides the path along which to route the session.

The SETUP packet, which is transmitted first over the path to reserve the required capacity and set the routing tables, is followed after the offset-interval by the data packets. The offset-interval is equal to the number of hops on the path times the processing time of a setup packet at a node, and is estimated to be less than 1 ms in the Thunder and Lightning network. Once a setup packet is processed at a switch and makes the needed reservations, the data packets can be routed through the switch without any (or with minimal) processing overhead. Therefore, the offset-interval is the minimum time by which the connection setup phase and the data transmission phase must be separated to ensure that the data packets do not overpass the setup packet.

If the residual capacity at an intermediate node is sufficient to accommodate the session, the node reserves capacity for that session and forwards the SETUP packet along the path. Otherwise, the node reserves the maximum available capacity for the session, reduces the requested rate field to the rate granted to the session, and forwards the the SETUP packet to the next node. The node also transmits to the previous node on the path a THROTTLE packet, which informs it that the entire capacity requested by the session could not be allocated to it, and requests appropriate actions to control the transmission rate of the session. The previous node *freezes* the capacity that could not be allocated to this session at the next node, which means that this capacity is not available for use by other sessions until the buffer space corresponding to it at the next node becomes free. Therefore, a key concept in the RGVC protocol is that the capacity available on a link is coupled with the free buffer space at the next node in order to ensure that no buffer overflow occurs. The way this is done is described shortly for RAM buffers, and is described in [17] for FIFO buffers. A node *defreezes* this capacity only when, based on its estimates, the buffer space occupied by packets of this session at the next node becomes free. Each node has for every incoming link, buffer space equal to at least $2t_pC$, where t_p is the propagation delay on that link and C is the link capacity. The exact buffer requirements depend on whether the FIFO or the RAM implementation of the protocol is being used. When the source receives the THROTTLE packet, it stops transmission for a time sufficient to clear the buffers at the nodes ahead of the packets accumulated at them due to the throttled session.

In addition to the control packets mentioned previously, two other control packets used by the protocol are the REFRESH and the LAST packet. A

REFRESH packet is transmitted periodically by the source and informs the intermediate nodes on the path that the connection is active. The LAST packet is transmitted by the source after all data packets, and signals that the session has terminated, which allows the intermediate nodes to release the reserved capacity and update the routing tables.

In the RAM case, since we can control the rate of an individual session, the operation of the protocol can be described for a particular session $\mathcal{S}$. The path followed by session $\mathcal{S}$ is denoted by $s_0, s_1, \ldots, s_h$, where s_0 is the source node and s_h is the destination node. To describe the operation of the protocol, we will use the following notation for the variables associated with a particular session $\mathcal{S}$.

$R_i(\mathcal{S})$ = allowable rate for session $\mathcal{S}$ at node s_i.
$B_i(\mathcal{S})$ = buffer space occupied by packets of session $\mathcal{S}$ at node s_i.
$A_i(\mathcal{S})$ = available capacity for session $\mathcal{S}$ at node s_i.
$F_i(\mathcal{S})$ = capacity *frozen* due to session $\mathcal{S}$ at node s_i.
$R^i_{i+1}(\mathcal{S})$ = allowable rate for session $\mathcal{S}$ at node s_{i+1}, as known at node s_i.
$B^i_{i+1}(\mathcal{S})$ = buffer space occupied by packets of session $\mathcal{S}$ at node s_{i+1}, as known at node s_i.
$R^i_{i-1}(\mathcal{S})$ = allowable rate for session $\mathcal{S}$ at node s_{i-1}, as known at node s_i.

When the rate of a session at a node s_i (including the source) changes, a RATE packet containing $R_i(\mathcal{S})$ is sent to the next node on the path. If the available capacity at node s_{i+1} is sufficient, $R^{i+1}_i(\mathcal{S})$ is updated to $R_i(\mathcal{S})$, and the node forwards the RATE packet. If the available capacity is insufficient, node s_{i+1} allocates (as described later) capacity $R_{i+1}(\mathcal{S})$ to session $\mathcal{S}$ and buffers the excess. At the same time, a THROTTLE packet containing the new rate $R_{i+1}(\mathcal{S})$ and the buffer space $B_{i+1}(\mathcal{S})$ occupied at node s_{i+1}, is sent to node s_i. Node s_i then updates the frozen capacity due to session $\mathcal{S}$ according to

$$F_i(\mathcal{S}) = \frac{B^i_{i+1}(\mathcal{S}) + D_i(\mathcal{S}) - 2t_{i,i+1}R^i_{i+1}(\mathcal{S})}{2t_{i,i+1}}, \quad (2)$$

where $D_i(\mathcal{S})$ is the amount of data that node s_i transmitted to node s_{i+1} in the $2t_{i,i+1}$ seconds prior to the arrival of the THROTTLE packet (we explain shortly how to obtain the estimates $D_i(\mathcal{S})$), and $2t_{i,i+1}$ is the round-trip propagation delay between node s_i and node s_{i+1}. The estimate $B^i_{i+1}(\mathcal{S})$ of the buffer space at node s_{i+1} as known at node s_i does not include $D_i(\mathcal{S})$ because this data was in transit (and had therefore not arrived) when node s_{i+1} sent the THROTTLE packet. Note that if $R^i_{i+1}(\mathcal{S})$ does not change (in which case $F_i(\mathcal{S})$ will be updated again) $B^i_{i+1}(\mathcal{S}) + D_i(\mathcal{S}) - 2t_{i,i+1}R^i_{i+1}(\mathcal{S})$ is the buffer space that will be occupied at node s_{i+1} due to session $\mathcal{S}$. The THROTTLE packet sent from node s_{i+1} to node s_i also has a field to record the cumulative buffer space $\sum_{j=i+1}^{h-1} 2t_{j,j+1}F_j(\mathcal{S})$ occupied at the nodes ahead, and is used by the source to calculate the time for which it should cease transmission (see Fig. 6).

The capacity $A_i(\mathcal{S})$ available for a session $\mathcal{S}$ at the outgoing link of node s_i is equal to

$$A_i(\mathcal{S}) = C - \sum_{\substack{\text{all } \mathcal{S}' \\ \mathcal{S}' \neq \mathcal{S}}} R_i(\mathcal{S}') - \sum_{\mathcal{S}' \in \mathcal{Q}_{i+1}(\mathcal{S})} F_i(\mathcal{S}), \quad (3)$$

where $\mathcal{Q}_{i+1}(\mathcal{S})$ is the set of sessions that use the same buffer at node s_{i+1} as session $\mathcal{S}$.

Observe that in the above equation the available capacity excludes frozen capacity, that is, the capacity that corresponds to the buffer space occupied at node s_{i+1} by packets of sessions that use the same buffer as session $\mathcal{S}$. Thus, the total permissible input rate of buffer $\mathcal{Q}_{i+1}(\mathcal{S})$ is always less than what it has space to store at that time without overflow. When the capacity available on its incoming link is equal to $A_i(\mathcal{S})$, buffer $\mathcal{Q}_{i+1}(\mathcal{S})$ has space to store $2t_{i,i+1}A_i(\mathcal{S})$ bits, which is adequate to store the (maximum of) $2t_{i,i+1}A_i(\mathcal{S})$ bits that node s_{i+1} may receive before the input rate of session $\mathcal{S}$ becomes equal to its output rate and further accumulation of packets of session $\mathcal{S}$ at $\mathcal{Q}_{i+1}(\mathcal{S})$ stops. This is because it takes time $2t_{i,i+1}$ for a THROTTLE packet sent by node s_{i+1} to travel to node s_i, for node s_i reduces its rate to $R^i_{i+1}(\mathcal{S})$, and for the reduction in rate to become effective at node s_{i+1}. Thus, freezing of capacity ensures that the communication is lossless.

Finally, node s_i allocates the rate $R_i(\mathcal{S})$ to session $\mathcal{S}$ according to

$$R_i(\mathcal{S}) = \min(A_i(\mathcal{S}), R^i_{i-1}(\mathcal{S}) + \frac{B_i(\mathcal{S})}{T_p}), \quad (4)$$

where T_p is a parameter, which controls the rate at which the buffer at node s_i should be emptied. The above choice of the rate ensures that the outgoing rate of session $\mathcal{S}$ at node s_i includes the incoming rate plus the rate at which the node should transmit to empty its buffer within time T_p.

When the source node receives the THROTTLE packet, it stops transmission until its estimate of the buffer space occupied by packets of this session at the nodes of the path becomes zero. That is, the source ceases transmission for a time equal to $\sum_{j=0}^{h-1} 2t_{i,i+1}F_j(\mathcal{S})/G_{\mathcal{S}}$, where $G_{\mathcal{S}}$ is the rate granted to the session at the node at which the THROTTLE packet originated (both quantities are contained in the THROTTLE packet), which allows the buffers at the nodes ahead to empty (see Fig. 6).

We now explain how a node s_i obtains an estimate of the amount of data $D_i(\mathcal{S})$ that it transmits to node s_{i+1} in an interval $2t_{i,i+1}$. Since the rate of a session $\mathcal{S}$ at a node s_i will be a step function, conceptually the node can obtain this estimate simply by calculating the area under the rate-function (that is, the function which represents how the rate of a session changes with time) of session $\mathcal{S}$. In practice, an efficient way for the node to do this is to maintain a linked list, each of whose elements stores the rate and the corresponding time, at which the rate-function made jumps in the $2t_{i,i+1}$ seconds preceding the current time. In other words, the linked

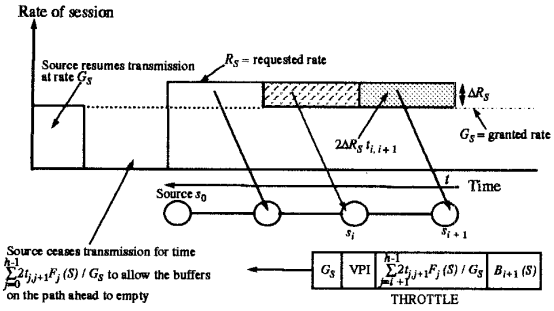

Figure 7: Illustrates the actions of the source node on receiving a THROTTLE packet. The x-axis represents time. In the above figure, the rate available at s_{i+1} at time t is equal to G_S , while the rate at which packets arrive at s_{i+1} is $G_S + \triangle R_S$. Thus, at time t node s_{i+1} transmits a THROTTLE packet. At time $t + t_{i,i+1}$, node s_i receives the THROTTLE packet, reduces its rate to G_S, and forwards the THROTTLE packet to node s_{i-1}. Finally, at time $t + \sum_{j=0}^{j=i} t_{j,j+1}$, the source receives the THROTTLE packet and ceases transmission for time $\sum_{j=0}^{h-1} 2t_{j,j+1}F_j(\mathcal{S})/G_S$, which clears the buffer space occupied by session $\mathcal{S}$ at the nodes on the path.

list can be thought of as a sliding window of length $2t_{i,i+1}$, which captures a snapshot of the profile of the rate-function in the last $2t_{i,i+1}$ seconds. The data transmitted to the subsequent node is then obtained by using these rates and times to calculate the area under the rate-function in the last $2t_{i,i+1}$ seconds.

5 Concluding remarks

We have presented the main features of the ERVC and the RGVC connection control protocols designed for the 40 Gbit/s Thunder and Lightning network, currently under development at UCSB. We illustrated how the ERVC protocol is particularly suited to take advantage of the long propagation delays in very-high speed long haul networks, and we indicated the lossless character of the RGVC protocol by introducing the concept of *freezing* of capacity. We point out that our description assumed that at one time only one of the protocols operates in the network, or alternatively, that there are two sets of buffers at each port, one set for constant-rate sessions (that use the ERVC protocol) and another set for delay-sensitive variable-rate sessions (that use the RGVC protocol). When the switches at the nodes use FIFO buffers that are shared by sessions using the two protocols, their packets cannot be isolated from one another and are no longer independent. We are currently investigating this issue to see whether a unified design combining the important features of both the protocols can be proposed.

References

[1] I. Cidon and I. S. Gopal, "Paris : An approach to integrated high-speed private networks", *Int'l J. Digital and Analog Cabled Systems*, vol. 1, no. 2, pp. 77–86, Apr-June 1988.

[2] D. L. Tennenhouse and I. M. Leslie, "A testbed for wide area atm research", in *Proc. ACM SIGCOMM'89*, Austin, TX, September 1990, pp. 182–190.

[3] I. Cidon, I. Gopal, P.M. Gopal, R. Guérin, et al., "The planet/orbit high-speed network", *Journal of High Speed Networks*, vol. 2, no. 3, pp. 171–208, 1993.

[4] D. D. Clark, B. S. Davie, D. J. Farber, and I. S. Gopal, "The aurora gigabit testbed", *Computer Networks and ISDN Systems*, vol. 58, pp. 599–621, May 1970.

[5] M. Devault, J. Y. Cochennec, and M. Servel, "The "prelude" atd experiments : assessments and future prospects", *IEEE J. Selected Areas Commun.*, vol. 6, no. 9, pp. 1528–1537, December 1988.

[6] D. J. Greaves, D. Lioupis, and A. Hopper, "The cambridge backbone ring", in *Proc. IEEE Infocom90*, San Fransisco, CA, June 1990, pp. 8–14.

[7] B. Butscher and J. Kanzow, "The berkom project: a b-isdn field trial in berlin", in *Proc. 10th Int'l Conference on Computer Communication*, New Delhi, India, Nov. 1990, pp. 4–9.

[8] B. Pehrson and L. Gauffin, "Multig distributed multimedia applications and gigabit networks", in *Proc. Int'l Switching Symp.*, Yokohama, Japan, 25-30 October 1990.

[9] A. Segall and J. M. Jaffe, "Route setup with local identifiers", *IEEE Trans. on Comm.*, vol. 34, no. 1, pp. 1019–1028, January 1986.

[10] J. Y. Hui, "Resource allocation for broadband networks", *IEEE J. Selected Areas Commun.*, vol. 6, no. 9, December 1988.

[11] I. Cidon, I. S. Gopal, and R. Guérin, "Bandwidth management and congestion control in planet", *IEEE Communications Magazine*, vol. 29, no. 10, pp. 54–64, October 1991.

[12] P. E. Boyer and D. P. Tranchier, "A reservation principle with applications to the atm traffic control", *Computer Networks and ISDN Systems*, vol. 24, no. 4, pp. 321–324, May 1992.

[13] D. P. Tranchier, P. E. Boyer, Y. M. Rouaud, and J. Y. Mazeas, "Fast bandwidth allocation in atm networks", in *Proc. Int'l Switching Symposium*, Tokyo, Japan, 25-30 October 1992.

[14] I. Cidon, I. S. Gopal, and A. Segall, "Connection establishment in high-speed networks", *IEEE/ACM Trans. on Networking*, vol. 1, no. 4, pp. 469–481, August 1993.

[15] L. Gun and R. Guérin, "Bandwidth management and congestion control framework of the broadband network architecture", *Computer Networks and ISDN Systems*, vol. 26, no. 1, pp. 61–78, September 1993.

[16] C. Partridge, "Protocols for high-speed networks: some questions and a few answers", *Computer Networks and ISDN Systems*, vol. 25, no. 9, pp. 1019–1028, June 1993.

[17] E. A. Varvarigos and V. Sharma, "A loss-free connection control protocol for the Thunder and Lightning network", March 1995, submitted *Globecom'95*.

[18] E. A. Varvarigos and V. Sharma, "An efficient reservation connection control protocol for gigabit networks", submitted *IEEE/ACM Trans. on Networking*, January 1995.