Congestion Control in Interconnected LANs

Mario Gerla
Leonard Kleinrock

Introduction

Local area networks (LANs) are routinely interconnected with bridges and routers to achieve one or more of the following benefits:

1.) geographical extension of LAN coverage,
2.) better fault isolation and containment,
3.) security and access control,
4.) heterogeneous LAN interconnection,
5.) throughput efficiency improvement.

As usual, all of these attractive features do not come without cost. One potential problem is congestion of LANs, bridges, and routers in the interconnection chain. Congestion in network interconnection is a well-known problem. But the situation is particularly challenging here for the following reasons:

1.) In the caternet there may be a dramatic channel bandwidth mismatch between links [e.g., 10-Mbps Ethernets, 100-Mbps fiber distributed data interface (FDDI), 56-kbps wide area network (WANs), etc.].
2.) Even if the LANs are of the same type and have the same bandwidth (which is usually the case when interconnection is via bridges), the load may be unbalanced. Thus, congestion occurs at the boundaries. Media-access-control (MAC) level bridges are ill equipped to prevent congestion. In fact, the MAC level can very effectively control congestion in a single LAN [e.g., round-robin access in token rings and buses, binary back off in CSMA-CD (carrier sense multiple access with collision detection) schemes, etc.]; but, it cannot extend this control across LANs.
3.) State-of-the-art bridges can forward only a few thousand packets per second, at most. Routers process even less, because of the higher protocol overhead. This is in sharp contrast with the very high LAN speeds (e.g., 100 Mbps in FDDI). Thus, in a LAN interconnection, the bridge or router packet processing rate is one or two orders of magnitude lower than the LAN speed. This is just the opposite of what happens in WAN interconnections, where gateway throughputs generally exceed network throughputs. It is clear then that bridges and routers, far from relieving congestion caused by LAN speed or load mismatch, may themselves be the prime cause of congestion since they are the bottlenecks on the path.

The preceding point shows that congestion control in LAN interconnection is much more complex than in conventional packet-switched environments. One often-proposed remedy is to rely on upper-layer (transport and above) flow control. This approach, however, may prove costly (performance wise), unfair, or simply infeasible for certain types of traffic (e.g., real-time animated graphics) that cannot be delivered in a flow-controlled, stop-and-go mode.

Some form of congestion control must be implemented in bridges and routers. This means monitoring the loads in adjacent LANs and taking actions on transit traffic as well as on traffic sources, so that the adverse effects of congestion (i.e., unacceptable delays, degraded throughput, deadlocks, etc.) are avoided; and efficient and fair performance is maintained. In the following, we review the typical functions and characteristics of bridges and routers, identify the types of traffic requirements generally offered to a LAN, and discuss various congestion control mechanisms that could be made to work in this environment.

Bridges and Routers

Bridges

The main attribute of bridges is transparency. They operate at the MAC level [sometimes at the logical link control (LLC) level] and do not interfere with workstation protocol layers (i.e., network and above). Bridges interconnect LANs with a uniform address domain (therefore, no address conversion is required); in the majority of cases, they connect homogeneous LANs (i.e., same MAC protocols).

The simplest type of bridge is the full-broadcast bridge, which relays whatever it receives from one LAN to all the other connected LANs. This implementation, however, does not provide any throughput improvements over the repeater connection. A more common and effective bridge implementation is the "filtering" bridge, which can selectively forward the packet based on its destination address. Essentially, the address is checked against a table of addresses; the packet is forwarded if a match is found. The table may be preloaded at system initialization; or it may be

0890-8044/88/0001-0072$01.00©1988 IEEE

"learned" dynamically by the bridges [1,2]. The latter solution is more robust to failures and more flexible in that it can handle station relocation; but it requires a tree topology to avoid loops. This obstacle can be overcome by defining a virtual spanning tree over the initial mesh topology using a distributed spanning tree algorithm run by the bridges [2].

Another approach to selective forwarding is source routing [3]. In this scheme, the source node issues a scouting packet, which is broadcast through the network to the destination. The scouting packet picks up the addresses of the intermediate bridges along the path; upon reaching the destination, it is returned to the source along the same path (stamped in the header). The source examines the various scouting packets returned by the destination (one for each alternate path) and selects the most effective path. The path is then stamped in the header of all subsequent packets and is used to drive the packets to the destination (source routing).

Routers

As opposed to the bridge, the router is not transparent to user protocols. It implements the network protocol layer (e.g., DOD IP, DECNET, XNS, etc.) and, thus, has peer counterparts in other routers as well as in user workstations and hosts. The router terminates the MAC and the LLC layers of each connected LAN and permits translation between different address domains. Because of the higher protocol processing overhead, the router generally delivers lower throughput than the bridge. It provides, however, more efficient routing and flow control than a bridge, since it operates at the network level and can exploit the traffic management procedures that are a part of that layer.

"Brouters"

In comparing bridges with routers, we find that bridges offer the advantage of transparency but suffer of the limitation of poor traffic control and management. Very recently, proposals have been made to span the gap between bridges and routers with an intermediate-class device (by some called the "brouter"), which should incorporate many of the traffic control features of the router yet retaining the transparency of the bridge. Some of the proposed enhancements are: LLC-level processing (that is, the bridge can terminate and manage LLC connections) [1]; interconnection of different MAC protocols [4]; implementation of distributed, minimum delay routing algorithms (which remove the spanning tree restriction); and flow and congestion control. The price one pays for these enhancements is a reduction in processing capacity (packets per second), due to the additional work performed on each packet.

Traffic Requirements

Before discussing possible flow and congestion control solutions, we review the types of traffic requirements that can be found in a LAN, the demands they pose on the resources, and the form of flow control to which they can be subjected to.

Interactive traffic (e.g., word processing, inquiry response, remote computing, etc.) will typically be only a small fraction of the total LAN traffic in spite of the fact that it may actually be volume, the largest application in terms of connect time. Interactive sessions require low response time and involve relatively low traffic rates. Thus, flow control should preferably not be applied to interactive traffic since the saving in network resources (bandwidth, buffers, etc.) does not make up for the risk of violating the user delay constraints.

File transfers tend to use a large portion of LAN bandwidth. For very large files, the user may tolerate delays on the order of minutes (instead of seconds, as for interactive applications). The issue of sharing LAN bandwidth efficiently and fairly among several simultaneous file transfers is important. Connection-oriented transport is desirable, and backpressure-type flow control should be applied along the path, to avoid congestion.

In certain environments, real-time data transfers (e.g., remotely refreshed graphic animation, radar data, etc.) may also amount to a significant portion of LAN bandwidth. Different from file transfers, real-time traffic cannot be flow controlled once the session is established. Rather, one must make sure that there is enough bandwidth prior to call establishment. Likewise, bandwidth reservation is necessary for real-time voice and video requirements.

Flow and Congestion Control Techniques

We now proceed to review the flow and congestion control mechanisms that can be used in an interconnected LAN environment. Some of these mechanisms are actually implemented in existing networks; others are just proposals inspired by similar schemes used in conventional packet networks. As we shall see, some of the mechanisms can be implemented in bridges; others require more sophistication and/or access to higher protocols, thus are more appropriate for routers/brouters.

In this context, the term "flow control" refers to the regulation of the traffic flowing on an individual connection; thus, the flow control procedure is run in the source and destination host, with the possible participation of intermediate nodes along the path. The main goal of flow control is to prevent overflow of the buffers dedicated to the connection. On the other hand, congestion control refers to a more global procedure carried out by internal network nodes (in our case, bridges and routers) to prevent network congestion; the controlling action may be exercised on many source/destination pairs indiscriminately and simultaneously.

In this paper, the focus is on congestion control (that is, prevention of internal congestion); however some of the proposed schemes require the interaction of flow and congestion control.

Dropping Packets

Dropping packets when buffers are full is currently the most popular and expedient way to relieve congestion in bridges and routers. This approach is consistent with the "best effort" delivery philosophy in datagram networks. The task of retransmitting packets is then delegated to the LLC level (in bridges) or to the transport level (in routers). Indiscriminate dropping of packets, however, is often counterproductive because it triggers end-to-end retransmissions (thus, it does not reduce the offered load)

[5]; and it allows the near-congestion situation (heavy queues and high end-to-end delays) to persist. Furthermore, the high packet drop rate and high delay may render the system unsuitable for real-time traffic.

Another drawback of packet dropping is lack of fairness: connections with fewer network hops (in the limit, connections within the same LAN) lose fewer packets and thus get better performance [5]. Similarly, sources with the shortest retransmission time-out have a better chance to "get in" the bridge (or router). Thus, the users who push harder get more throughput. This creates unfairness and also amplifies congestion.

Input Buffer Limit

A congestion control technique often employed in datagram, connectionless networks is the input buffer limit scheme [6]. A limit is set on the maximum number of input packets (i.e., packets from local hosts) that can be buffered in the packet switch. When the limit is exceeded, input packets are dropped. Since there is no limit on transit packets (i.e., packets from remote hosts), the method favors transit traffic over local traffic. The rationale is that transit packets are more valuable in that the network has already invested some of its resources in them; furthermore, dropping a transit packet triggers remote retransmissions and causes more traffic in the network, while the retransmission of a local packet does not impact internet traffic.

The input buffer limit scheme may be applied also to the bridge and router environment to make packet dropping more selective and less harmful to overall performance. In this setting, the local packets are the packets originating from the LANs directly connected to the bridge or router. The remote packets are packets coming from networks two or more network hops away. If nodal buffers become congested (either because of nodal processor congestion or overload in one of the intervening networks), it is clearly preferable to drop local packets rather than remote packets, since the former will cause retransmissions in the local network only.

The input buffer limit scheme is fairly straightforward to implement in routers, since a router can tell from the source internet address whether a packet is local or remote. In transparent spanning tree bridges, this method is not applicable since the address space is uniform across all interconnected nets and the bridge cannot distinguish between local and remote addresses. In source routing bridges, on the other hand, the bridge reads the route from the packet header; thus, it knows its exact position along the path. In particular, if the bridge address is in the first position, the packet is a local packet. The input buffer limit scheme can, therefore, be applied.

"Choke" Packets

The problem with indiscriminate packet dropping or even local packet dropping is that it does not provide a direct feedback to the traffic sources. Ideally, when a network resource becomes congested, we would like to slow down the sources that feed traffic to it. A method that can be used in connectionless networks is the "choke packet": namely, whenever a bridge or router experiences congestion (e.g., it notices that one of the adjacent LANs is heavily loaded), it returns to the source a choke packet containing the header of the packet traveling in the congested direction (the information packet itself may be forwarded if that is still possible; otherwise, it is dropped) [6,7]. The source, upon receiving the choke packet, declares the destination (which it reads from the header) congested, and slows (or stops altogether, for a period of time) traffic to that destination.

The "choke packet" scheme bears resemblance to the "source quench" mechanism in TCP/IP. However, while the source quench is available only to routers, the choke packet can be issued by bridges as well. In fact, this mechanism is probably more crucial for bridges than for routers. In general, the routers run a dynamic routing algorithm that is sensitive to loads and, therefore, can eventually report a congested situation to the hosts. Bridges, in their simplest version, are not equipped with routing algorithms; thus, the choke packet may prove the easiest way to reflect congestion information from bridge to source.

Backpressure

In a conventional packet network (e.g., X.25) backpressure is the regulation of flow along a virtual connection [6]. This method is available only in "virtual-circuit"-type networks, where the network layer protocol is implemented with a virtual circuit. Intermediate nodes along the virtual circuit can throttle the flow by closing the window of permissible outstanding packets. If a link along the path becomes congested, the node upstream of it will close the window, i.e., it will not return credits to its predecessor. This starts a chain reaction, which eventually cuts off the supply of new packets to the virtual circuit. When the congestion clears, credits are returned and the flow is resumed.

Backpressure is much more effective than the choke packet because it permits a smoother regulation of the flow (the window can be progressively reduced before being closed) in a selective and fair manner. However, it requires a virtual circuit implementation, and the participation of all intermediate nodes in the management of the virtual circuit.

In a router, backpressure can be implemented in two ways. The direct way is to use a virtual-circuit-type protocol for the internet level (e.g., X.75). One must admit, however, that the most popular router implementations (XNS, TCP/IP, and DECNET) are all datagram oriented. The "indirect" way is to use connection-oriented LLCs in the underlying LANs, and to concatenate and synchronize the LLC windows across the repeaters (i.e., the repeater passes a window credit to the upstream LLC only after it receives one from the downstream LLC). Some tricky issues arise when multiple paths exist between the source and destination, since the datagram routers may route different packets of the same session on different paths. Thus, the router may be required to distribute credits among several upstream LLC connections within the same session (perhaps using a round-robin policy). We observe that this is the same problem as that of interconnecting X.25 networks with IP gateways.

In a conventional bridge, the LLC header is passed through transparently; thus, backpressure flow control is not available. However, in an LLC bridge (i.e., a bridge that terminates the LLC protocol), backpressure can be exercised on the concatenation of the LLC segments, as in the

case of routers. One advantage here is the fact that bridges typically forward packets on a fixed path; thus, the bridge need not be concerned with managing the return of credits on multiple paths.

An alternative method for backpressure with transparent bridges was proposed in [5]. The source-destination LLC connection (which in that application is assumed to pass transparently through the bridges) is operated with a dynamic window. When a frame is dropped at the bridges (because of congestion), the destination returns a REJECT frame to the source. The source then sets the window to 1 and retransmits the frame. The window is gradually expanded as more frames are successfully transmitted and acknowledged. Simulation shows that the dynamic window adjustment can yield significant improvements over the fixed window scheme. One problem with this approach, however, is the fact that a dynamic window is not a part of the LLC standard.

From the preceding discussion, it is clear that backpressure can be applied only to connection-oriented (as opposed to connectionless) applications. Referring to the traffic models presented in the Traffic Requirements section, we would argue that the connection-oriented mode (and, therefore, backpressure flow control) is most appropriate for file transfers. A file transfer, in fact, can tolerate being slowed down and even stopped (and later resumed) as long as the total transfer time is within given constraints.

Real-time traffic should be transmitted in a connectionless mode because of the tight delay constraint that exists on the delivery of each packet. The connection-oriented mode is undesirable since it would increase delay and processing time at intermediate bridges. Furthermore, the major benefits of the connection-oriented mode (error detection and retransmission) cannot be enjoyed anyway by real-time traffic because retransmissions would likely violate the delay constraint. Likewise, backpressure flow control would be infeasible because it would introduce unacceptable delays.

For interactive traffic, the use of the connection-oriented mode would be justified mainly for error recovery. However, given that error rates are extremely low in LANs, this reason alone is not a very strong one. As for congestion protection, the relative volume of the interactive traffic is generally a small fraction of the total LAN traffic; thus, there is not much to be gained by exercising backpressure on this traffic. On the contrary, backpressure may cause undesirable delays for the user.

Congestion Prevention

In the previous section, we have stated that real-time traffic cannot be effectively flow controlled using backpressure. For similar reasons, packet dropping and choke packets are not applicable either. Yet, real-time applications (voice, video, animated graphics, etc.) may end up representing a large portion of the traffic in future LANs; thus, the ability to flow control this type of traffic will be of critical importance.

Probably, for real-time traffic, the only effective way to avoid congestion is to prevent it. That is, a voice or video connection should be accepted only if there is enough "bandwidth" in the network to support it. Available bandwidth would be measured, of course, in a statistical, rather than deterministic, sense (the deterministic measure being feasible only in time-division-switched schemes). Methods for bandwidth control in packet networks have been proposed in the literature [8]. These methods could be adapted to the bridge and router environment.

The basic idea is for each node (bridge or router) to measure the available bandwidth in the LANs to which it is connected. A distributed algorithm (which requires the periodic exchange of bandwidth measurements among neighboring nodes) permits each node to estimate the shortest path to each destination LAN as well as the bandwidth on such a path. One can actually show that, with some extra effort, the set of all possible paths ranked by increasing length and bandwidth can be generated [8]. This information is passed on by each node (bridge or router) to each workstation or the connected LAN. From this information, the workstation can then compute the bandwidth available to each destination and the next node on the path to such a destination.

Congestion prevention is accomplished at connection setup time. Upon receiving a real-time connection request from the user (or process), the workstation examines (or estimates) the bandwidth requirement of the connection and can determine whether to accept or reject it on the basis of the information contained in the bandwidth and routing tables.

The preceding "bandwidth control" procedure can be implemented in routers at the network level. It can also be implemented in "brouters" able to communicate with each other and to maintain routing and bandwidth tables.

For simple bridges, the bandwidth control algorithm may be too complex to run because of interbridge communications and internal processing requirements. As an alternative, one may propose to use a modified version of source routing, where each "scouting" packet carries not only the trace of the path in its header, but also the bandwidth available on the path. Namely, each intermediate bridge stamps in the header of the scouting packet its address and the value of residual bandwidth available on the adjacent LAN. The scouting packets are returned to the source, which then selects a feasible path (if any).

One should point out that, in future, high-speed LANs the nodes (bridges and routers) are more likely to be the bottlenecks than the channels. Thus, the bandwidth computation should take into account the bandwidth available in the nodal processor. It can be easily seen that both bandwidth control and source routing algorithms can be modified to include nodal bandwidth.

Conclusion

Congestion control in interconnected local area networks (LANs) poses more challenging problems than in conventional packet nets because of the following reasons: speed mismatch between LANs; processing limitations of bridges and routers; functional limitations of the bridge protocols; and emergence of real-time traffic (voice, video, graphics, etc.) as a key application in local networking environments.

The current practice of using "best effort" delivery and dropping packets when the bridge or router are congested is not adequate and should be used only as a last resort. We advocate the use of combined flow and congestion control for file transfers (through choke packets or backpressure);

and the use of congestion prevention (via bandwidth control or enhanced source routing) for real-time traffic.

The above-mentioned flow and congestion control mechanisms require some modifications and enhancements of the bridge design (e.g., interbridge control message exchange, logical link control protocol support, dynamic routing algorithm support, etc.). The main challenge will be to retain the transparency and throughput efficiency of the bridges while introducing these additional features.

Acknowledgments

This research was supported by DARPA MDA 903-87-C-0063, NSF-INT-85-16798, NSF-INT-85-14377, and Pacific Bell and the State of California (through a MICRO grant).

References

[1] Seifert, W. M., "Bridges and Routers," *IEEE Network,* this issue.

[2] Backes, F., "Transparent Bridges for Interconnection of IEEE 802 LANs," *IEEE Network,* this issue.

[3] Pitt, D. A., and Dixon, R., "Addressing, Bridging, and Routing," *IEEE Network,* this issue.

[4] Hart, J., "Extending the IEEE 802.1 Standard to Remote Bridges," *IEEE Network,* this issue.

[5] Bux, W., and Grillo, D., "Flow Control in Local Area Networks of Interconnected Bridges," *IEEE Trans. Commun.,* Vol. COM-33, No. 10, Oct. 1985.

[6] Gerla, M., and Kleinrock, L., "Flow Control Protocols," in *Computer Network Protocols and Architectures,* Edited by P. Green, Plenum, 1982.

[7] Majithia, J. C., et al., "Experiments in Congestion Techniques," *Proc. Int. Symp. Flow Control Comp. Nets,* Versailles, France, Feb. 1979.

[8] Gerla, M., "Routing and Flow Control in ISDN's," *ICCC '86 Proc.,* Munich, Sept. 1986.

Mario Gerla is a Professor of Computer Science at UCLA. He received a graduate degree in Electrical Engineering from Politecnico di Milano, Italy in 1966 and a Ph.D. in Computer Science from UCLA in 1973.

His research interests include analysis, design and control of distributed communications networks and systems; design and performance evaluation of data communications processors; algorithms for distributed computation; computer network protocol evaluation; modeling and evaluation of distributed operating systems; design and evaluation of high-speed fiber optics network architectures.

Dr. Gerla consults for industry and government in the areas of network planning, design and performance evaluation.

He has published more than 60 technical papers on various subjects including network optimization, network performance evaluation, routing, flow control. He has contributed chapters to several books.

Leonard Kleinrock is a professor of Computer Science at the University of California, Los Angeles. He received his B.S. degree in Electrical Engineering from the City College of New York in 1957 (evening session) and his M.S.E.E. and Ph.D.E.E. degrees from the Massachusetts Institute of Technology in 1959 and 1963, respectively. While at M.I.T., he worked at the Research Laboratory for Electronics, as well as with the computer research group of Lincoln Laboratory in advanced technology. He joined the faculty at U.C.L.A. in 1963. His research interests focus on local area networks, computer networks, performance evaluation and distributed systems. He has had over 150 papers published and is the author of five books—*Communication Nets; Stochastic Message Flow and Delay, 1964; Queueing Systems, Volume I: Theory, 1975; Queueing Systems, Volume II: Computer Applications, 1976; Solutions Manual for Queueing Systems, Volume I, 1982,* and, most recently, *Solutions Manual for Queueing Systems, Volume II, 1986.* Dr. Kleinrock is co-director of the U.C.L.A. Computer Science Department Center for Experimental Computer Science and is a well-known lecturer in the computer industry. He is the principal investigator for the DARPA Advanced Teleprocessing Systems contract at U.C.L.A. Dr. Kleinrock is a member of the National Academy of Engineering, is a Guggenheim Fellow, an IEEE Fellow, a member of the IBM Science Advisory Committee, and in 1986, he became a member of the Computer Science and Technology Board of the National Research Council. He has received numerous best paper and teaching awards, including the ICC 1978 Prize Winning Paper Award, the 1976 Lanchester Prize for outstanding work in Operations Research, and the Communications Society 1975 Leonard G. Abraham Prize Paper Award. In 1982, as well as having been selected to receive the C.C.N.Y. Townsend Harris Medal, he was co-winner of the L. M. Ericsson Prize, presented by His Majesty King Carl Gustaf of Sweden, for his outstanding contribution in packet switching technology. In July of 1986, Dr. Kleinrock received the 12th Marconi International Fellowship Award, presented by His Royal Highness Prince Albert, brother of King Baudoin of Belgium, for his pioneering work in the field of computer networks. Dr. Kleinrock is also founder and CEO of Technology Transfer Institute, a computer/communications seminar and consulting organization located in Santa Monica, California.