

Congestion Control in Frame-Relay Networks

LAN-to-LAN data-transmission strategies

William Stallings

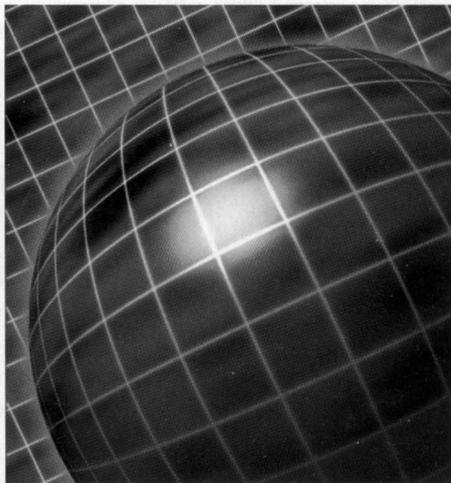
Frame relay is a standardized, public, packet-switched, data-network service that functions as a public wide area network (WAN) backbone connecting individual local area networks (LANs). Data is transmitted in packets from one LAN to another through a frame-relay network over high-speed leased lines. Because of its design, the Frame Relay ANSI T1.606 Standard allows for an easy migration path from present to future network architectures, as existing systems (including T1, ISDN, and others) can be upgraded, and hence preserved, via software.

Additionally, frame relay can manage bursty, unpredictable data traffic and provide single-line access to the network with logical connections to other destinations. The end result is minimal hardware, a simple network design, and reduced operating costs.

With all this in mind, it's no surprise that frame-relay bearer services—that is, those services designed to serve both LAN interconnections and existing host-computer environments—are becoming increasingly popular. CompuServe's Frame-Net frame-relay service, for example, enables dedicated Internet access at up to T1 speeds (1.536 Mbits/sec), and 14.4 Kbits/sec dial-up access via point-to-point protocol (PPP).

William is an independent consultant and president of Comp-Comm Consulting of Brewster, MA. He is the author of over a dozen books on data communications and computer networking, his most recent being ISDN and Broadband ISDN, with Frame Relay and ATM, third edition (Prentice-Hall, 1994). William can be reached at stallings@acm.org.

Still, the frame-relay standard does not specify a mechanism for flow control and error control between users and the network. The data-link control protocol for frame relay (LAPF) uses a frame structure that does not contain a control field and therefore has no sequence numbers to work with. While this streamlined protocol provides for efficient data transfer, it lays the network open to the possibility of congestion. To deal with this problem, standards organizations have proposed a variety of congestion-control techniques. Because of the large number of techniques



that can be used alternatively or in conjunction with one another, and because the specifications are scattered through various documents in no particular order, this is the most confusing aspect of frame relay.

In this article, I'll provide an overview of frame-relay congestion control, looking first at the explicit congestion-control techniques proposed for frame-relay bearer services, and then present a set of algorithms that split the responsibility for congestion control between the network and the subscriber.

Congestion in Frame-Relay Networks

A frame-relay network is a form of packet-switching network in which the "packets"

are layer-two frames. As in any packet-switching network, one of the key areas in the design of a frame-relay network is congestion control. In essence, a frame-relay network is a network of queues. At each frame handler, there is a queue of frames for each outgoing link. If the rate at which frames arrive and queue up exceeds the rate at which frames can be transmitted, the queue size grows without bounds and the delay experienced by a frame goes to infinity. Even if the frame-arrival rate is less than the frame-transmission rate, queue length will grow dramatically as the arrival rate approaches the transmission rate. As a rule of thumb, when the line for which frames are queuing exceeds 80 percent utilization, the queue-length growth rate becomes a problem.

Figure 1 shows the effect of congestion in general terms. Figure 1(a) plots the throughput of a network (number of frames delivered to a destination station per unit time) versus the offered load (number of frames transmitted by all subscribers), while Figure 1(b) plots the average delay from entry to exit across the network. At light loads, throughput and network utilization increase as the offered load increases. As the load continues to increase, a point is reached (point A in the plot) beyond which the throughput of the network increases at a rate slower than that of the offered load. This is due to network entry into a mild-congestion state. At this level, the network continues to cope with the load, although with increased delays.

As the load on the network increases, a point is eventually reached (point B) beyond which throughput drops with increased offered load. The reason for this is that the buffers at each node are of finite size. When the buffers at a frame handler become full, it must discard frames. Thus, the sources must retransmit the discarded frames in addition to new frames. This only exacerbates the situation: As more and more frames are retransmitted, the load on the system grows, and more buffers become saturated. While the sys-

ORBitize™

**"Using IDL just
got a whole
lot easier."**

Now that you are using an ORB, you need ORBitize™. ORBitize™ lets you quickly and effectively browse and build OMG IDL definitions with any ORB. So, to make IDL development a breeze, give us a call to find out more about ORBitize™.

Features

- Browses IDL files and interface repositories
- Easily navigate to related definitions (even across files)
- Browser displays containment and inheritance hierarchies
- Easy-to-use dialog boxes for editing each kind of IDL definition
- Complete on-the-fly validation of all OMG IDL syntax
- Includes IDL examples

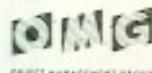
Platforms

- UNIX, MS Windows
- ObjectBroker, SOMobjects, Orbix, TME, DOE, ORBeline
- Other platforms and ORBs in progress, call for details

The Company

NetLinks Technology, Inc. is a leading ORB-independent supplier of distributed object tools, training, and consulting services.

phone: (603)594-0525
fax: (603)594-0527
email: info@netlinks.com



Distributed Object Specialists

NetLinks
Technology, Inc.

(continued from page 60)

tem is trying desperately to clear the backlog, users are pumping old and new frames into the system. Even successfully delivered frames may be retransmitted because it took too long at a higher layer (for example, the transport layer) to acknowledge them: The sender assumes the frame did not get through. Under these circumstances, the effective capacity of the system is virtually zero.

Avoiding these catastrophic events is the task of congestion control. The object of all congestion-control techniques is to limit queue lengths at the nodes so as to avoid throughput collapse.

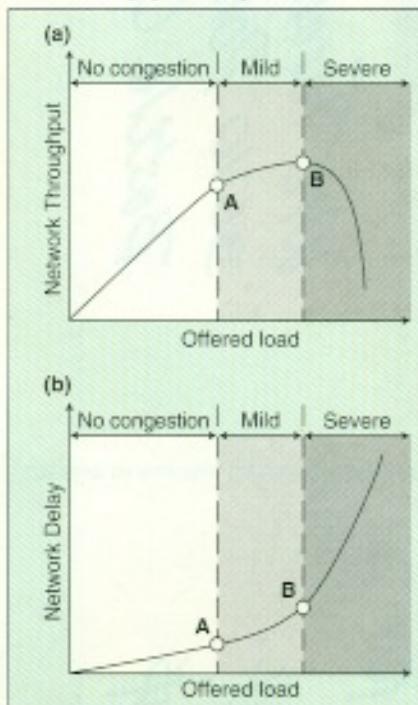


Figure 1: Effects of congestion: (a) throughput; (b) delay.

Congestion-Control Strategies

Congestion control is the joint responsibility of the network and end users. The network (that is, the collection of frame handlers) is in the best position to monitor the degree of congestion, while the end users are in the best position to control congestion by limiting the flow of traffic.

Table 1 lists the congestion-control techniques defined in the various standards documents. Discard strategy deals with the most fundamental response to congestion: When congestion becomes severe enough, the network is forced to discard frames. You would like to do this in a way that is fair to all users.

Congestion-avoidance procedures are used at the onset of congestion to minimize the effect on the network; thus, these procedures would be initiated at or prior to point A in Figure 1, to prevent congestion from progressing to point B. Near point A, there would be little evidence available to end users that congestion is increasing, so there must be some explicit signaling mechanism from the network that will trigger the congestion avoidance.

Congestion-recovery procedures are used to prevent network collapse in the face of severe congestion. These procedures are typically initiated when the network has begun to drop frames due to congestion. Such dropped frames are reported by some higher layer of software and serve as an implicit signaling mechanism. Congestion-recovery procedures operate around point B and within the region of severe congestion, as shown in Figure 1.

Congestion avoidance with explicit signaling and congestion recovery with implicit signaling are complementary forms of congestion control in the frame-relaying service.

Type	Technique	Function	Key Elements
Discard strategy	Discard control	Provides guidance to network concerning which frames to discard.	DE bit
Congestion avoidance	Backward explicit congestion notification	Provides guidance to end systems about congestion in network.	BECN bit or CLM message
Congestion avoidance	Forward explicit congestion notification	Provides guidance to end systems about congestion in network.	FECN bit
Congestion recovery	Implicit congestion notification	End system infers congestion from frame loss.	Sequence numbers in higher-layer PDU

Table 1: Frame-relay congestion-control techniques.

(continued from page 62)

Discard Strategy

As a last resort, a frame-relaying network must discard frames to cope with congestion. There is no getting around this fact. Since each frame handler in the network has finite memory available for queuing frames, it is possible for a queue to overflow, necessitating the discard of either the most recently arrived frame or some other frame.

The simplest way to cope with congestion is for the frame-relaying network to discard frames arbitrarily, with no regard to their source. In that case, since there is no reward for restraint, the best strategy for any individual end system is

to transmit frames as rapidly as possible. This, of course, exacerbates the congestion problem.

Network Use of CIR

To provide for a fairer allocation of resources, the frame-relay bearer service includes the concept of a committed information rate (CIR)—a rate, in bits per second, that the network agrees to support for a particular frame-mode connection. Any data transmitted in excess of the CIR is vulnerable to discard in the event of congestion. Despite the use of the term "committed," there is no guarantee that even the CIR will be met. In cases of extreme congestion, the network may be forced to

provide service at less than the CIR for a given connection; however, the network will choose to discard frames on connections that are exceeding their CIR before discarding frames on those that are not.

Explicit Congestion Avoidance

It is desirable to use as much of the available capacity in a frame-relay network as possible but still react to congestion in a controlled and fair manner. This is the purpose of explicit congestion-avoidance techniques wherein the network alerts end systems to growing congestion within the network and the end systems take steps to reduce the offered load to the network.

As the standards for explicit congestion avoidance were being developed, two general strategies were considered. One group believed that congestion always occurred slowly and almost always in the network egress nodes. Another group had seen cases in which congestion grew very quickly in the internal nodes and required quick, decisive action to prevent network congestion. These two approaches are reflected in the forward and backward explicit congestion-avoidance techniques, respectively.

With congestion-avoidance techniques, the network signals congestion to those end users with affected frame-relay connections. This explicit signaling may make use of one of two bits in the LAPF address field of each frame, or a special LAPF control message. Either bit may be set by any frame handler that detects congestion. If a frame handler receives a frame in which one or both of these bits are set, it must not clear the bits before forwarding the frame. Thus, the bits constitute signals from the network to the end user. The two bits are:

Backward Explicit Congestion Notification (BECN) bit. The user is notified that congestion-avoidance procedures should be initiated where applicable for traffic in the opposite direction of the received frame. BECN indicates that the frames the user transmits on this logical connection may encounter congested resources.

Forward Explicit Congestion Notification (FECN) bit. The user is notified that congestion-avoidance procedures should be initiated where applicable for traffic in the same direction as the received frame. FECN indicates that this frame, on this logical connection, has encountered congested resources.

In addition, a frame handler may use a Consolidated Link Layer Management (CLLM) message to notify the user that congestion-avoidance procedures should be initiated where applicable for traffic in the opposite direction of the received frame. It indicates that the frames the user

Distinct TCP/IP

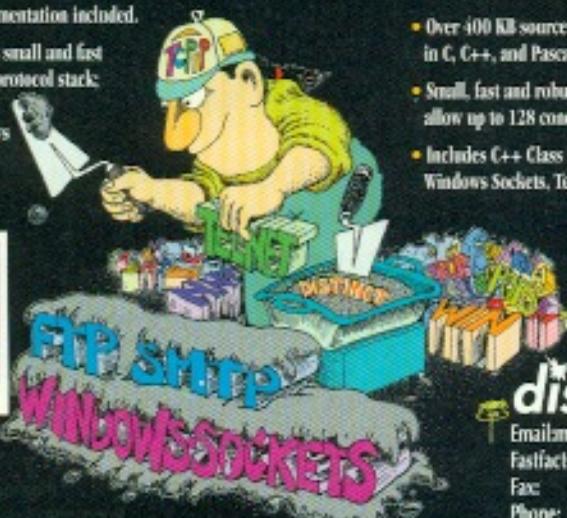
For C, C++, Visual C++, Turbo Pascal, Visual Basic and other Windows programmers.

SOFTWARE DEVELOPMENT 9015
Visual Edition

SOFTWARE DEVELOPMENT 9015
Professional Edition

SOFTWARE DEVELOPMENT 9015
Standard Edition

- TCP/IP Custom Controls for Windows Sockets, Telnet, FTP, RCP, R-commands (remote shell, rlogin and rexec), Mail (SMTP, POP2/POP3) and News (NNTP). For Visual Basic users only.
- Just define properties and handle events, no DLLs to call or TSR to use.
- 10 sample applications and over 200 pages of documentation included.
- Includes Distinct's small and fast Windows Sockets protocol stack; extensions run over other Windows Sockets compliant stacks.
- Includes all the features of the Standard Edition PLUS Telnet VT emulator, drag and drop FTP client and server, TFTP, LPR, LPD, transparent printing, Network Monitor, Finger and more.
- APIs for Windows Sockets, Berkeley Sockets, OMC RPC/ADR, Telnet, FTP, SMTP, POP2/POP3, NNTP and NetWin.
- Supports NDIS, ODI or Packet drivers, SLIP, CSLIP and PPP with scripting.
- Co-exists with Windows for Workgroups, Netware, Lan Manager, Vines, Lantastic and Pathworks.
- Over 400 KB source code samples in C, C++, and Pascal.
- Small, fast and robust 100% DLLs, allow up to 128 concurrent sessions.
- Includes C++ Class Libraries for Windows Sockets, Telnet and FTP.



distinct

Email: mk@distinct.com
 Fax: 408.366.2101
 Phone: 408.366.0133

CIRCLE NO. 125 ON READER SERVICE CARD

(continued from page 64)

transmits on a set of logical connections may encounter congested resources.

In all of these cases, the network only supplies the notification. The actual protocol for responding to the notification is supplied by layers above the frame-relaying bearer service. The standards define some suggested protocols.

Network Notification of Congestion

For the network to be able to detect and signal congestion, each frame handler must monitor its queuing behavior. If queue lengths begin to grow to a dangerous level, then either forward- or backward-explicit notification, or a combination, should be used to try to reduce the flow of frames through that frame handler. The choice of forward or backward may be determined at configuration time by whether the end users on a given logical connection are prepared to respond to one or the other of these notifications. In any case, the frame handler has some choice as to which logical connections should be alerted to congestion. If congestion is becoming serious, all logical connections through a frame handler might be notified. In the early stages of congestion, the frame handler might notify just those users generating the most traffic.

The following procedure for monitoring queue lengths is suggested in the standards: A cycle begins when the outgoing circuit goes from idle (queue empty) to busy (nonzero queue size, including the current frame). The average queue size over the previous and current cycles is calculated. If the average size exceeds a threshold value, then the circuit is in a state of incipient congestion, and the congestion-avoidance bits should be set on some or all logical connections using that circuit. By averaging over two cycles instead of just monitoring current queue length, the system avoids reacting to temporary surges that would not necessarily produce congestion.

The average queue length may be computed by determining the area (product of queue size and time interval) over the two cycles and dividing by the time of the two cycles. This algorithm is illustrated in Figure 2.

Forward Explicit Congestion Notification

The FECN bit is set to notify the receiving-end system that the marked frame has encountered congestion. In response, the receiving system should try to reduce the flow of data from the sending system on this frame-relay connection. The mecha-

nism for doing so must be above the level of the frame-relay bearer service, which provides no direct flow-control facilities.

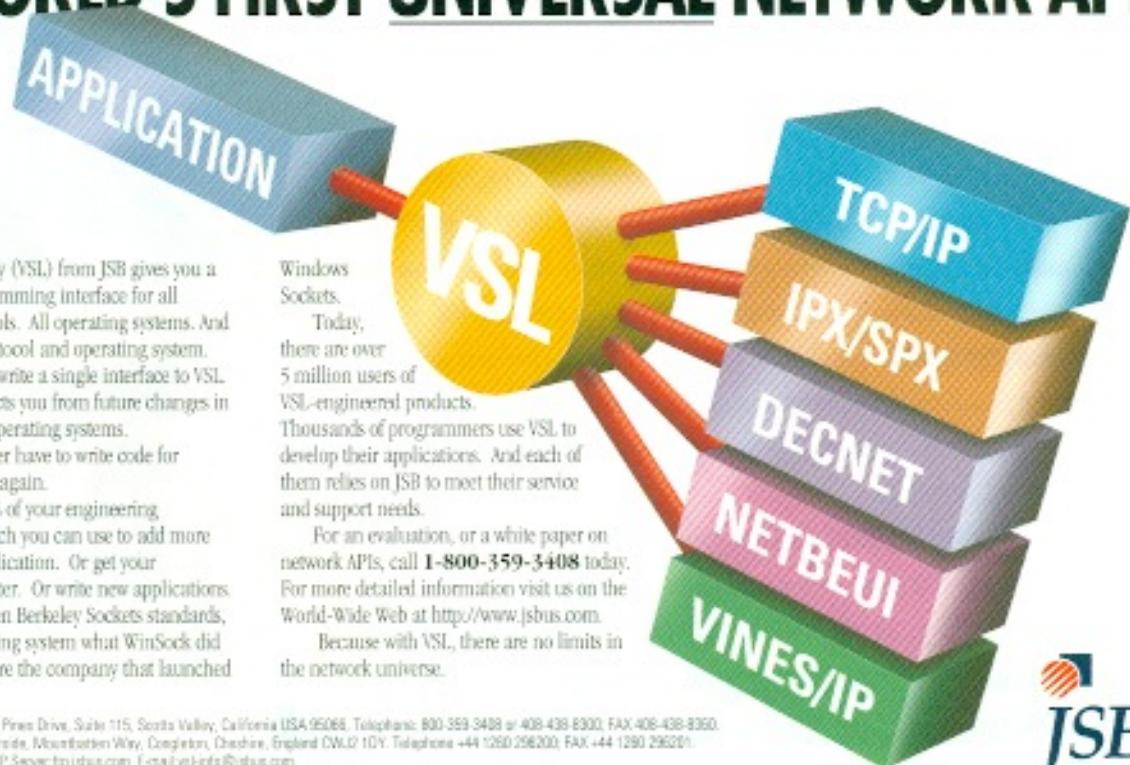
The receiving-end system should use this strategy for each connection:

1. Compute the fraction of frames for which the FECN bit is set over some measurement interval.
2. If more frames have the FECN bit set than have an FECN bit of zero, reduce the flow of frames from the sending system.
3. If the congestion condition persists, institute additional reductions.
4. When the congestion condition ends, gradually increase the flow of frames.

This strategy reacts slowly to congestion notifications for two reasons: First the end system does not react immediately to a particular FECN bit, but waits until the average behavior of the system over an interval indicates congestion. Second, the end system does not immediately reduce its outgoing flow, but rather signals its peers to reduce the incoming flow. All of this is consistent with a belief that congestion occurs slowly.

The details of the algorithm depend on whether the end system has actual control of the information rate from the source system or uses some sort of sliding-window,

INTRODUCING VIRTUAL SOCKET LIBRARY. THE WORLD'S FIRST UNIVERSAL NETWORK API.TM



The Virtual Socket Library (VSL) from JSB gives you a single application programming interface for all network transport protocols. All operating systems. And every combination of protocol and operating system.

All you have to do is write a single interface to VSL. That's it. VSL even protects you from future changes in network transports and operating systems.

In fact, you may never have to write code for proprietary network APIs again.

You'll save up to 30% of your engineering efforts—and budget—which you can use to add more functionality to your application. Or get your application to market faster. Or write new applications.

VSL is based on proven Berkeley Sockets standards, and does for every operating system what WinSock did for Windows. In fact, we're the company that launched

Windows Sockets.

Today, there are over 5 million users of VSL-engineered products. Thousands of programmers use VSL to develop their applications. And each of them relies on JSB to meet their service and support needs.

For an evaluation, or a white paper on network APIs, call **1-800-359-3408** today. For more detailed information visit us on the World-Wide Web at <http://www.jsbus.com>.

Because with VSL, there are no limits in the network universe.

(continued from page 66)

flow-control scheme. A rate-based system can provide a more precise control of information flow since it is based on the actual information rate in bits per second. Since frame relay does not require the use of fixed-size frames, a window-based system can provide only an approximate control over information rate. Such control is reasonably precise only if the statistical variance of the frame size is small.

Rate-based control. For rate-based con-

trol, it is assumed that a destination system has a means of regulating the data rate at the source-end system. Let us refer to current data rate as R . On each connection, the end system maintains two counters: FECN0 is the number of LAPP frames with FECN=0, and FECN1 is the number of LAPP frames with FECN=1. These counts are accumulated over a measurement interval δ_j^i . The standards suggest a value of δ_j^i approximately equal to four times the end-to-end transit delay.

1. Queue-length update: Beginning with $q_0 := 0$, if the i th event is an arrival event, $q_i := q_{i-1} + 1$; if the i th event is a departure event, $q_i := q_{i-1} - 1$.

2. Queue-area (integral) update:

$$\text{Area of the previous cycle} = \sum_{t_i \in (T_0, T_1)} q_{i-1} (t_i - t_{i-1}) \quad \text{Area of the current cycle} = \sum_{t_i \in (T_1, T_2)} q_{i-1} (t_i - t_{i-1})$$

3. Average queue-length update:

$$\text{Average queue length over the two cycles} = \frac{\text{Area of the two cycles}}{\text{Time of the two cycles}} = \frac{\text{Area of the two cycles}}{t - T_0}$$

Figure 2: Queue-length averaging algorithm. t = current time; t_i = time of i th arrival or departure event; q_i = number of frames in the system after the event; T_0 = time at the beginning of the previous cycle; and T_1 = time at the beginning of the current cycle.

FREE IMAGING SOFTWARE

The Fastest Imaging Software Available Anywhere



LEAD Maximum CMP, 51.1 kB
(compressed from original TIF, 5.63 MB)



JPEG Maximum, 71.1 kB
(compressed from original TIF, 5.63 MB)

Read • Write • Print • Dither (7 methods) • Display • Process • Convert (any bit depth, any file format) • Compress • Image Manipulation • Image File Loading • Zoom and Scroll • Twin Support • DIB and DDB Support and 24-bit DIB Driver to Allow Calls to Any GDI Drawing Function to a Bitmap • 22 Different Screen Point Effects • LEAD Windows Class (Overlapped, Popup, Child Window) • Access Part of Image • Image Processing: (Transpose, Mirror, Rotate, Resize, Crop, Average, Blur, Sharpen, Emphasize Edges, Line Detect, Convert Halftone, Histogram) • Unlimited Image Size

NEW: Visual Basic/Visual C++ Control with over 100 properties!

JPEG PCX GIF TGA BMP DIB WMF EPS
TIFF JTIF JFIF DDB LZW OS/2-BMP CCITT
CALS CMP WPG MAC IMG MSP PCD PCT RAS MORE!

DOS WINDOWS WINDOWS NT VISUAL BASIC PDF PRG CLIPPER C/C++

LEAD
TECHNOLOGIES
INCORPORATED

900 Boker St. Charlotte, NC 28204 704-332-5522 (Fax) 704-372-8161 Computer: "GO LEADTECH"

CIRCLE NO. 90 ON READER SERVICE CARD

If you are a programmer, call now to receive a free copy of LEAD's image compression and management utility, LEADTOOLS 4.0 Imagehandler™ for Windows. It's the only tool you'll ever need to handle images and a great way to evaluate the capabilities of our new LEADTOOLS 4.0 toolkit. (DOS version available.) LEADTOOLS will allow you to add any imaging function you can imagine to your application. Our compression, decompression and manipulation of color images is the fastest you can buy. LEADTOOLS 4.0 includes more functions and more example source code than any other developer toolkit. An Image Guarantee! LEADTOOLS 4.0 supports more file formats than any other toolkit: 47 implementations of the JPEG file interchange format and over 30 other file formats. Have an unsupported file format? Send it to us - we can make it work! New 24 bit DIB driver allows you to call all standard Windows GDI drawing functions directly to a hitmap. Kodak Photo CD, TWAIN support and much more!

FASTEST IMAGING TOOLKIT GUARANTEED!

LEADTOOLS compresses, decompresses, converts and processes raster images faster than any toolkit on the market. 30% to 50% faster! Compare LEADTOOLS with any of our competitors - RISK FREE - and see for yourself!

30 DAY RISK-FREE TRIAL!

What The Experts Say...

"We chose LEAD's (JPEG and BMP) compression technology over other available solutions for its image quality, ease of integration and speed performance."

"...if you need top-notch color support, consider LEAD's terrific 24-bit toolkit."

-Dr. Michael Covland, President and CEO, Corel Corp.

-Imaging Magazine, March, 1994

Join the growing family of happy LEAD customers:

• 3M • Compuq • Kodak • Polaroid • Sharp
• Adobe • Corel • Martin Marietta • Prolog • TRUVISION
• Canon • IBM • Olympus • Reuters • Xerox

Call for complete info and FREE demo disk!

800-637-1837

(continued from page 68)

terval, set $FECN0=FECN1=0$. At the end of each measurement interval, if $FECN1 \geq FECN0$, set $W = \lfloor \text{MAX}[0.875 \times W, 1] \rfloor$; if $FECN1 < FECN0$, set $W = \text{MIN}[W + 1, W_{\text{max}}]$.

If a connection has been idle for a long time, W should be set to 1 for that connection.

Backward Explicit Congestion Notification

Backward explicit congestion notification can be achieved with either the BECN bit in the LAPP address field or a consolidated link layer management (CLLM) message carried in a LAPP frame.

The BECN bit is set to notify the receiving system that the frames it transmits on this connection may encounter congestion. In response to this, the receiving system should reduce the flow of data transmitted on that connection.

The receiving-end system should use the following strategy for each connection:

1. When the first frame with the BECN bit set is received, reduce the information rate to CIR.
2. If additional consecutive frames with the BECN bit set are received, then institute additional reductions.
3. If a consecutive sequence of frames with the BECN bit set to zero are received, then gradually increase the flow of frames.

This strategy reacts rapidly to congestion notifications because, the end system immediately reacts to a single BECN bit and reduces its outgoing flow rather than signaling its peers to reduce the incoming flow. This reflects a belief that congestion occurs quickly.

As with the response to forward-explicit congestion notification, the details of the algorithm depend on whether control is rate based or window based.

Rate-based control. The standards define a step count S that is used to determine when the transmitter may increase or decrease its rate. The algorithm is as

- (a) Set $W = \text{MAX}[W-1, W_{\text{min}}]$
 (b) Set $W = W_{\text{min}}$
 (c) Set $W = \text{MAX}[aW, W_{\text{min}}]$, where $0 < a < 1$

Example 1: Three classes of adaptive window schemes.

- (a) Set $W = \text{MIN}[W+1, W_{\text{max}}]$ after N consecutive successful transmissions.
 (b) Set $W = \text{MIN}[W+1, W_{\text{max}}]$ after W consecutive successful transmissions.

Example 2: Successful transmissions may indicate that the congestion has decreased and window size should be increased.

follows: Initially, set $IR=CIR$ or less in the transmit direction. Then:

1. If a frame with BECN set to 1 is received, and the user's offered rate, R , is greater than CIR, then reduce the offered rate to CIR.
2. If S consecutive frames are subsequently received with the BECN bit set to 1, the user should reduce its rate to the next lower "step." Further rate reductions should not occur until an additional S consecutive frames are received with the BECN bit set. The step rates are $R=0.675 \times CIR$, $R=0.5 \times CIR$, and $R=0.25 \times CIR$.
3. After the user has reduced its rate due to receipt of BECN signals, it may increase its rate by a factor of 0.125 after any $S/2$ consecutive frames are received with the BECN bit clear; that is, $R=1.125 \times R$.

If a connection has been idle for a long time, then R should be set to CIR for that connection.

Window-based control. Here, the step count S is defined to be the interval during which one frame is transmitted and acknowledged. The algorithm is as follows: Initially, set the window size W to some small value such as 1 or $0.5 \times \text{last window size}$. Then:

1. If a frame with BECN set to 1 is received, then set $W = \lfloor \text{MAX}[0.625 \times W, 1] \rfloor$.
2. If S consecutive frames are subsequently received with the BECN bit set to 1, the user should repeat the reduction.
3. After the user has reduced its window size due to receipt of BECN signals, it may increase its window size by one after any $S/2$ consecutive frames are received with the BECN bit clear; that is, $W = \text{MIN}[W+1, W_{\text{max}}]$.

If a connection has been idle for a long time, then W should be set to its initial value for that connection.

Consolidated Link Layer Management (CLLM). CLLM is a variation of backward explicit congestion notification that uses a message rather than the BECN bit to signal congestion. The CLLM technique can be used when congestion occurs at a network node, but no reverse traffic is available to carry the BECN indication. CLLM messages carry a list of congested DLCLs to reduce the traffic load on the network.

The CLLM is a message carried in the information XID LAPP frame. The body of the XID frame lists the frame-relay connections that are congested and identifies the cause. The following cause values are recognized:

- Network congestion due to excessive traffic.
- Facility or equipment failure.
- Maintenance action.
- Unknown.

If these conditions are short term, they are anticipated to have a duration of seconds or minutes; otherwise the designation is long term.

Implicit Congestion Control

Implicit signaling occurs when the network discards a frame, and this fact is detected by the end user at a higher, end-to-end layer. When this occurs, the end-user software may deduce that congestion exists.

For example, in a data-link control protocol that uses a sliding window flow and error-control technique, the protocol detects the loss of an information frame in one of two ways. When a frame is dropped by the network, the following frame will generate an REJ frame from the receiving end point. When a frame is dropped by the network, no acknowledgment is returned from the other end. Eventually, the source end will time out and transmit a command with the P bit set to 1. The subsequent response with the F bit set to 1 should indicate that the receive sequence number NR from the other side is less than the current send sequence number.

Once congestion is detected, the protocol uses flow control to recover from the congestion. Assume that the layer-two window size, W , can vary between the parameters W_{min} and W_{max} , and is initially set to W_{max} . We would like to reduce W as congestion increases to gradually throttle the transmission of frames. Example 1 lists three classes of adaptive window schemes based on response to one of the two aforementioned conditions.

Successful transmissions (measured by receipt of acknowledgments) may indicate that the congestion has decreased and window size should be increased. Example 2 shows two possible approaches. Studies suggest that the use of the strategy in Example 1(c) with $a=0.5$ plus the strategy in Example 2(b) provides good performance over a wide range of network parameters and traffic patterns. This is the strategy recommended in the standards.

DDJ

To view for your favorite article, circle inquiry no. 7.