CONGESTION CONTROL IN ISDN FRAME-RELAY NETWORKS

Bharat T. Doshi and Han Q. Nguyen

Bharat T. Doshi *is a supervisor in the Performance Analysis Department at AT&T Bell Laboratories in Holmdel, New Jersey. Work in his group involves performance analysis and traffic engineering for switching, data communication, computer, and production systems. His research interests include queueing theory and stochastic processes and their applications to performance analysis. He received a B.Tech. degree in mechanical engineering from the India Institute of Technology, Bombay, and a Ph.D. in operations research from Cornell University. He joined AT&T in 1979.* ***Han Q. Nguyen*** *is a supervisor in the Data Architecture Planning Department at AT&T Bell Laboratories in Holmdel, New Jersey. His group is responsible for overall architecture and protocol planning for* (continued on page 46)

Like X.25 packet networks, ISDN frame-relay networks require effective congestion control mechanisms to cope with unanticipated network component failures and overloads. Unlike X.25 packet networks, ISDN frame-relay networks perform the requisite packet-switching function *without* terminating the link and network layer data-transfer protocols. They therefore cannot use delayed acknowledgment and/or receiver-not-ready indications embedded in these protocols for congestion control. This paper reviews the measures that can be used to control congestion effectively in ISDN frame-relay networks.

Introduction

In ISDN frame-relay virtual-circuit networking, to take advantage of the higher speed and quality of digital transmission facilities, the network's protocol processing function in the virtual-circuit data-transfer phase is streamlined to minimize transit delay and maximize throughput. At each network node, only the core procedures of LAPD are performed in the relay process. That is, incoming LAPD frames are checked only for valid frame-check-sequence (FCS) and address fields; invalid frames are simply discarded, while valid frames are switched ("relayed") toward their destination on the basis of their virtual-circuit identity as indicated in the frame address. The remaining LAPD protocol procedures and the layer 3 protocol—including, in particular, error-correction (via retransmission) and flow-control procedures—are left to operate between the LAPD end points on an end-to-end basis.

Thus, unlike today's X.25 packet networks, ISDN frame-relay networks cannot make use of delayed acknowledgment and receiver-not-ready (RNR) indication for congestion control. Consequently, formulating effective and efficient alternative congestion controls is particularly important in the architectural design of ISDN frame-relay networks.

In this article, we review design considerations and discuss control techniques for ISDN frame-relay networks that use a virtual-circuit-based edge-to-edge internodal relay architecture. Some of

Panel 1. Terms Used in This Paper

FCS	frame check sequence
FIFO	first-in, first-out
ISDN	Integrated Service Digital Network
LAN	local-area network
LAPD	link-access procedures for D channel
REJECT	A message sent from receiver to transmitter when an out-of-sequence frame is received
RNR	receiver not ready
RR	receiver ready

the techniques are common to all virtual-circuit networks and thus will be described only briefly. Others have been developed for the particular constraints and flexibility of ISDN frame-relay networks; these will be discussed in greater detail.

Congestion Control Objectives

Network component (node, trunk) failures and unanticipated high traffic demand are potential causes for congestion in packet networks in general and in frame-relay networks in particular. When a network component fails, retransmissions initiated by the end points increase traffic levels on the network paths that lead to the failed component. Excessively high traffic levels, whether due to failure recovery attempts or network component service oversubscription, can cause network transit delays to exceed the end-to-end acknowledgment time-out values, and/or frames to be discarded by the network nodes because of buffer overflow. (The term *oversubscription* is used loosely to refer to the condition when too many virtual circuits have been set up or when a moderately large number of virtual circuits transmit simultaneously—a rare occurrence.) Acknowledgment time-outs and frame loss (and out-of-sequence errors) in turn increase the frequency of retransmission, further aggravating congestion and potentially causing it to spread.

Without some form of congestion control, as first pointed out by Kleinrock[1] for packet networks and demonstrated by Rege and Chen[2] for ISDN frame-relay networks, the network's *useful* throughput would greatly decrease while the network transit delay would grow unacceptably large. This performance degradation is not discriminatory. All users will suffer service degradation even if the congestion is caused by a few "overactive" users, unless special control measures are taken to ensure fairness.

The objectives of an overall control strategy are:

1. The probability of congestion during the data-transfer phase must be kept low. If congestion does develop, the adverse effects on the network and user-perceived performance should be minimized.
2. The network must not have to rely on cooperation (i.e., voluntary flow-rate reductions) from the end points to protect itself against congestion collapse, although such cooperation would enhance performance.
3. The controls should be effective over a wide range of network speeds [e.g., 56/64 kilobits per second (kb/s), T1.5, T45] and traffic characteristics.
4. The controls should not interfere with the natural data-transfer operations in the absence of congestion, and should incur only negligible overhead.

Elements of an Overall Control Plan

A comprehensive congestion avoidance and control plan consists of distributed real-time controls, centralized network-management (near real-time) controls, and long-term network engineering procedures and practices.

Distributed real-time controls are exercised by the network nodes and the end points. These controls can be characterized by their roles (preventive controls versus reactive controls) and their operating time scales (slow-acting versus fast-acting), both of which directly depend on the operational traffic units (virtual circuit versus frame).

At the high end of the operating time scale (seconds to minutes), the network will attempt to prevent congestion by spreading traffic load for each given source-destination node pair among a set of multiple alternate paths in setting up new virtual calls. (For permanent virtual circuits, load balancing is done as part of the

engineering and provisioning process.) If all alternate paths for a source-destination node pair already are loaded sufficiently in comparison with the available bandwidth and buffer resources, then new call-setup requests between that node pair will be denied. In the event that severe congestion develops and persists at some network component, the network may attempt to reroute selected existing virtual calls locally around the congested component or onto completely different alternate paths. If the rerouting cannot be done (e.g., in case of global overload) then some existing virtual calls may be disconnected.

At the low end of the operating time scale (milliseconds to seconds), control techniques are applied by the network to minimize "hogging" of resources by any particular virtual circuit and to maximize useful throughput if a network component becomes overloaded. The control techniques include throughput enforcement, buffer management, internodal trunk service discipline, and frame discarding, all of which are discussed later in this paper.

Because no distributed control scheme can be expected to deal flawlessly (or even adequately) with each and every unanticipated network event, it is necessary to supplement the distributed controls with centralized network management's override capabilities (i.e., to change the distributed nodal controls' parameter values and/or block or unblock their actions). The override capabilities are invoked according to global network status to fine-tune the distributed controls to the particular network situation as necessary. For example, automatic controls such as virtual-circuit rerouting and virtual-call disconnecting are initiated under centralized network management's close supervision and control, since these controls can have severe impact on existing traffic. Because of their centralized and global nature, override controls have a latency of several minutes.

With these controls, a network should be able to protect itself adequately against congestion. Of course, if the end points cooperate by shedding their loads when the network becomes congested, performance will be improved even more.

Note that, together, the above control mechanisms can only optimize traffic performance within the confines of the available network resources. Whenever the available network resources fall short of the service demand level, traffic performance (e.g., virtual call-setup blocking rate) would still be unavoidably degraded. Consequently, long-term network engineering procedures, usually done on a weekly or monthly basis, must be relied upon to ensure that network resources are adequate to keep the probability of traffic overload (from unanticipated failures or statistical surge in service demand) as low as possible without making the network service uneconomical.

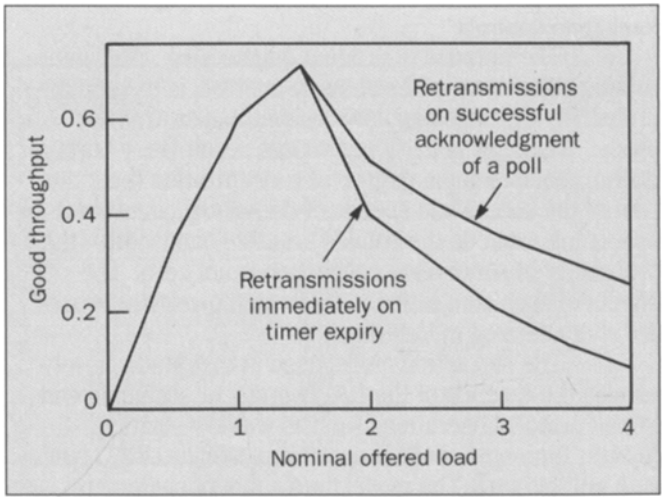

Figure 1. Good throughput with and without polling.

Overall, network engineering procedures, network management, and routing-related controls used for today's packet-switched, virtual-circuit networks, such as X.25 networks, can be applied directly to ISDN frame-relay networks. However, X.25 networks rely on the flow-control mechanisms made available through layer 2 and 3 protocol termination to effect real-time congestion control. For ISDN frame-relay networks, alternative real-time control mechanisms must be designed and evaluated. We now focus on these alternative real-time controls.

Real-Time Controls

The purpose of prudent engineering, intelligent routing algorithms, and call-setup controls is to keep the probability of congestion low during the data-transfer phase. When low activity per virtual circuit (very bursty traffic) requires some degree of concentration (i.e., the sum of the access-line speeds of the virtual circuits set up on a trunk exceeds the trunk's available bandwidth), the probability of short-term congestion is not zero. The effects of such congestion in ISDN frame-relay networks are characterized in Reference 2.

The simulation model used in that study closely mimics the essence of the LAPD protocol, including end-to-end protocol operations such as window rotation, rejects, time-outs, polls [i.e., receiver-ready (RR) frames with poll-bit set]. The model uses a mix of character-interactive, block-interactive, and file-transfer types of traffic. The study presents two cases. In the first, a LAPD acknowledgment timer expiry (T200 = 1 second) initiates retransmission of all unacknowledged frames. In the second, a poll is sent every time the T200 timer expires, and retransmissions are initiated only when acknowledgment for the poll frame is received. The results, illustrated in Figure 1, show that, without any additional control measures, the good throughput on the congested trunk drops dramatically as the congestion level increases. The throughput drop for the second scheme is much less severe than in the first scheme but is still not acceptable. This suggests two things:

- Polling on the expiry of T200 timer is strongly recommended for all end points and should be made mandatory, if possible.
- If the congestion-level and buffer-size scenarios studied in Reference 2 are likely, a real-time congestion control mechanism is necessary to maintain high useful throughput during periods of congestion.

Other effects of short-term congestion demonstrated in Reference 2 and in other studies of similar networks[1,3-5] include violation of delay requirements for delay-critical applications (e.g., echoplexing), throughput and delay degradation for all virtual circuits although only a few overactive virtual circuits are responsible for congestion, spread of congestion to initially uncongested components because of excessive retransmissions, and session disconnections caused by T200 timers running out too many times.

In light of these potential effects on the network and user-perceived performance, real-time congestion controls should have the following objectives:

1. Maintain a high level of useful throughput by minimizing time-outs and out-of-sequence deliveries.
2. Prevent spread of congestion.
3. Protect well-behaved users from the misbehaved ones. In the event of a general overload, divide the "pain" equitably. This is what is generally referred to as *fairness*. The issue of fairness has been the focus of many studies of flow and congestion control problems.[6] However, its definition remains vague. Thus, it is defined practically as *preventing hogging of resources by a small number of users.*
4. To the extent possible, provide delays consistent with the service objectives (especially for very delay-sensitive applications).
5. Prevent session disconnections unless desired for congestion control.

In addition, real-time controls should foster network self-reliance, robustness, and efficiency.

We will now discuss a set of congestion control mechanisms and their effectiveness in meeting the above objectives during the data-transfer phase. These schemes can be broadly categorized as follows:

- Controls that reduce load, maintain delay objectives, and ensure a degree of fairness through the natural elasticity in window-based end-to-end protocol, adequate buffer provisioning, and queue management.
- Controls that (1) involve explicit or implicit detection of congestion by the congested component or by the end points and (2) take actions to reduce the load offered to the congested component, to minimize retransmissions, and to preserve a degree of fairness. The control

Table I. Buffer Size and Maximum Delay

Access speed C_a (b/s)	Trunk speed C_L (b/s)	Per-virtual-circuit activity level α: 0.01 Buffer (kB)	0.01 D_{max} (s)	0.05 Buffer (kB)	0.05 D_{max} (s)	0.25 Buffer (kB)	0.25 D_{max} (s)	0.50 Buffer (kB)	0.50 D_{max} (s)
16 k	64 k	113	14.1	22.5	2.81	4.5	0.56	2.25	0.28
16 k	1.544 M	2020	10.5	404	2.11	81	0.42	40	0.21
64 k	1.544 M	960	5.0	192	1.0	38	0.20	19	0.10
64 k	64 k	68	8.5	14	1.7	2.7	0.34	1.35	0.17
1.544 M	1.544 M	620	3.23	124	0.65	25	0.13	12.4	0.07
1.544 M	45 M	14,207	3.08	2841	0.62	568	0.12	284	0.06

Round-trip propagation delay = 60 ms; LAPD frame size = 136 bytes; Links in the connection = 6; kB = kilobytes; s = seconds

Table II. Buffer Size and Overflow Probability with Dedicated and Shared Buffers

Access speed C_a HP (kb/s)	LP (Mb/s)	Trunk speed C_L(Mb/s)	Buffer size, kilobytes: Fully dedicated	Hybrid HP	Hybrid LP	Probability, HP overflow: Normal (0.5)	Overload (2.0)
16	1.544	1.544	1716	15	58	7.4×10^{-23}	2.8×10^{-6}
64	1.544	1.544	887	25	58	3.4×10^{-22}	2.5×10^{-6}

HP = high priority; LP = low priority

actions may be in the congested component, in the end points, or in both.

Assumptions. The design of a network's distributed controls is highly dependent on the particular network's *internodal* relay architecture, i.e., the procedure by which user-data frames are relayed from one node to the next between the network edges. Just as with today's packet networks, frame-relay networks' internodal relay architectures are likely to be vendor-proprietary and vary significantly from one vendor's network to another. Plausible architectures could range from a quasi-datagram-based approach with sufficient built-in procedural precaution to ensure a negligible probability of virtual-circuit frames arriving out of sequence, to a virtual-route-based approach, to a virtual-circuit-based approach.

In this article, we assume that the network uses a straightforward virtual-circuit-based internodal relay architecture. At each node, the incoming line/trunk identity (ID) and logical-link address (in the LAPD address field) of an incoming frame are used to map into the frame's outgoing line/trunk ID and logical-link address. The outgoing logical-link address is substituted for the incoming logical-link address in the frame's LAPD address field before the frame is sent out. This procedure requires each switch

(including, in particular, tandem switches) to maintain *per-virtual-circuit* translation records for all of its virtual circuits in order to perform the frame-relay operations during the data-transfer phase. With this type of internodal relay architecture, the transmitting and receiving protocol handlers at each network node can identify the virtual circuit to which a particular frame belongs and, therefore, can apply appropriate control measures with the degree of selectivity required to maintain fairness.

For other internal protocols in which the internal network nodes do not have access to the identity of the virtual circuit sending a given frame, some aspects of congestion control (especially the actions in the end points) remain similar. Others are different in the sense that some actions are taken at the network edge rather than at the congested component.[7]

The other assumption we make is that the end points use a window-based protocol (the LAPD I-frame procedures or similar procedures) above the LAPD core procedures. For ease of discussion, we choose to use the LAPD I-frame procedures as an example.

Implicit Controls. For any window-based protocol, the delays caused by higher occupancy of trunk transmit buffers during congestion automatically slow down window rotation, reducing the offered load. If each trunk buffer is sized to hold the entire window's worth of frames for each virtual circuit that the call setup control allows, then, as long as the T200 timers do not run out, no frames will be dropped and the natural elasticity of the protocol will achieve the desired load shedding. Also, since no frames are dropped due to buffer overflow, no bandwidth is wasted for retransmissions and the ideal network throughput objective is achieved.

Two factors affect the feasibility of full buffer dedication: (1) the size, cost, and management complexity of the buffer and (2) the delay induced by a large buffer under congestion and the resulting T200 timer expiries, which may cause retransmissions and/or session disconnects. In addition, unless the buffers are managed with a sophisticated selective service strategy, the resulting delays will not be acceptable for delay-sensitive applications.

Table I shows the total buffer size and worst-case frame delay [under first-in, first-out (FIFO) discipline] for various access-line speeds, trunk speeds, and activity levels. We assume that the call-setup control limits the number of virtual circuits so that the trunk is busy no more than 50 percent of the time in the long run. It seems that when the ratio of the trunk speed C_L to the access-line speed C_a is not very large and the activity level α is high enough, full buffer dedication is indeed possible. At higher values of C_L/C_a and lower activity levels, the required buffer size is large and the worst-case delay is much bigger than the default value of the T200 timer (1 second). If polling on timer expiry is made mandatory, timer expiry may not result in immediate retransmissions. However, frequent expiry of timers may result in a LAPD end point disconnecting the session. Finally, the delays shown in Table I for large C_L/C_a and small α may not be acceptable for the application mix even if priorities are used to provide better service to some applications at the expense of others.

Suppose the traffic mix consists of a large number of virtual circuits having high values of C_L/C_a and short bursts of data, coupled with a small number of virtual circuits having small values of C_L/C_a and longer bursts of data. For example, the former may be interactive and low-speed file-transfer traffic and the latter may be very high speed file transfers and LAN bridge traffic. Then the above discussion suggests the following strategy: For the former type of virtual circuits (type I), provide a shared buffer sized to keep the probability of overflow small. For the latter (type II), dedicate a full window's worth of buffer for each virtual circuit set up on the trunk. Serve type I traffic with nonpreemptive priority over type II traffic. Higher priority allows a small shared buffer without causing a high probability of overflow. Lower priority to type II traffic magnifies the natural elasticity and slows down the truly high-speed virtual circuits for which the possibility of a buffer overflow has been eliminated. Although there is a small probability that the type I buffer can overflow, the large number of virtual circuits involved makes the event statistically less likely. In addition, since type I virtual circuits are assumed to send short bursts of data, congestions caused

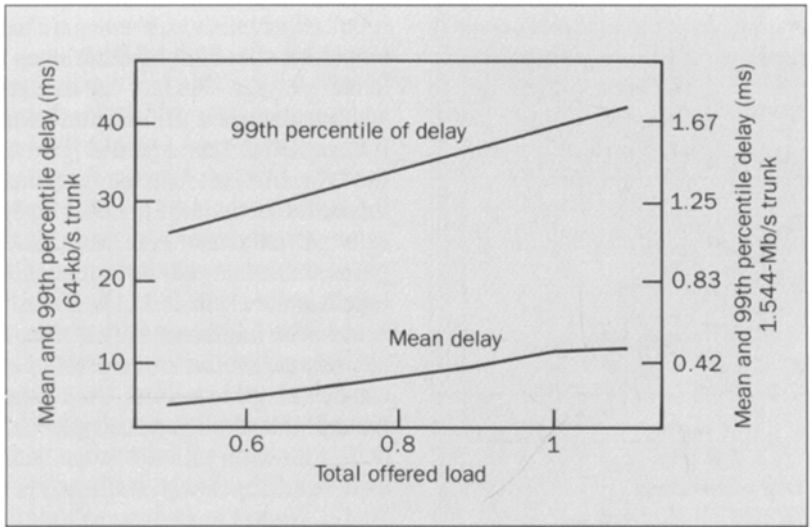

Figure 2. Mean and 99th percentile delay for a 10-byte frame in high-priority queue. The high-priority traffic mix is 90 percent 10-byte frames and 10 percent 136-byte frames.

by statistical fluctuations will be of short duration and will not have significant detrimental effects if they remain infrequent. Table II illustrates the saving in buffer size possible using this scheme. (Rege and Chen[8] give details of the model and more extensive results.) If the primary danger of congestion comes from relatively few high-speed virtual circuits (access line speed close to the trunk speed), then this scheme is very attractive.

If a sizable fraction of traffic comes from high-speed virtual circuits with relatively low activity level, then the above scheme does not help to reduce the buffer requirement. Also, even if most of the virtual circuits are of type I, unusual events may cause unpredictably high (in a statistical sense) type I traffic. This results in extended periods of congestion, buffer overflow, and retransmissions. Some form of additional real-time control is then necessary. We will discuss those controls under "Explicit Real-Time Controls."

Even when full buffer dedication is economically feasible and typical delays can be maintained below the T200 timer value, the delays may not be acceptable for some applications. Moreover, with the full buffer dedication, a few overactive virtual circuits can cause high buffer content, higher delays for all virtual circuits, and smaller throughput even for well behaved virtual circuits. These considerations are important even under moderate to heavy load and remain important when the offered load reaches true congestion levels. Additional buffer management and service strategies are necessary to protect delay-sensitive applications and maintain fairness. The former objective can be achieved by serving delay-sensitive applications at higher priority than the others. As long as the load in the high-priority queue is kept at moderate level, the delay there will remain small even if the overall load is high. Two types of priority mechanisms are possible: *explicit* and *implicit*.

If, as discussed earlier, the primary danger of high load and congestion comes from virtual circuits that require very high speed data transfer (high-speed file transfer, LAN traffic, etc.), they can be identified at virtual-circuit setup and served at low priority during the data-transfer phase. This gives an explicit priority to the frames from other virtual circuits (type I traffic). Figure 2 shows the mean and 99th percentile of the delay seen by a 10-byte frame in the high-priority queue as functions of the total load for the trunk speeds of 64 kb/s and 1.544 mega-

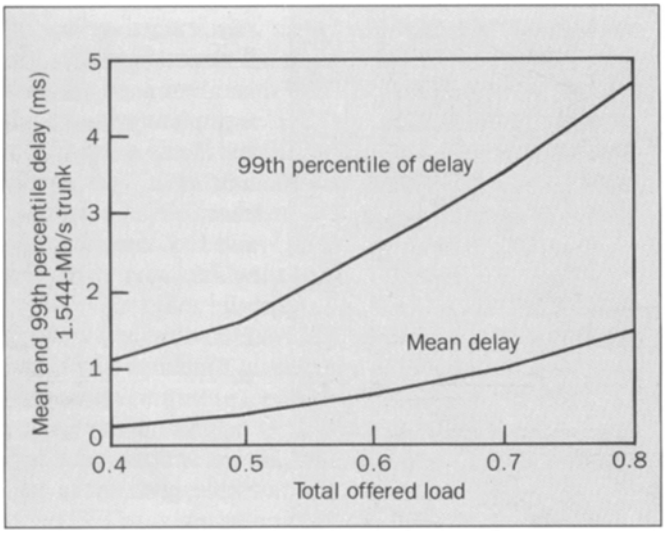

Figure 3. Mean and 99th percentile delay for a 10-byte frame over two trunks. The traffic mix is 87 percent 10-byte frames, 9 percent 48-byte frames, and 4 percent 136-byte frames.

bits per second (Mb/s) when the frame size in the low-priority queue is 136 bytes. (The mean and 99th percentile are obtained through analytical modeling and by inverting the Laplace transforms.[9]) Other parameters are specified in the figure. Clearly, excellent low-delay performance is possible for the high-priority traffic even at high overall load.

Many type I virtual circuits, however, may have mixed delay requirements. For example, a terminal may be in editing mode, requiring very short echoplexing delay; or it may be receiving a screen full of data from the host. In this latter mode, a much longer delay can be tolerated. Treating all this type I traffic in a single high-priority queue may result in high load in the high-priority queue and defeat the purpose of the priority treatment; as Figure 2 shows, this is more crucial at lower trunk speeds.

This suggests additional discriminatory treatment even within the type I traffic. Typically, low delays are required when short frames are sent infrequently, while longer bursts of full LAPD frames can tolerate somewhat longer delays. This fact can be exploited to provide an additional implicit priority to the delay-sensitive short frames within type I traffic. Two queues are maintained for the type I traffic. When a frame arrives at the trunk buffer, the buffer is checked to see if any frame belonging to the same virtual circuit is present. If there is, the new frame is put in the lower-priority queue. Otherwise, the frame length is checked. If it is below a threshold, the frame is sent to the higher-priority queue. If it is above the threshold, it is put in the lower-priority queue. The type II traffic can be served in a third, even lower-priority, queue. Variations of this implicit priority mechanism have ensured low delay for character interactive traffic in AT&T data switches.[10] For analysis of such schemes, see References 10-12. While the analysis in References 10 and 11 refers to a byte-stream protocol, extension to a LAPD-based protocol is immediate.

The need for this implicit priority scheme depends on the aggregate type I traffic and the trunk speed and becomes less crucial at higher trunk speeds. Suppose, for example, that the type I traffic is limited to 80 percent of the trunk capacity even under congestion. Then, as shown in Figure 3, the 99th percentile of the delay over two trunks with FIFO service for type I traffic is under 5 milliseconds (ms) for 1.544-Mb/s trunks. This delay for 64-kb/s trunks would be about 120 ms and very sensitive to the load in the queue, as well as to the traffic mix and the frame lengths. Additional implicit priority at lower trunk speeds can maintain the load in the highest priority well below 80 percent and provide robust delay performance.

While priority queueing gives excellent service to delay-sensitive applications even under overall congestion, it does not protect some overactive virtual circuits from degrading delay and throughput performance of other virtual circuits in the low-priority queue. If all virtual circuits (other than those sending very delay-sensitive traffic) are equally important (e.g., if they have the same throughput

class), serving the low-priority queue in a round-robin manner will allow roughly equal throughput to all virtual circuits under heavy congestion.[13]

This is demonstrated in Table III, which shows the throughput allocation among four hypothetical virtual circuits, two of which offer more load than the others, under various total traffic-load levels. If the virtual circuits have different legitimate throughput requirements, the round-robin service can be modified so that more than one frame (depending on the throughput class of the virtual circuit) can be transmitted from a virtual circuit during each shot at the service. Another suitable mechanism for protecting well-behaved virtual circuits and ensuring a degree of fairness is to monitor the buffer occupancy (and/or bandwidth usage) by virtual circuit and serve the recently overactive virtual circuits from a third, lowest-priority queue. Since this allows all the queues to be served FIFO, the queue management is simplified. This may make it more attractive at higher trunk speeds.

Overall, when a full window's worth of buffer dedication is economically feasible and does not cause T200 timer expiry problems, it is possible to maintain high overall throughput even under congestion. In addition, with appropriate priority structure and service discipline, it is possible to protect delay-sensitive applications effectively and maintain a degree of fairness under congestion. In fact, in this case the congestion control mechanism is built directly into the call-setup controls and buffer-management strategies. No explicit actions are needed during periods of short-term congestion.

Explicit Real-Time Controls. As mentioned earlier, it is not always feasible to dedicate a full window's worth of buffers for every virtual circuit and achieve all congestion-control objectives automatically. When the sum of the windows for the virtual circuits set up on a trunk exceeds the trunk transmit-buffer size, the buffer can overflow during periods when the sum of the access-line speeds of active virtual circuits exceeds the trunk bandwidth. A frame dropped due to buffer overflow may cause the following frames on the same virtual circuit to be received out of sequence, resulting in a REJECT and retransmission of up to a whole window's worth of frames. If a whole window's worth of data gets dropped, the T200 timer will expire and a poll will be sent. If this succeeds, the whole window will be retransmitted. These retransmissions result in a dramatic drop in good throughput, as observed in Reference 2. Schemes are necessary to detect congestion and minimize its adverse effects. The following observations suggest some effective control mechanisms:

- During congestion, the effective throughput on the trunk drops because it uses some of its bandwidth to transmit frames which will be received out of sequence and will have to be retransmitted in any case. A strategy which identifies such frames and discards them, even if they do not cause buffer overflow, will reduce the throughput degradation. The frames received on a virtual circuit immediately after a frame is dropped for that virtual circuit because of buffer overflow are candidates for dropping. Such a selective discard strategy can be enhanced further to monitor buffer occupancy or bandwidth usage of virtual circuits and to penalize overactive virtual circuits by initiating discards at a lower threshold for them.
- Buffer overflow, frame losses, and retransmissions

Table III. Throughput Equity under Round-Robin Discipline

	Nominal peak offered load from VCs			Trunk utilization due to VCs		
Case	1 and 2	3 and 4	Total	1 and 2	3 and 4	Total
1	0.06	0.41	0.47	0.06	0.41	0.47
2	0.47	0.823	1.293	0.41	0.588	0.998
3	0.615	1.273	1.888	0.454	0.546	1.000
4	0.889	1.556	2.445	0.487	0.513	1.000
5	1.2	2.0	3.2	0.500	0.500	1.000

Note: Four VCs; VCs 1 and 2 offer less load than VCs 3 and 4.
(VC = virtual circuit.)

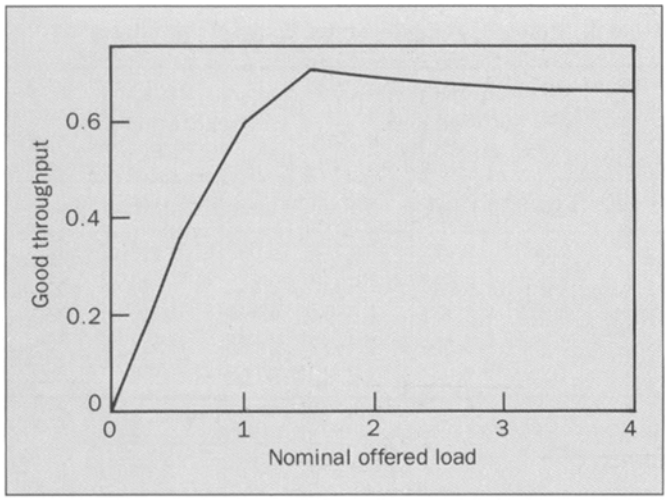

Figure 4. Good throughput with a selective discard control. Scheme ensures fair treatment.

occur because the sum of the window sizes for the currently active virtual circuits exceeds the available buffer size. Frame losses can be minimized if the effective window can be reduced for each virtual circuit and/or the number of virtual circuits simultaneously transmitting data to the trunk can be reduced. Reducing the window also reduces the number of frames to be retransmitted in the event of a frame loss.

Network-based control. A selective discarding scheme of the type discussed above was included in the simulation mentioned earlier.[2] The results, illustrated in Figure 4, show excellent good-throughput performance. As mentioned above, this scheme also permits throughput enforcement and hence fair treatment under congestion. In addition, this type of control is network-based and does not rely on cooperation from the end point. On the other hand, frames are not dropped until they reach the congested component. This may affect the load on preceding trunks in the connections and degrade their delay performance. This potential for the spread of congestion can be reduced if the end points cooperate and reduce the load they offer to the network under congestion.

End-point-based controls. When a frame is dropped because of buffer overflow in the network, the following frame will generate a REJECT from the receiving end point. If a whole window's worth of frames is lost, the sending end point will time-out. In either case, the sending end point may assume the existence of congestion along the connection and reduce the load it is offering to the network.

One mechanism is to lower the end point's effective window size. Various window-reduction schemes have been proposed.[3-5] Typical schemes are:

- Reduce window size by 1 to W_{min}, the minimum window size.
- Reduce to W_{min}.
- Reduce to max $\{W_{min}, \alpha W\}$, where W is the current effective window size and α is a fraction, $0 < \alpha < 1$.

Successful transmissions (and acknowledgments) may indicate that the congestion has gone away and window size should be increased. Once again, various window-relaxation schemes have been proposed. For example:

- Increase by 1 up to W_{max}, the maximum window size, after N consecutive successful transmissions.
- Increase by 1 (up to W_{max}) after W successful transmissions, where W is the current effective window size.

Many of these combinations have been analyzed under a variety of traffic patterns and network parameters by B. Barbour, K.-J. Chen, and K. M. Rege at Bell Laboratories. Chen et al. report on some of this work in Reference 14. Overall, a scheme that reduces W to W_{min} (typically 1) and increases it by 1 after a fixed number of successive successful transmissions performs best when the nominal window size is small (3 or 4). Close to this scheme in performance is a scheme that reduces the window to half its current value instead of to W_{min}. When the nominal window size is large (10 or more), however, the latter scheme outperforms the former except under very heavy congestion. Both schemes achieve very good overall throughput performance.

Table IV. Good Throughput with and without Fairness Enforcement

Nominal offered load	Good throughput under various control strategies		
	A	B	C
1.5	0.68	0.53	0.69
3.0	0.66	0.50	0.68
4.5	0.64	0.45	0.65

A: All VCs adapt windows, no network control
B: Half the VCs adapt windows, no network control
C: Half the VCs adapt windows, fairness enforced at congested component

Another mechanism to reduce the offered load is to stop transmitting for a fixed or random duration on receiving a REJECT, or on T200 timer expiry. With enough randomness, this will reduce the number of virtual circuits simultaneously sending data to the same trunks. When the voluntary stop period expires, the virtual circuit may begin with window size 1 and change the size in increments as in the scheme discussed above.

Adaptive schemes in the end points and selective discard schemes at the congested trunk can work together synergistically to achieve all the objectives of congestion control. Endpoints reduce the offered load at the source. The actions by the congested component guarantee high good throughput, even if some end points do not adapt, and enforce fair treatment of all virtual circuits. For example, Table IV shows the throughput performance under three scenarios: A—all virtual circuits adapt their windows with no network control; B—only half of the virtual circuits adapt their windows with no network control; C—only half of the virtual circuits adapt their windows, and the network nodes employ a discard strategy in which the discard threshold for a virtual circuit depends on the number of frames it has in the transmit buffer. The effectiveness of the throughput enforcement mechanism is obvious from the results. In addition, the priority queueing and service discipline (round-robin, for example) can protect delay-sensitive applications and provide a fairer delay/throughput performance.

Controls with network–end-point interaction. In previous sections, we discussed controls in which both congestion detection and control action are at the same place (at the congested component or in the end point). Usually, congestion detection is more accurate in the congested component (for example, a frame received out of sequence does not mean that congestion has been encountered, even if it results in a REJECT). On the other hand, load shedding is more effective in the end points (it cuts the traffic at the source before any of the network resources are used on a frame that will be dropped eventually). This suggests that improved performance is possible if the congested component detects the congestion (high buffer or trunk occupancy) and communicates to the end points via advisory "messages," and the end points react to these "messages" by load-shedding actions. Load can be shed by adapting windows or stopping for a duration. The results in References 2 and 14 indicate excellent throughput performance from such schemes. However, such schemes involve additional complexity and overhead and need "message" standardization; the amount of complexity and overhead depend on the particular "message" conveyance mechanism chosen. These should be considered in deciding whether to use advisory "messages" for short-term real-time control. In the event of longer-term, more widely spread congestion, a slower-acting control involving advisory "messages" may be very effective as part of the overall network management.

Summary

Congestion is always a possibility in any data communications network (in any service system, for that matter) that allows some degree of sharing for economic reasons. Prudent engineering, path diversity, adaptive routing, and call-setup controls can minimize but cannot, in general, eliminate the potential for congestion during the data-transfer phase. Also, these long-term and near-real-time preventive controls (as well as reactive network-management-type controls) are similar, in general, for all virtual-circuit networks (e.g., X.25). It is during the data-transfer phase, where a real-time reactive control is needed to minimize the adverse effects of congestion, that the differences between ISDN frame-relay and X.25-based networks are more important. Thus, we have concentrated

our discussion on real-time controls. Unlike X.25-based networks, ISDN frame-relay networks do not terminate level 2 or 3 in any network node and thus cannot use delayed acknowledgments and/or RNR for real-time congestion control. Our discussion shows that controls that, together, maintain high effective throughput, protect delay-sensitive applications, and ensure a degree of fairness under congestion can be provided for ISDN frame-relay networks.

Acknowledgment

Special acknowledgments are due to K.-J. Chen, K. Rege, and B. Barbour for their analysis of many of the controls discussed in this article. In addition, we have benefited from technical discussions with A. E. Eckberg, A. G. Fraser, D. T. Luan, D. M. Lucantoni, and D. Sheng.

References

1. L. Kleinrock, "On Flow Control in Computer Networks," *International Conference on Communications*, Toronto, June 1978.
2. K. M. Rege and K.-J. Chen, "A Performance Study of LAPD Frame-Relay Protocols for Packetized Data Transfer over ISDN Networks," *Fifth International Teletraffic Conference Seminar*, Lake Como, Italy, 1987.
3. W. Bux and D. Grillo, "Flow Control in Local-Area Networks of Interconnected Token Rings," *IEEE Transactions on Communications*, Vol. COM-33, No. 10, October 1985.
4. R. Jain, "A Timeout-Based Congestion Control Scheme for Window Flow-Controlled Networks," *IEEE Journal on Selected Areas in Communications*, Vol. SAC-4, No. 7, October 1986.
5. R. Jain, K. K. Ramakrishnan, and D. Chiu, "Congestion Avoidance in Computer Networks with a Connectionless Network Layer," *Tenth Data Communications Symposium*, October 1987.
6. M. Gerla, H. W. Chan, and J. R. B. deMarca, "Fairness in Computer Networks," *International Conference on Communications*, Chicago, June 1985.
7. D. M. Lucantoni and D. T. Luan, "Throughput Analysis of an Adaptive Window Based Flow-Control Subject to Bandwidth Management," *International Teletraffic Conference*, Turin, Italy, June 1988.
8. K. M. Rege and K.-J. Chen, "An Analytical Model for Buffer and Trunk Sizing and Severe Congestion Avoidance in LAPD Frame-Relay Networks," *Record of the International Conference on Communications*, Philadelphia, June 1988.
9. D. L. Jagerman, "An Inversion Technique for the Laplace Transform with Application to Approximation," *Bell System Technical Journal*, Vol. 57, No. 3, March 1978, pp. 669-710.
10. B. T. Doshi and K. M. Rege, "Analysis of a Multistage Queue," *The Bell System Technical Journal*, Vol. 57, No. 3, March 1978, pp. 669-710.
11. A. G. Fraser and S. P. Morgan, "Queueing and Framing Disciplines for a Mixture of Data Traffic Types," *AT&T Technical Journal*, Vol. 63, No. 6, July-August 1984, pp. 1061-1087.
12. C. Y. Lo, "Performance Analysis of a Two Priority Packet Queue," *AT&T Technical Journal*, Vol. 66, No. 3, May-June 1987.
13. E. L. Hahne and R. G. Gallager, "Round Robin Scheduling for Fairness Flow Control in Data Communication Networks," International Conference on Communications, June 1986.
14. K.-J. Chen et al., "Performance of LAPD Frame-Relay Networks: Transmission Error Effects and Congestion Control," *International Teletraffic Conference*, Turin, Italy, June 1988.

Biographies (continued)

advanced, wide-area data networks. He received a B.S. in industrial and systems engineering from Ohio University and an M.S. in operations research from Virginia Polytechnic Institute. He joined AT&T in 1978.

(Manuscript received September 29, 1988)