High Performance Interconnection Between High Data Rate Networks

by

E.C. Foudriat, K. Maly, C.M. Overstreet, L. Zhang, W. Sun
Computer Science Department, Old Dominion University
Norfolk, VA 23529

Presented at the International Workshop on Advanced Communications and Applications for High Speed Networks
March 16 - 19, 1992
Munich, Germany

Abstract

The paper discusses the bridge/gateway system needed to interconnect a wide range of computer networks to support a wide range of user quality-of-service requirements. The bridge/gateway must handle a wide range of message types including synchronous and asynchronous traffic, large, bursty messages, short, self-contained messages, time critical messages, etc. The paper shows that messages can be classified into three basic classes, synchronous and large and small asynchronous messages. The first two require call setup so that packet identification, buffer handling, etc. can be supported in the bridge/gateway. Identification enalbes resequencing of messages at the bridge/gateway which supports interconnection between networks having large differences in packet size. The third class is for messages which do not require call setup. Resequencing hardware is presented in the paper based to handle two types of resequencing problems. The first is for virtual parallel circuitwhich can scramble channel bytes. The secondsystem is effective in handling both synchronous and asynchronous traffic between networks with highly differing packet sizes and data rates. The two other major needs for the bridge/gateway are congestion and error control. The paper presents a new dynamic, lossless congestion control scheme which can easily support effective error correction. Results indicate that the congestion control scheme provide close to optimal capacity under congested conditions. Under conditions where error may develop due to intervening networks which are not lossless, intermediate error recovery and correction takes 1/3 less time than equivalent end-to-end error correction under similar conditions.

I. Introduction

Network systems require support for interconnection between networks and to users since they span widely different environments and must provide a variety of Quality-of-Service (QOS) features. This is especially true now that gigabit networks and multimedia environments [13] are becoming a reality. This paper investigates systems required to interconnecting high data rate networks. Special attention is directed toward providing support for a wide variety of applications which are implied by multimedia applications.

Present network interfacing supports service between connectionless networks usually via bridges or gateways [1]. Tthe major interconnection problems has been routing. Gateways, such as in Internet, provide a compatible protocol at the network layer which forwards packets to toward their destination on the selected "best route" and, where necessary, breaks up the packet to conform to any lower layer packet length restrictions [1, 2]. Thus, all stations must implement the same network layer protocol.

To over come the problem of common network layer, especially where not needed the transparent bridge approach is used. Bridges generally implement some form of spanning tree algorithm which eventually gets the packet to its destination but some times via a "non-shortest" route [1, 2, 3, 4]. Packet reformatting in bridges is minimal at best.

Typical interconnection between networks include those which support X.25, SNA, DECNet, XNS. etc., and span the data rate range from modems to LANs. DARPA's Internet [2] is a packet switched WAN network implemented over leased telephone and satellite links but its protocols have been used in LANs also.

Interconnecting systems for high data rate networks, that is, systems directed toward the implementation of Broadband ISDN (BISDN) also consider mainly connectionless protocol support. One system considers the use of DQDB for MAN interconnections [5]. Another supports the OSI connectionless network protocol (CLNP) in a ISDN environment [6]. Still another, frame-relay provides interconnection support for ISDN but does so by supporting critical functions at the end points only [7]. Other forms of network interconnections include the HIPPI-to-ATM HAS interconnection scheme for Nectar [8], and direct ATM-to-host for workstation coupling [9, 10] but,

[1]Research support has ben provided by Sun Microsystems, RF539044, NASA. Langley Research Center grant NAG-1-908. and Virginia Center for Innovative Technology grant INF-89-002-01

for the most part, these are point-to-point connections which do not span a range of different networks. Others have proposed transferring most of the protocol operations to the application level [11] but this means that the each host will have to implement the QOS needs for the user.

While there is no question that BISDN and its related systems provide capable, flexible network technology, they do not appear to be suitable for the wide range of network interconnections which are possible and which certainly might be considered desirable. First, they do not consider connection to vastly different formats or data rates, such as Ethernet and most certainly not to twisted pair Ethernet systems which are so common in personal computer network systems. A recent paper [12] does consider an ATM - FDDI gateway but provides only minimal user support. Second, while BISDN networks are suppose to support a wide range of services, it is not clear that all services will be provided by the interconnecting network itself as noted for frame relays [7] and discussions of ATM systems [14]. Third, BISDN packet and protocol header size lead to considerable bandwidth inefficiency for connecting networks which do not maintain its format but must incorporate its packets [12]. Further, many network problems such as routing are considerably different when "on-premise" networks are included. Finally, in some common carrier network situations, it may be reasonable to provide improved performance and/or lower cost by outright leasing of bandwidth and using network interface systems provided by the customer [15]. Many of the above problems and situations are best handled by employing bridge/gateway interfacing to common carrier high data rate networks and between private systems.

In this paper, we discuss new features which, when incorporated into bridge/gateway systems, provide significant improvement in overall network connectivity. In the next section, the requirements bridge/gateways are presented and QOS factors discussed. These lead to bridge/gateway features, presented in Section 3. Features include call handling, message classification and packet identification to support QOS requirements. Next, receiver handling which provides resequencing and restructuring systems, and sender handling which includes a new lossless congestion and error control system are discussed. Performance information about both handlers is presented. The last subsection presents information on intermediate error correction available in the receiver and sender handling systems. It substantially reduces error correction times.

II. Bridge/Gateway Requirements

Bridge/gateway requirements include interfacing to:

1. a wide range of data rates from submegabit/sec. to gigabit/sec.;
2. a wide range of protocol and packet structures;
3. a wide range of connectivity and topology structures; and
4. support for a wide range of quality of service requirements which can include, synchronous, asynchronous, and bursty traffic, low latency, acceptable error to error free and reliable service, etc.

Each of these factors influence the bridge/gateway structures which are needed. For example, a typical scenario might be:

> Three people cooperate to develop an architectural drawing including layouts, 3D moving scenes, correlated voice commentary, etc. Each person has a different graphics configuration and connectivity through a different network system. The information is developed by one person, shipped to the others and then they participate both on-line and off-line in changes. Finally, the finished product is shipped to a fourth party again through a different network and displayed as an advertisement but not copied. The original and transfer files are compressed resulting in traffic conditions which include bursty, error free, alternatively high and low data rates, multicast, long periods where no data is being exchanged, etc. When shipped to the fourth party, delay shift for the components is critical.

In this situation, a direct network to host connection would require each cooperating party to have multiple network interfaces and a considerable variety of software to handle the differing network interconnections efficiently. Would it not be better to connect each host to a flexible bridge/gateway system which could effectively support a large range of interfaces with tailored quality of service requirements for a large group of clients?

III. Bridge/gateway Features

To support the diverse bridge/gateway requirements, we need to isolate the bridge/gateway operational features. The critical feature are:

1. call handling - especially service requirements and resource allocation;
2. receiving - especially resequencing for the next network segment;
3. sending - formatting and controlling flow for minimal loss; and
4. error detection and correction which involves both sending and receiving.

Figure 1 illustrates the general logic structure for a bridge/gateway interconnection system. On the receiver

side, the first set of blocks are those needed to handle the access protocol, that is, decoding the bit pattern, maintaining clock synchronization and decoding header and trailer information. This latter decoding provides information including message type, destination, etc. The basic packet is stored in a shift register while decoding and checking are accomplished so that the packet may be forwarded in the event that its destination is for another node.

The shaded blocks in Figure 1 are those devoted to special bridge/gateway operations. Here the packet header information determines the message identity and places or links packets in memory in order to properly resequence packets. In a next section, operations needed for handling specific message classes will be discussed. New packets are then prepared for transmittal to the outgoing network.

A. Call Handling

Although this paper is mainly directed toward the hardware and lower layer software to control network interfacing, certain aspects of call handling need to be discussed. It is assumed that call handling will be supported mainly in software, so its features are reasonably flexible to suit specific needs.

Call handling for multimedia operations is considerably more complex than reserving channels in frames or routing packets at immediate nodes toward their destination. It is expected that call handling will require a dialogue between the caller and the network system where the user specifies and even may negotiate with the network over services which are available. The elements of the dialog will include the QOS requirements which the user deems to be critical and the ability of the intervening network to provide such service and the expected cost. To further complicate the call handling operations, the QOS requirements may be transient, i.e., different QOS requirements may exist over different portions of the total call. While it would be nice to assume that the differing QOS requirements would be known at call time further complexity may be required to alter call QOS requirements because of unanticipated situations which arise after the call has started. The user may wish to examine various network configurations to arrive at the one which best suits all his needs.

Once the call handling is completed, then the bridge/gateway operations which send, receive and service the packets and message can take place.

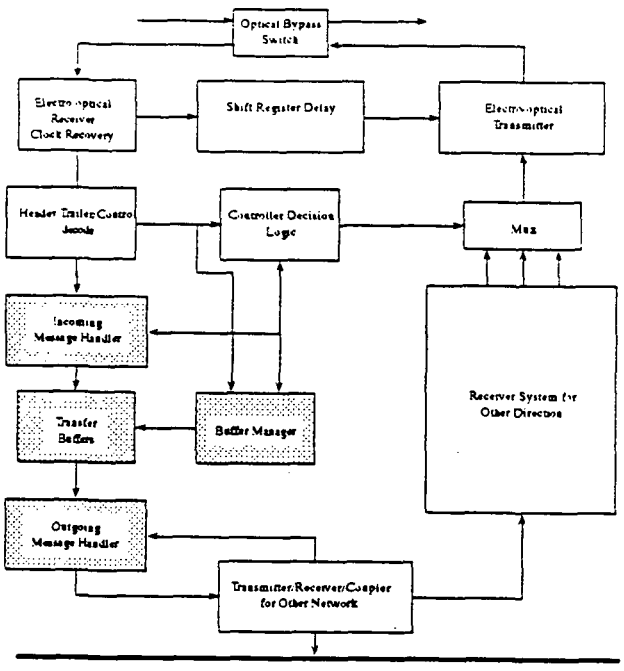

Figure 1. Bridge/Gateway Connection Diagram
Bridge/Gateway specific components shaded.

Message Classification

We do not treat many aspects of call handling in this paper. However, we identify three types of messages which are significant by the fact that they support a wide range of QOS requirements and should be identified individually for bridge/gateway operations. They are:

1. Synchronous messages which require call setup/termination at all B/G nodes between source and destination. Typically, these messages are voice or video type messages. Conditions for this class include known data rates, known route and known latency. This translates into operations where a fixed size buffer is allocated and headers for each link are built apriori. Synchronous messages are designated as class A
2. Asynchronous messages which choose to establish message control at B/Gs. Typically, large data blocks and file transfers use this class. Conditions for this class include unknown but source estimated data rates and known route. This translates into operations where fixed headers for each link are built apriori and where nominal buffer sizes are allocated. The operation requires a call setup/termination procedure. Asynchronous messages are designated as class B.
3. Asynchronous messages which are self-contained. A typical member of this class is e-mail. Data rates are not considered and routing is either self-contained or inserted by the bridge/gateway from a fixed route table. No call setup/termination operations are needed. Self-contained messages are designated as

class C.

The message classification scheme provides the minimum number of classes needed to satisfy the range of quality of service requirements. Class A, synchronous messages must have a unique classification since media access protocols support synchronous (isochronous) traffic differently from asynchronous traffic. Although voice and video have significantly different data rates, both are effectively handled with a single reservation type system. Class B and C traffic support typical asynchronous traffic situations. Most networks experience a bimodal traffic size distribution. Class B messages are identified through call setup and termination in order to preserve the efficiency over network links some of which require very small packet size, like ATM, and others accept large packet sizes, like FDDI. Class C traffic is for messages which need to get there but do not require a separate identity. Messages encompass those which are short and do not require call setup or negotiation for services which in itself may be time consuming. Latency is handled by establishing priorities within and between each traffic class and supporting it through queue control at the bridge/gateways. Note that reference [12] provides for two distinct types of service control, user and persistent connection/connectionless oriented data services but they are not oriented to the broad range of user needs.

B. Receiver Handling

A significant part of receiving packets and messages is the problem of resequencing. Based upon the nature of the interconnecting networks, two resequencing problems exist.

Virtual Circuit Resequencing

This resequencing problem arises when parallel virtual circuit channels are used to provide a high bandwidth connection. It occurs when individual blocks arriving at the receiver do not have message identity within their own right but whose ordering is provided by the location of the channels within the frames as they arrive. As in [16, 17], we assume that once the parallel channels to provide bandwidth have been reserved, that no further packet restructuring takes place dynamically. Hence, the resequencing between channels is formulated at call setup time and will remain fixed for the duration of the call.

The logic circuit, shown in Figure 2, consists of two parts. Based on present telephone virtual circuity, messages are separated into channels in frames. Each channel supports 64 Kbps, i.e., a byte of data each 12μsec. The FIFO buffers provides the necessary incremental delay to resequence channels in arriving frames assuming that each frame may arrive at the receiver via different routes including different intermediate switching nodes. At setup time, each buffer is loaded with the correct number of dummy bytes so that the byte at the head of each buffer represents the correct order for the bytes when sent. The stream address and transfer control portion of the circuit handles the reordering of channels when they arrive as a stream of bytes. Stream resequencing is required because bytes may be switched in a frame based upon random selection of channels circuits when the call is setup and the receiving circuity may arbitrarily order frame placement in the stream to the resequencer. The address stream is established at call setup by a special message where order is known and where the 125μsec. frame of bytes is deciphered to obtain the FIFO buffer address where each byte should be placed and the number of dummy bytes loaded into each FIFO buffer at initiation time.

Reference [17] discusses a similar resequencing system based upon transputers. In [16], we analyzed the use of parallel virtual circuit channels to support a high data rate ring network over existing telecommunications circuit. In that paper, we develop and discuss in greater detail the resequencing logic system shown in Figure 2. While the logic circuity is not able to handle dynamic bandwidth changes as easily as the transputer system [17], it reduces circuit latency from 5 msec. for the transputer system to a few μsec and should be significantly smaller and less costly to build.

Packet/Message Resequencing

The second resequencing situation occurs whenever packets are received and must be resequenced to be forwarded to another network where different operational conditions exist. In this situation, it is either impossible or impractical to maintain packets from messages in an identical format with only replacement of header information and operations such as checksum recalculation. Unlike virtual circuit resequencing, it is assumed that sufficient header information exists to correctly identify the packet and its placement in the message.

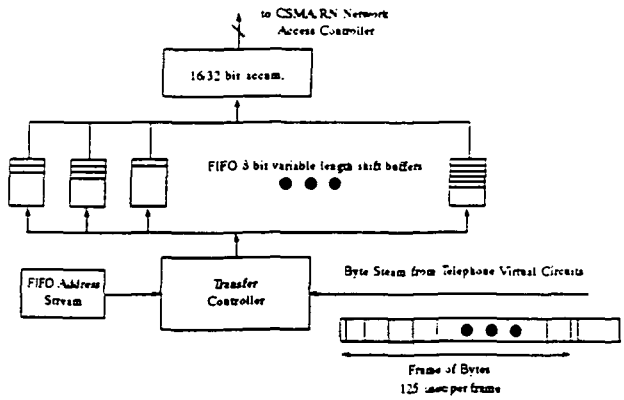

Figure 2 Virtual Circuit Resequencing Logic Diagram

Figure 3 shows the resequencing logic structure. As packets arrive, they are sent to the message handler which routes the message id and to the associative memory system and the packet length to the message update controller. A signal for additional space to place the packet is sent to the memory management unit. The message part of the packet is sent directly to free space in the memory buffer and is stored at the address provided by the MMU. The memory controller structure uses the concept of associative content addressable memory.

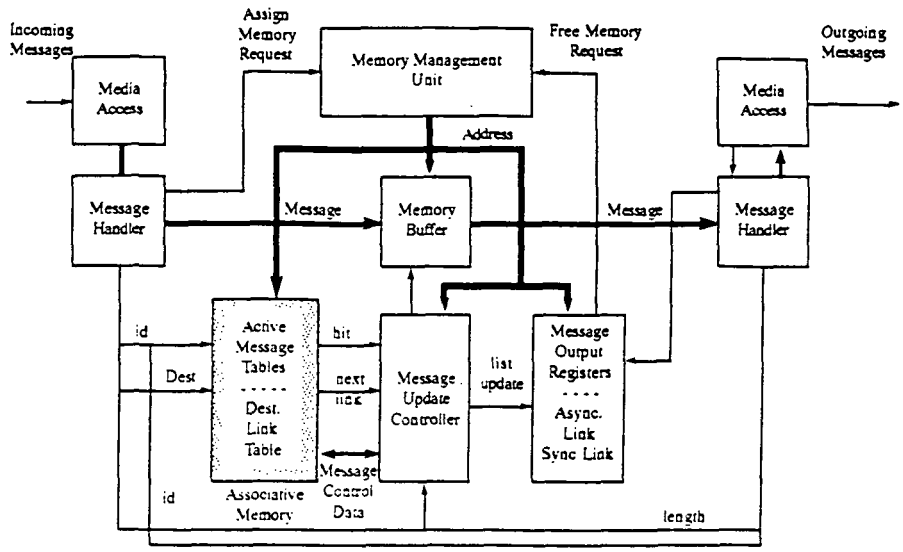

Figure 3 Bridge/Gateway Resequencing System

Outgoing messages are handled in a similar matter manner. After the controller selects the next message to be submitted to the outgoing channel, the control information is transferred to the message output registers. A number of message packets may be transferred since both synchronous and asynchronous messages ready to send may exit in the bridge/gateway memory buffer. Upon indication of the next media access, the information is transferred through the message handler and the information such as id and length which will change the content addressable memory data is feed back to the associate memory table to update the circuit information.

The system shown in Figure 3 is similar to systems which are being developed for ATM to host interfacing [9]. The major difference from the resequencing standpoint is that here a number of messages may exist simultaneously, that message classes require different handling than those to a host interface and that information obtained by the packet arrival is used to support both congestion and error control (see next two subsections).

Processing speeds for the resequencing system are estimated from the preliminary design. The resequencing system should be able to handle arrivals at high megabit data rates if nanosecond logic circuits are used [28]. Handling times, i.e., from the time the packet is received until it is placed in memory and linked properly, are estimated to be in the tens ofmsecs. While there is significant latency buildup while accumulating packets when the outgoing network has large block sizes, the actual added latency for the last message bit due to the resequencing is only the packet processing time. This is generally small in comparison to other delay such as end-to-end higher level protocol processing [18] and propagation delay in MAN and WAN systems.

C. Sender Handling

In past network systems, sender operations were generally straightforward. They consist of breaking messages into packet size blocks acceptable to the network, and, under the condition where the node has multiple links, routing packets. With the advent of ATM and the use of virtual paths which will be selected at call setup time [19, 20], the routing problem for networks using high data rate services over common carrier links will diminish. Routing for private "on-premise" systems is generally easily handled since the network configuration remains constant for large periods of time after installation or upgrade. However, with ATM and with the requirement to interconnect between a wide range of network data rates, the routing problem has been replaced by another equally important and difficult problem, that of congestion control

A number of congestion control methods have been proposed many based upon the concept of *statistical multiplexing* [21, 22] and implemented using the concept of a leaky bucket [21, 23]. The concept of statistical multiplexing implies that, should an overflow occur, the network can discard packets enroute to alleviate the congested condition. The argument in favor congestion alleviation by discarding packets is based upon the fact that both voice and video signals have a significant amount of redundancy so some loss should readily be tolerated. However, in multimedia environments, not all synchronous traffic will be voice/video and even then there is

[1]Many recent papers, conferences and journals have articles related to high data rate network flow control problems. The IEEE Communications Magazine, Vol. 29, No. 19, Oct 1991, devoted the entire issue to congestion control

strong evidence that compression techniques will be used since they can decrease bandwidth resource requirements by a factor of 50-100. Compressed data and many asynchronous messages have no tolerance for unreplaced lost packets. In the following subsection, we discuss a lossless congestion control scheme and follow that discussion an error control system with improved correction performance. Note that the FDDI-ATM interface [12] provides no flow or packet loss control services.

Lossless Congestion Control System

The lossless congestion control system operates similar to many end-to-end systems except that different feedback parameters are used and that control is exercised between bridge/gateway points. Here, the receiver periodically sends a control packet to its sender(s). This control packet contains the present free buffer space, the number of the last arriving packet accepted, and an error indication bit. When this information arrives at the sender, it calculates the remaining free buffer space at the receiver at the time the feedback packet was sent. It can send packets at the maximum rate until it has reduced the free buffer space to zero pending the arrival of a new feedback packet from the receiver. If the control packet indicates that an error has occurred then the number of the last arriving packet accepted is used by the sender as the starting point for resubmittal of all subsequent packets, i.e., a go-back-N scheme. In a direct replacement scheme, the feedback packet would contain the numbers of only the missing packets.

The concept of returning an acknowledge with the last accepted packet number is well known and has been used both for window congestion and error controls [24]. This information, in itself, is insufficient to avoid potential loss due to buffer overflow and packet discarding. However, adding free buffer space information allows the source to have sufficient knowledge to fill up but not overflow the destination buffer, regardless of the destination's ability to forward packets which frees additional space in its buffer. Thus, the system suffers no loss due to discarding at any node, intermediate or sender, participating in the control scheme.

Further, the lossless congestion control system is dynamic not only with respect to load but also resources. If the destination node has additional free buffers which it can commit to the message, when it sends its next control packet it adds these buffers to its free buffer count. The source does not know or care whether the amount of free buffer space is due to commitment of new buffers or the destination has been able to empty buffers by forwarding packets to subsequent destinations. Likewise, if the destination takes buffers from a link, it sends a control packet indicating fewer free buffers and as the buffers become empty, they can be assigned to another circuit. Again, the source is unaware of the cause. Additionally, the system is flexible with respect to potential loss. Although we have described the system as lossless, if the source wishes, it can send packets above those for which the destination indicates it has free buffers. It takes a chance that the buffer space will not be available upon their arrival at the destination and hence, may be lost. However, with the integrated error control, noted above and discussed more fully in the next subsection, the loss is easily and quickly replaced.

The lossless feedback scheme fulfills the requirement that the control information is readily and easily available. One needs only to keep a counter to maintain free buffer size and to register the last packet number when the incoming header and data are transferred to storage. No complex system is required to scan a frame or to do averaging to attain information such as present bandwidth use and no separate clocking is involved.

Figures 4a) - 4e) show typical results for the lossless congestion control system. Figure 4a) illustrates the tandem network test configuration and the conditions simulated. Simulator runs were takes for a steady state arrival and service rates. Data was started after the system had time to reach steady state, i.e., at least 10 times the maximum transfer delay between nodes, and data was taken so that 90% confidence interval results are expected. The lossless congestion control scheme and its performance are described more fully in reference [25].

Figure 4b) and 4c) show the mean packet delay time and mean packet service period at the source. As the congestion traffic is reduced, both the packet delay and service period decrease to their nominal conditions for an uncongested system. Figure 4d) shows that the feedback packet load on the network is less than 5% of the capable traffic and that it is not significantly influenced by the congestion condition. Hence, the control concept used to provide information to the sending node does not increase network use significantly. Figure 4e) illustrates that the control law is doing a good job, since under congested conditions, the percent of time that a node's queue is empty is less than 20%. This means that packets are available to be forwarded most of the time which is the best performance that can be obtained under congestion. Figure 5 shows the affect of buffer length during congestion. As buffer length increases, mean packet delay increases since the packets spend more time in intermediate node buffers but overall service rate decreases, since more total packets are transferred because the intermediate buffers are not empty as often. Thus, buffer assignment size is an important consideration is attaining overall performance.

[2]Note that packet identification for classification also supports *resequencing, congestion control and error correction.*

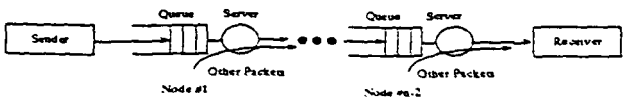

Figure 4a) Tandem Network Congestion Simulator

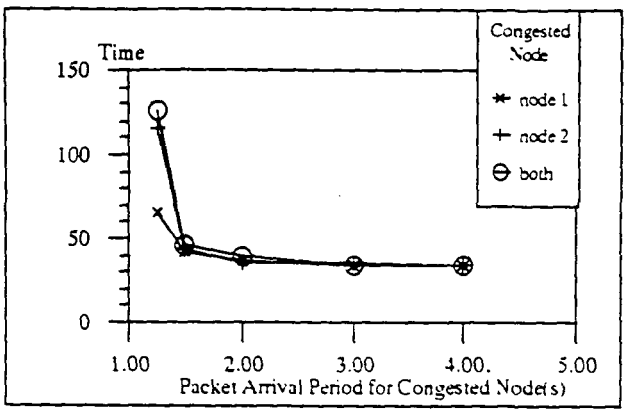

Figure 4b) Mean Transit Time

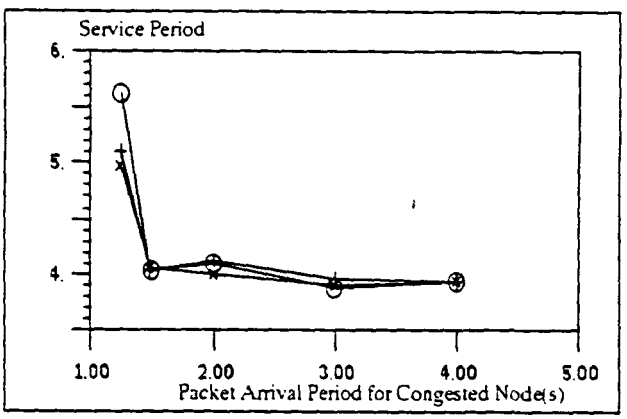

Figure 4c) Effective Source Service Period

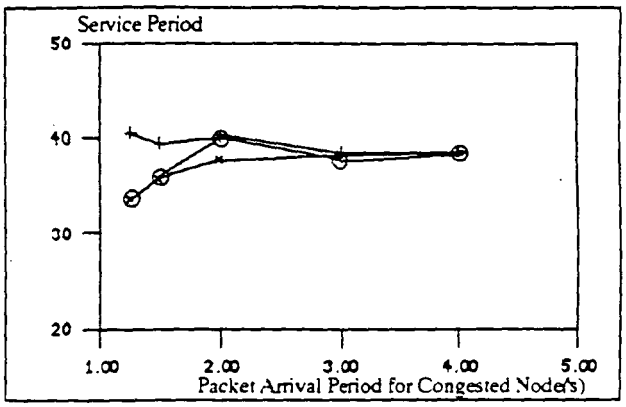

Figure 4d) Feedback Mean Service Period

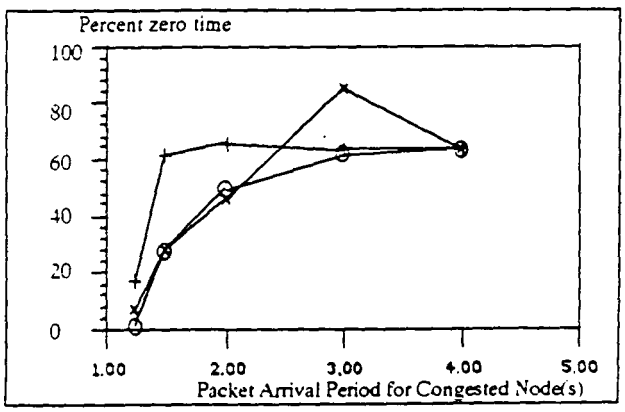

Figure 4e) Percent Time Node 1 Queue is Zero

Figure 4. Performance of Lossless Congestion Control System

Conditions - Nominal Service Period = 1.0
Other Packet Arrival Period = 1.25 - 4.0
Node Distance = 10.0
Sender Nominal Period = 4.0
(See Figure 4b) for Congestion Conditions)

D. Error Control

With the concept of congestion control relying heavily on the ability to drop packets, the question for error control becomes not whether but how. As with Internet, the mechanism suggested for ATM is based upon end-to-end error detection and correction at the transport layer [14, 26, 27]. While this mode is certainly feasible, it is highly questionable whether it is capable of enabling the wide QOS range that users have come to expect in LAN systems.

The error control technique as noted above can provide reliable datagrams between bridge/gateway systems handling messages. The congestion control scheme has the capability based upon the error indicator bit and last received packet number to alert the sender of loss regardless of cause. The error information is sufficient for the sender to replace the damage packet and reinitiate the remainder of the message without further coordination with the receiver. We have called this method of error correction between bridge/gateways intermediate error correction.

Performance for the intermediate error correction scheme is compared with the end-to-end error control implemented at transport level. The results are shown in Figure 6a) - 6b) for the conditions where intermediate links are 100 km and 1000 km long, respectively. It is assumed

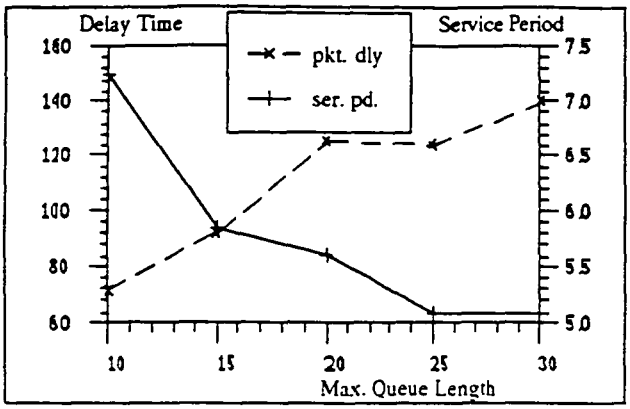

Figure 5 Effect of Queue Length on Congestion Control Performance

Conditions - Both node arrival periods - 1.25
Other conditions same as Figure 4

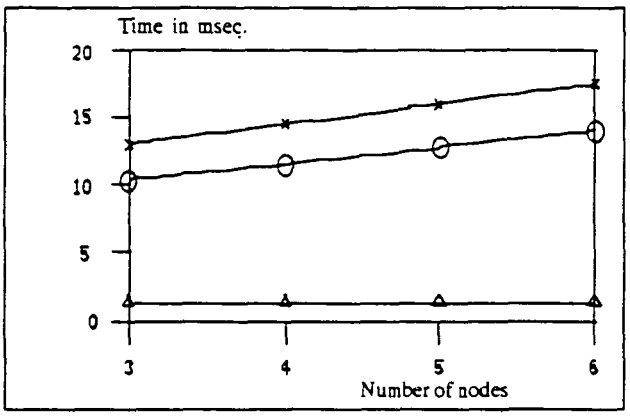

Figure 6a) Error Correction Time - 100km/link

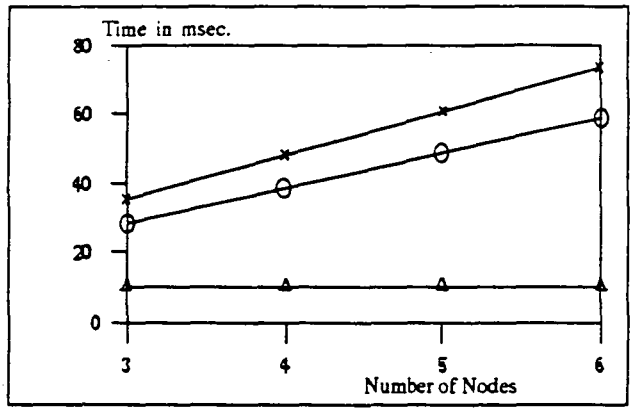

Figure 6b) Error Correction Time - 1000km/link

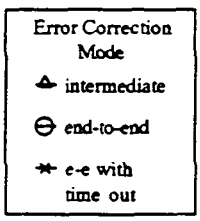

Figure 6 Intermediate Node Error Correction System

Conditions for Error Correction System
Node Processing
Upper Level - 2 msec.
Lower Level - 500μsec.

that bridge/gateway processing times are 100μsec and end-to-end transport processing for error control takes 2 msec. For transport layer correction, two mechanisms are modeled, one where the packet error is detected at the receiver and the error message immediately sent to the sender and the condition where the message times out. Time out is assumed as 1.5 times nominal end-to-end packet transfer time.

Link error correction shows significant improvement in time to correct an error. For example, correction time is always less than 1/3 of that required for end-to-end correction and as important, is dependent upon internode length instead of overall network length. This latter situation is especially valuable since constant QOS conditions can be supported independent of the total length of the communications link as long as maximum intermediate node distances are preserved.

V. Concluding Remarks

The bridge/gateway components required for interconnecting a wide range of high performance networks is developed in the paper. They exist because of the major requirement for bridge/gateway systems to support a wide range of network QOS requirements when hosts with differing capabilities are interconnected with through networks with equally wide range of capabilities.

Classification of messages into three classes where found to be the minimum by which bridge/gateways could support a wide range of traffic. Class A messages handle synchronous traffic; Class B and C for asynchronous traffic. Class B is for large messages which can tend to be bursty and requires a call setup and buffer allotment. Class C is for smaller "self-contained" messages which can be truly "packetized". To enable low latency message handling, the two asynchronous message classes, B and C, must have priority designations within the classes so that the bridge/gateway nodes can expedite critical traffic.

The paper presents effective resequencing hardware to support to conditions where arriving information may be disoriented. The first handles parallel virtual ciruits situations where messages may arrive in different channels. The combined switching buffering system provides rapid resequencing using hardware logic. with additional delays of only a few μsec.

The second resequencing systems is provided to handle messages where basic packet sizes are significantly different between the networks. If resequencing is not available, significant inefficiencies can occur. It is based upon identification of message packets as they arrive and to linking the packets in a buffering system. The structure of the buffering system is controlled by an associative memory system when packets are linked head-to-tail and where address pointers and lengths are associated with each

identifiable call. The resequencing system is designed to handle high data rates both incoming and outgoing and to provide delays ranging in the 10s to 100s of useconds.

Since message and packet identity are used for message classification and resequencing, the same information is availabe for congestion and error control. These features are extremely important in order to provide user QOS. A dynamic, lossless congestion control system is developed and its performance under typical operation is presented. The system, because it is based upon feedback of information from receiver to sender, completely avoids loss due to receiver buffer overflow. However, the system is very flexible so that operation where some loss may occur are easily implemented.

The parameters of the congestion control system directly enable intermediate node error recovery. Data is presented to demonstrate that intermediate node error control is able to correct errors usually within 1/3 the time as similar end-to-end error recovery systems. Thus, the combination of congestion and error control implemented at bridge/gateways provides significant performance improvement and hence, the ability to provide improved overall network QOS.

VI. References

1. Seifert, W.M.: "Bridges and Routers,"*IEEE Network;* Vol. 2; No. 1; Jan 1988.
2. Comer, D.:*Internetworking with TCP/IP*, Prentice Hall, NY. 1988.
3. Lin, Y.; Gerla, M.: "Brouter: The Transparent Bridge with Shortest Path in Interconnected LANs," Proc. of 16th LCN; Minneapolis, MN.; Oct. 14-17, 1991; pp. 175 - 183.
4. Hamner, M.C.; Samsen, G.R.: "Source Routing for Bridged Local Area Networks,"*IEEE Network* Vol 2; No. 1; Jan. 1988; pp. 33 - 36.
5. Perry, R.: "LAN/MAN Internetworking in the 802.6/SMDS Environment," Proc. of INFOCOM '90; pp. 639 - 648.
6. Braun, T.; Zittervart, M.: "A Transputer Based OSI-Gateway for LAN-Interconnection Across ISDN," Proc. of 16th LCN; Minneapolis, MN.; Oct. 14-17, 1991; pp. 158 - 165.
7. Chen, K.; Ho, K.; Saksena, V.R.: "Analysis and Design of a Highly Reliable Transport Architecture for ISDN Frame-Relay Networks,"*IEEE Jour. on SAC*, Vol. 7; No. 8; Oct. 1989; pp 1231 - 1242.
8. Cooper; E.C.; Steenkiste, P.A.; Samson, R.D.; Zill, B.D.: "Protocol Implementation on the Nectar Communication Processor," Proc. of SIGCOM '90; pp. 135 - 144.
9. Traw, C.B.; Smith, J.M.: "A High-Performance Host Interface for ATM Networks," Proc. of SIGCOM '91; pp. 317 - 325.
10. Davie, B.S.: "A Host-Network Interface Architecture for ATM," Proc. of SIGCOM '91; pp. 307 - 315.
11. Clark, D.D.; Tennenhouse, D.L.: "Architectural Considerations for a New Generation of Protocols," Proc. of SIGCOM '90; pp. 200 - 208.
12 Kapoor, S.; Parulkar, G.M.: "Design of An ATM-FDDI Gateway," Proc. of SIGCOM '91; pp. 173 - 183.
13. Little, T.D.C.; Ghafoor, A.: "Network Considerations for Distributed Multimedia Object Composition and Communication," *IEEE Network Magazine* Nov. 90; pp. 32 - 49.
14. Ohnishi, H.; Okada, T.: "Flow Control Schemes and Delay/Loss Trade-offs in ATM Networks,"*IEEE Jour. on SAC*; Vol. 6; No. 9; Dec. 1988; pp. 1609 - 1616.
15. Harita, B.R.; Leslie; I.M.: "Dynamic Bandwidth Management of Primary Rate ISDN to Support ATM Access," Proc. of SIGCOM '89; pp. 197 - 210.
16. Foudriat, E.C.; Maly, K; Zhang, L. Sun, W.: "Gateway Resequencing in Multinode Networks," ODU Tech Report; To Be Published.
17. Burren, J.W.: "Flexible Aggregation of Bandwidth fro Primary Rate ISDN," Proc. of SIGCOM '89; pp. 191 - 196.
18. Clark, D. Jacobson, V. Romkey, J. Salwen, H.: "Analysis of TCP Processing Overhead,"*IEEE Communications* Vol. 27; No. 6; June 1989; pp. 23 - 29.
19. Gerla, M.; et. al.: "Interconnecting LANs and MANs to ATM," Proc. of 16th LCN; Minneapolis, MN.; Oct. 14-17, 1991; pp. 259 - 270.
20 Sato, K; Ohta, S; Tokizawa, I.: "Broad-Band ATM Network Architecture Based on Virtual Paths,"*IEEE Trans. on Comm*; Vol 38; No. 8; Aug. 1990; pp. 1212 - 1222.
21. Dighe, R.; May, C.J.; Ramamurthy, G.: "Congestion Avoidance Strategies in Broadband Packet Networks," Proc. of INFOCOM '91; pp. 295 - 303.
22. Fendick, K.W.; Mitra, D.; et. al.: "An Approach to High-Performance, High-Speed Data Networks," IEEE Communications Magazine; Vol. 29; No. 10; Oct 1991; pp. 74 - 82.
23. Shoraby, K.; Sidi, M.: " On the Performance of Bursty and Correlated Sources Subject to Leaky Bucket Rate-Based Access Control Schemes," Proc. of INFOCOM '91; pp. 426 - 434.
24. Haas, Z.; Winters, J.H.: "Congestion Control by Adaptive Admission," Proc of INFOCOM '91; pp. 560 - 569.
25. Foudriat, E.C.; Maly, K; et. al.: "A Dynamic, Lossless Feedback System to Solve Network Congestion and Error Recovery," ODU Tech Report; To Be Published.
26. Mankin, A.: "Random Drop Congestion Control," Proc. of SIGCOM '90; pp. 1 - 7.
27. Jacobson, V.: " Congestion Control and Avoidance,*ACM Computer Comm*; April 1990; pp. 553 - 573.
28. Goksel, A.K.; et. al.: "A Content Addressable Memory Management Unit with On-Chip Data Cache,*IEEE Jour. of Solid-State Circuits*, Vol. 24; No. 3; June 1989; pp. 592 - 596.