[11] Edell, R.; Le, M. and McKeown, N.; "The BayBridge: A High Speed Bridge/Router," in *Proceedings of IFIPS Workshop on PfHSN*, May 92.

[12] Le, M, McKeown, N. and Edell, R.; "A High Performance SMDS Interface at STS-3c Rate," *IEEE J. Selected Areas in Comms*, submitted for this issue.

[13] Comer, D. E., "Internetworking with TCP/IP," *Prentice Hall*, Vol. 1, 2nd Ed., 1991

[14] Geiger, E.; Estes, G.; Sincoskie, W.D, Sammartino, F. and Lyles, B., "I Network Compatible ATM for Local Network Applications — Phase 1," *Proposal*, Vers. 1.0, April 92

6 Acknowledgements

We are grateful to Pacific Bell, Bellcore, Sun Microsystems, Advanced Micro Devices, TranSwitch, Integrated Device Technology, Viewlogic, Hewlett-Packard and MUSIC Semiconductors for their material and technical assistance. Much credit is also due to the following students at UC Berkeley who have contributed many hours to this project: Ting Kao, Fred Burghardt, Malik Audeh, George Kesidis, Karl Petty and Steve McCanne. And finally, we thank our faculty advisers Professors Pravin Varaiya and Jean Walrand for their support.

References

[1] "Generic Systems Requirements in Support of Switched Multi-megabit Data Service,"
Bellcore Technical Reference TR-TSV-000772, May 91

[2] Kapoor, S. and Parulkar, G.; "Design of an ATM-FDDI Gateway,"
in *Proc. of ACM/SIGCOMM'91 on Communication Architectures and Protocols*, Sept. 91

[3] "P802.1g MAC Remote Bridge Draft Standard," *IEEE Project 802 Committee P802.1g*

[4] Katz, D.; "A Proposed Standard for the Transmission of IP Datagrams over FDDI Networks,"
Draft RFC, Oct. 90

[5] Piscitello, D.; Lawrence, J.; "The Transmission of IP Datagrams over the SMDS Service,"
Draft RFC, Mar. 91

[6] Hedrick, C.; "Routing Information Protocol," Draft RFC, June 88

[7] Fedor, M; "GATED: A Multi-Routing Protocol Daemon for UNIX,"
Proceedings of the 1988 USENIX conference, San Francisco, California

[8] Moy, J.; "The OSPF Specification," Draft RFC, Oct. 89

[9] Institute of Electrical and Electronic Engineers, Inc. IEEE Standard 802.1D.

[10] Backes, F.; "Transparent Bridges for Interconnection of IEEE 802 LANs,"
IEEE Network, Vol. 2, No. 1, Jan. 88

more marked in the case of FDDI to FDDI bridging in which an increase of 138% is achieved if no learning is carried out. A large improvement in throughput could be achieved by adopting a scheme to reduce the time spent learning during periods of high throughput. For example: (1) source addresses could be stored in a separate FIFO and learnt during a less busy time, for example when the input FIFO becomes empty or when a frame is forwarded that is much longer than L_{min}; (2) many source addresses could be packed into a single frame and sent to the local host, which could periodically update the address tables; (3) source addresses from busy stations will still be learnt in a short time with an increase in performance.

5.2 Improving Performance Further

We have seen that high performance and flexibility is provided by the Protocol Converter. This is in part due to its customized design and partly because both Protocol Converters are able to operate in parallel for most of the time. But there are a number of limitations of the current design that are being considered for follow-on work:

Multiple Ports. The current design only allows two network interfaces. The architcture is fully-connected and does not lend itself to expansion to more ports. We are currently considering a design based on a fully-interconnected backplane using a fast, parallel crosspoint switch. The switch would effectively be an ATM LAN switch [14] with high speed bridging and routing capabilities on each port. Each port would contain one or more network interfaces and an integrated Protocol Converter sion unit.

Shared Address Tables. The current design is limited, in some configurations, by the bandwidth of the address tables. This would become more of bottleneck with the multiple port configuration described above. To overcome this, we are considering the use of a separate address-cache for each port. This could be in two levels —a large associative cache using a CAM and a small on-chip cache of similar size to a translation lookaside buffer (TLB) used in conventional processors. In addition, one or more ports would be used for large dedicated address tables for looking up unknown physical or network addresses. If the local cache(s) does not contain the destination address, the central address tables are consulted.

Header Prediction. A further advantage of the TLB cache is that MAC or IP headers could be cached and used as templates for header prediction.

5.1 The Cost of Encapsulation, Validation and Learning for FDDI-SMDS

It is worth summarizing the reasons for the lower performance when bridging between FDDI and SMDS rather than between FDDI and FDDI. These are:

Encapsulation— building the encapsulating SMDS and LLC/SNAP headers takes 2.25μs and contributes 33.8% of the time to bridge one packet from FDDI to SMDS. Although encapsulation is necessary for bridging, the processing time is made worse by a large number of SMDS header fields and 8 bytes of LLC/SNAP header (redundant, in the authors' opinion [12]).

SMDS Source Address Validation—in a private LAN such as FDDI, it is not usual for a local bridge to validate that a frame was received from a known host: in fact, in a learning bridge, there may be no such thing as a "known host" at start-up. But in an encapsulating bridge, where frames are received from a small number of remote bridges, it is feasible to verify that the frame was received from a legitimate member of the bridge group. This is prudent when frames are received over a public network. But we see that in this implementation, validation contributes over 23% of the time to process a frame received from SMDS. It should perhaps be left to the discretion of the network administrator to determine whether source address validation is necessary.

FDDI Destination Address Validation—A potential benefit offered by an encapsulating bridge is that all encapsulated FDDI frames received across the SMDS interface are potentially for stations reached via the local FDDI ring. This means that filtering is not strictly necessary. However, a bridge could receive an unnecessarily multicast SMDS frame from a remote bridge with less informative address tables than its own (the remote bridge may have only recently joined the group or could have no room left in its address tables). The receiving bridge may *know* that the frame is not destined for this FDDI ring and discard it. This potentially reduces traffic on the local ring. But it comes with a penalty of almost 1μs per frame, reducing the maximum throughput by over 17%. This is almost certainly not worth it, particularly as the remote bridge should quickly learn the correct address to forward frames to. Also, unlike SMDS, in most organizations the FDDI service is not charged per-frame.

Another large component of the per-frame processing time is address learning for bridging. This contributes almost 20% of the processing time in the FDDI to SMDS direction and over 30% in the SMDS to FDDI direction. This is even

Configuration	Per port or Aggregate	Conditions	Throughput (frames/sec)	T^b_{hdr} (μs)	L^b_{min} (bytes)
FDDI to SMDS	1 port	learning	150,375	6.65	187
		static	185,185	5.40	145
SMDS to FDDI	1 port	address validation	196,078	5.10	23
		no address validation	333,333	3.00	0
FDDI-SMDS	Aggregate	learning	300,751	—	—
		learning & no validation	321,637	—	—
FDDI-FDDI	1 port	learning	327,870	3.05	64
		static	555,555	1.80	26
	Aggregate	learning	465,116	—	—
		static	1,111,111	—	—

Table 1: Summary of bridging performance for different configurations of The BayBridge.

Configuration	Per port or Aggregate	Conditions	Throughput (frames/sec)	T^r_{hdr} (μs)	L^r_{min} (bytes)
FDDI to SMDS	1 port	with bridging	132,450	7.55	143
		no bridging	158,730	6.30	101
SMDS to FDDI	1 port	bridging & validation	133,333	7.50	197
		no bridging or validation	212,765	4.70	20
FDDI-SMDS	Aggregate	bridging & validation	266,666	—	—
		no bridging or validation	372,340	—	—
FDDI-FDDI	1 port	with bridging	168,067	5.95	89
		no bridging	212,765	4.70	48
	Aggregate	with bridging	336,134	—	—
		no bridging	425,530	—	—

Table 2: Summary of routing performance for different configurations of The BayBridge.

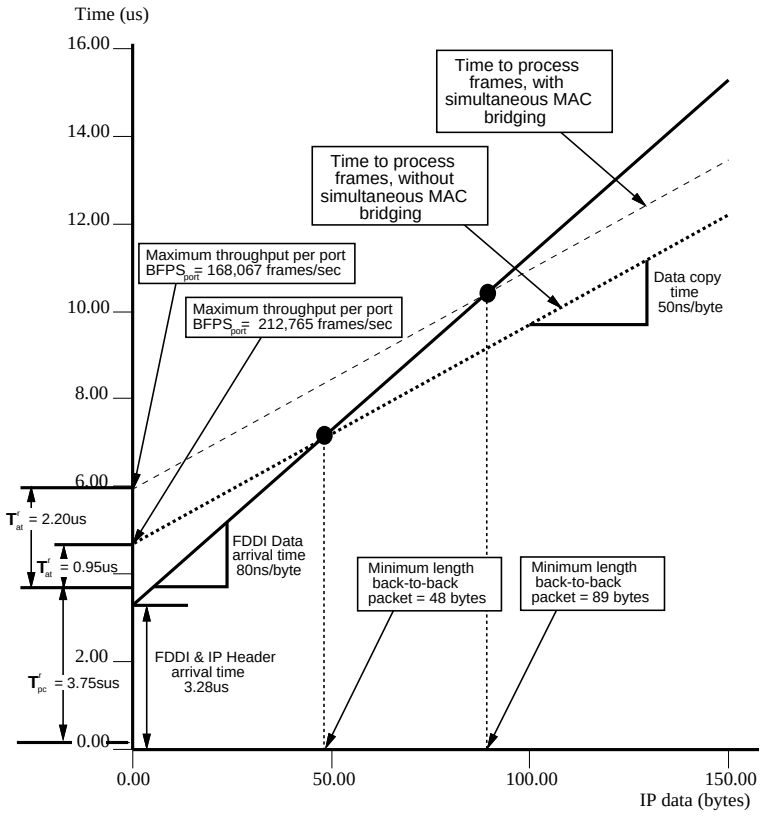


Figure 14: Performance of one port of The BayBridge for FDDI-FDDI routing with and without simultaneous bridging with learning.

Building LLC/SNAP & IP Headers 35 cycles

Address Lookup 19 cycles

Address Learning 25 cycles (if also bridging)

Copying IP packet 1 cycle per byte

Hence, $T^b_{hdr} = 5.95\mu s$. This corresponds to a maximum throughput of $RPPS_{port} = 168,067$ packets per second per port. If the router is *not* also working as a bridge then the throughput increases by 26.6% to $RPPS_{port} = 212,766$ packets per second per port.

Figure 14 illustrates the performance of one port of The Bay Bridge for FDDI-FDDI routing with and without learning. As can be seen from the figure, $L^r_{min} = 89$ bytes if the router is also acting as a bridge. This decreases to $L^r_{min} = 48$ bytes if the router does not learn MAC addresses.

4.3.5 Aggregate FDDI-FDDI IP Routing

For FDDI-FDDI routing, the address tables do not provide a bottleneck and so the aggregate throughput is simply twice the per-port throughput. With simultaneous bridging $RPPS_{agg} = 336,1346$ packets per second and without bridging increases by 26% to $RPPS_{agg} = 425,530$ aggregate packets per second.

4.4 Summary of Performance

Table 1 summarizes the bridging performance for The Bay Bridge configured as an FDDI-FDDI and FDDI-SMDS bridge and Table 2 summarizes the bridging performance for The Bay Bridge configured as a router.

5 Conclusions

We have described in some detail architecture, function and performance of The Bay Bridge for interconnecting FDDI LANs over the public SMDS network. Total aggregate throughput for FDDI-SMDS was shown to exceed 300,000 frames per second for bridging and 250,000 packets per second for routing. We have also presented results for FDDI-FDDI bridging and routing. In this configuration, the aggregate bridging throughput increases by over 50% to more than 450,000 frames per second.

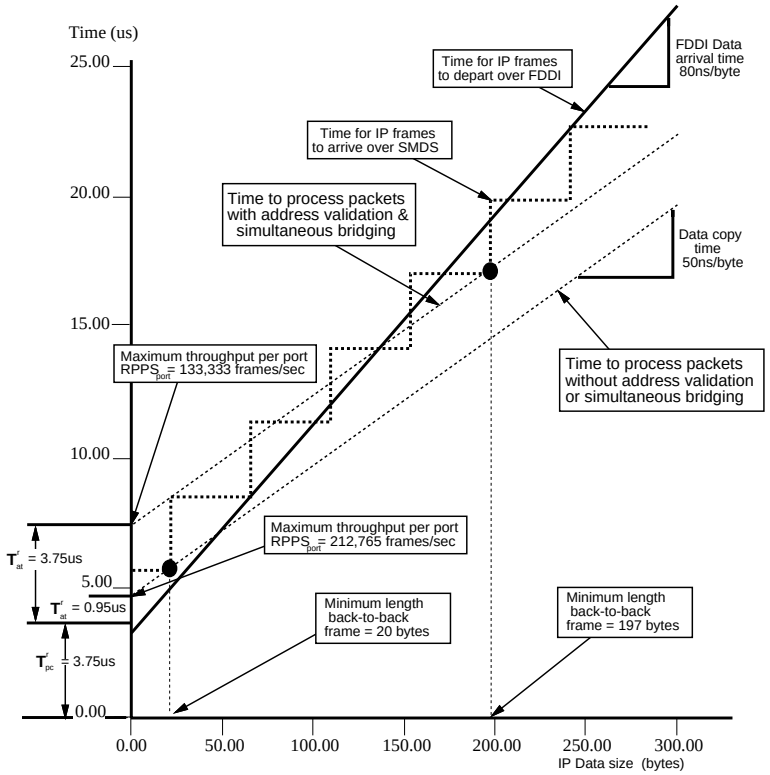


Figure 13: Performance of one port of The Bay Bridge for SMDS-FDDI routing with and without simultaneous bridging and address validation.

4.3.2 SMDS to FDDI IP routing

The time for The Bay Bridge to route one IP packet in hardware from an SMDS network to an FDDI network is made up of the following components:

Protocol Converter Set-Up 40 cycles

Building LLC/SNAP & IP Headers 35 cycles

Address Lookup 19 cycles

Address Learning 32 cycles (if also bridging)

Validate SMDS Source Address 24 cycles

Copying IP packet 1 cycle per byte

Hence, T^b_{hdr} =7.50μs. This corresponds to a maximum throughput of $RPPS_{port}$ = 133,333 packets per second per port. If the router is *not* also working as a bridge and SMDS source addresses are not validated then the throughput increases by 59.6% to $RPPS_{port}$ =212,766 packets per second per port.

Figure 13 illustrates the performance of one port of The Bay Bridge for SMDS-FDDI routing with and without bridging and SMDS source address validation. L^r_{min} = 197 bytes in this case and drops to L^r_{min} = 20 bytes if the router is not simultaneously acting as a bridge and if SMDS source addresses are not validated.

4.3.3 Aggregate FDDI-SMDS IP Routing

Using the same technique as illustrated in Figure 8 we determine the aggregate throughput for FDDI-SMDS IP routing. With simultaneous bridging in both directions and with address validation, $RPPS_{agg}$=266,666 aggregate packets per second. Without simultaneous bridging or address validation, throughput increases by 40% and $RPPS_{agg}$ =372,340 aggregate packets per second.

4.3.4 FDDI-FDDI IP Routing

The time for The Bay Bridge to route one IP packet in hardware from one FDDI network to another FDDI network is made up of the following components:

Protocol Converter Set-Up 40 cycles

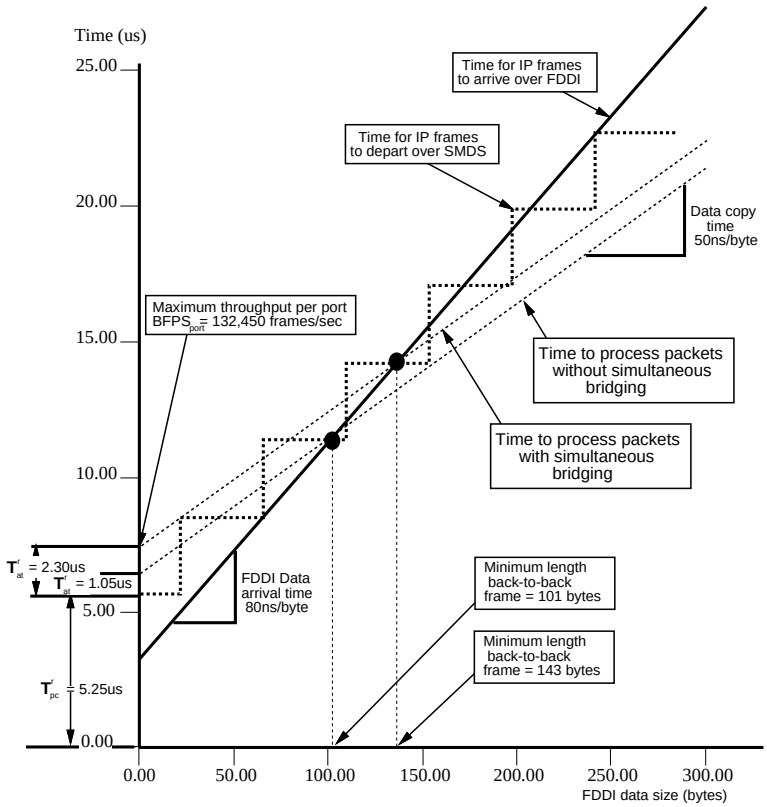


Figure 12: Performance of one port of The Bay Bridge for FDDI-SMDS routing with and without simultaneous bridging with learning.

Address Learning 25 cycles (if also bridging)

Copying IP packet 1 cycle per byte

Hence, T^b_{hdr} =7.55μs which corresponds to a maximum throughput of $RPPS_{port}$ = 132, 450 packets per second per port. If the router is *not* also working as a bridge then the throughput increases by 19.8% to $RPPS_{port}$ = 158, 730 packets per second per port.

Figure 12 illustrates the performance for FDDI to SMDS routing with and without simultaneous bridging. With bridging, continuous back-to-back IP packets containing L_{min} = 143 bytes of data may be routed. Without bridging, L^r_{min} =101.

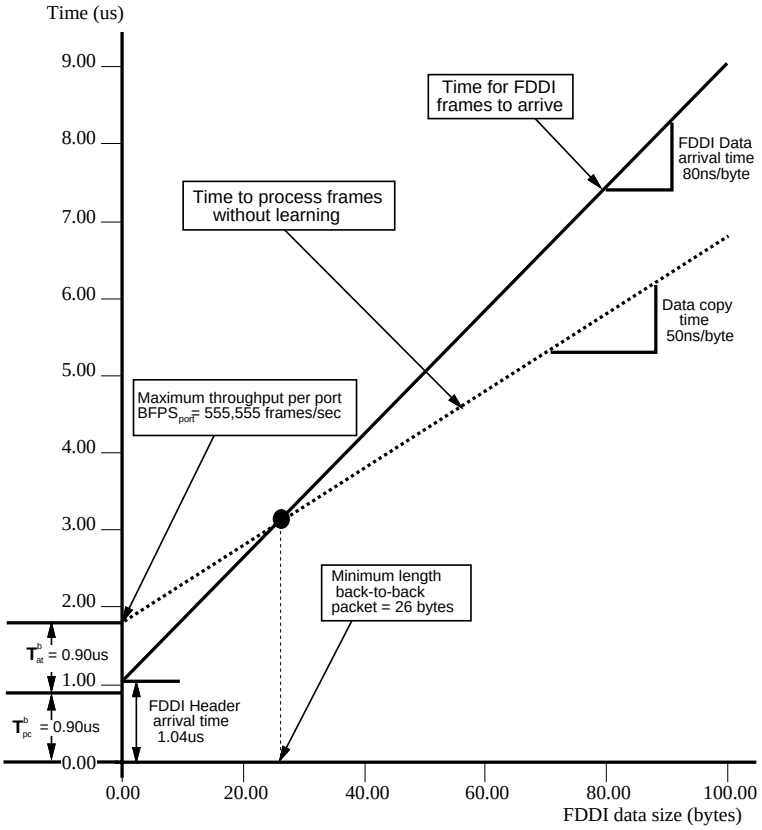


Figure 11: Performance of one port of The BayBridge for FDDI-FDDI bridging with static address tables.

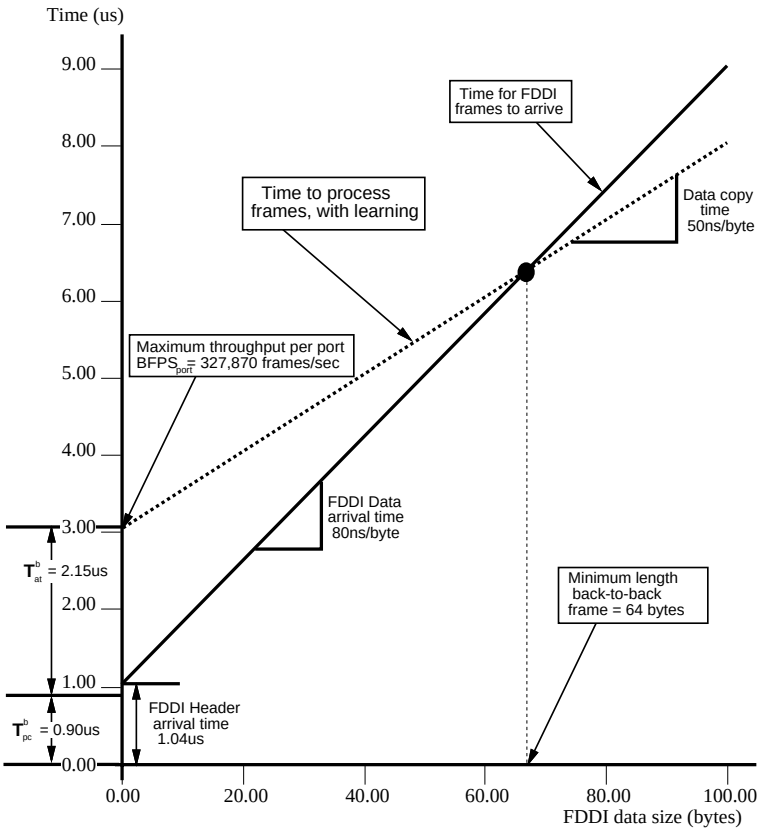


Figure 10: Performance of one port of The BayBridge for FDDI-FDDI bridging with source address learning.

Protocol Converter Set-Up 18 cycles

Address Lookup 18 cycles

Address Learning 25 cycles

Copying FDDI frame 1 cycle per byte

Hence, $T^b_{hdr} = 3.05\mu s$.

This corresponds to a maximum throughput of $BFPS_{port} = 327,870$ packets per second per port and $BFPS_{agg} = 465,116$ aggregate frames per second. It is interesting to compare this with the performance that may be obtained if *static* address tables are assigned by the network administrator: $BFPS_{port} = 555,555$ packets per second per port and $BFPS_{agg} = 1.1$ million aggregate frames per second.

Figures 10 and 11 illustrate the performance of one port of The Bay Bridge for FDDI-FDDI bridging with and without learning respectively. For a learning bridge, back-to-back frames of length $L_{min} = 64$ bytes may be forwarded. For a statically allocated address table, $L_{min} = 26$. This may be compared with the minimum length TCP/IP packet of 48 bytes [4].

4.3 Routing Performance

The routing performance described here covers the routing of IP packets that conform to the proposed standards for FDDI and SMDS [4, 5]. It is assumed that the IP checksum is *not* checked, but simply modified to reflect the change in the *Time To Live* field. Fragmentation may be required in SMDS to FDDI routing, but is only required for packets longer than 4500 bytes. As we are only considering small packets here, fragmentation will be ignored.

4.3.1 FDDI to SMDS IP routing

The time for The Bay Bridge to route one IP packet in hardware from an FDDI network to an SMDS network is made up of the following components:

Protocol Converter Set-Up 40 cycles

Building LLC/SNAP & IP Headers 65 cycles

Address Lookup 21 cycles

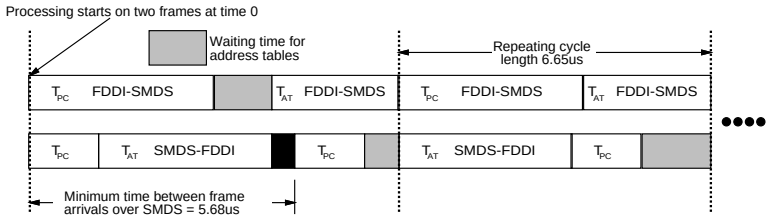


Figure 8: Calculation of aggregate throughput for FDDI-SMDS bridging in both directions simultaneously. After processing of frames starts at time 0, contention for the address tables will lead to repetitive cycles of constant length, $6.65\mu s$.

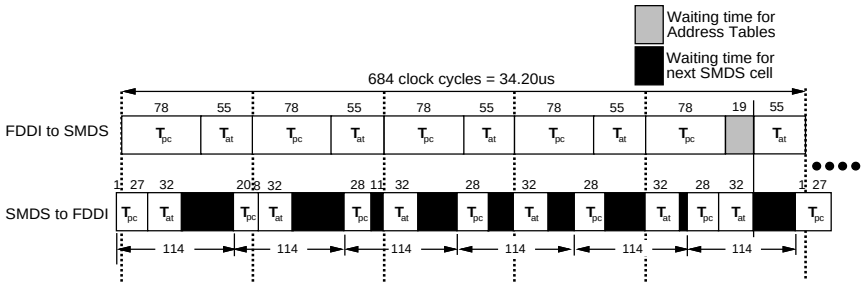


Figure 9: Calculation of aggregate throughput for FDDI-SMDS bridging in both directions simultaneously without address validation by the SMDS interface. At maximum throughput, repetitive cycles occur of constant length, $34.2\mu s$.

arrive over the SMDS interface every 2 cell times ($5.68\mu s$). The processing falls into a recurring cycle of length $6.65\mu s$ in which time two frames are processed. This corresponds to an aggregate throughput $BFPS_{agg} = 300,751$ frames per second.

If addresses are not validated on the SMDS interface then the situation is more complicated still, see Figure 9. This time the recurring cycle will be $34.2\mu s$ long in which time 11 frames are bridged. This corresponds to just a 7% increase in aggregate throughput to $BFPS_{agg} = 321,637$ frames per second.

423 FDDI to FDDI Bridging

The time to bridge one frame is made up of the following components:

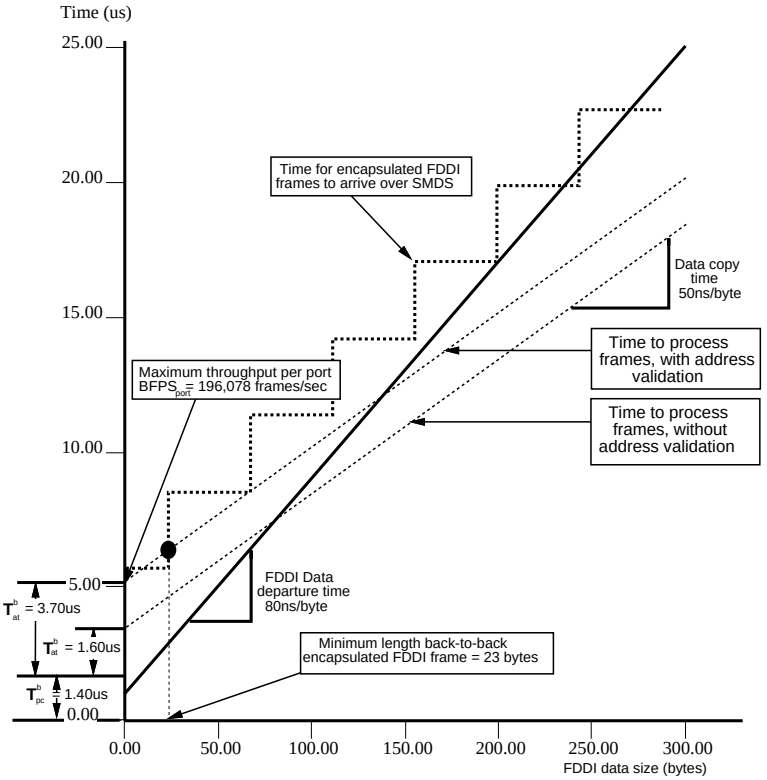


Figure 7: Performance of one port of The BayBridge for SMDS-FDDI bridging with and without SMDS source and FDDI destination address validation.

Figures 7 illustrates the performance of one port of The BayBridge for SMDS to FDDI bridging respectively with and without SMDS source address validation and FDDI destination address validation. The per-frame overhead time $^{b}_{hdr}T$ is broken down into T^b_{pc} and T^b_{at}. Without address checking, $^{b}_{min}L = 0$ bytes.

422 Aggregate FDDI-SMDS Bridging

The aggregate throughput of The BayBridge for FDDI-SMDS bridging is not as straightforward as for FDDI-FDDI bridging because of the asymetric use of the address tables and the constraint that encapsulated frames can not arrive with a separation of less than 2 cell times over the SMDS interface. The aggregate throughput calculation is illustrated in Figure 8. Assume that back-to-back minimum length frames arrive over the FDDI interface and that frames

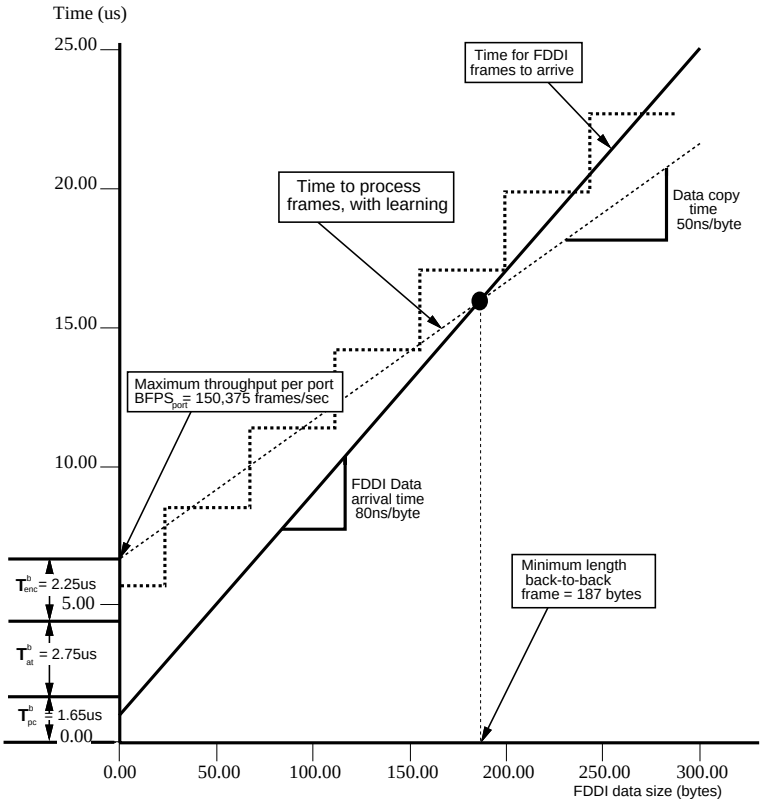


Figure 6: Performance of one port of The BayBridge for FDDI-SMDS bridging with source address learning.

Building SMDS Header 45 cycles

Copying FDDI frame 1 cycle per byte

where one cycle is 50ns. Hence, $^{b}_{hdr}T = 6.65\mu s$. The per-frame processing time is dominated by the time to build an SMDS header (36 bytes plus 8 bytes of LLC/SNAP header). The protocol converter must also calculate the *BAsize* (length) field that appears in the SMDS header and the padding bytes. As we shall see, the address lookup takes longer than for FDDI-FDDI bridging because a 64-bit SMDS address must be retrieved from the address tables. The per-frame processing time corresponds to a maximum throughput of, $BFPS_{port} = 150,375$ frames per second per port. With *static* address tables allocated by the network administrator the throughput increases by 23% to $BFPS_{port} = 185,185$ frames per second per port. However, we shall not consider the option of static address tables for FDDI-SMDS as this would potentially lead to redundant frames being passed over and billed by the SMDS service.

Figures 6 illustrates the performance of one port of The Bay Bridge for FDDI-SMDS bridging with learning. The per-frame overhead time, $^{b}_{hdr}T$ is broken down into T^{b}_{pc}, T^{b}_{at}, and T^{b}_{enc}. We see from the graph that for a learning bridge, back-to-back frames of length, $^{b}_{min}L = 187$ bytes may be forwarded.

We now consider the performance of The Bay Bridge bridging from SMDS to FDDI:

Protocol Converter Set-Up 24 cycles

Validate SMDS Source Address 24 cycles

Validate FDDI Destination Address 18 cycles

SMDS/FDDI Address Learning 32 cycles

Discard SMDS Header 4 cycles

Copying FDDI frame 1 cycle per byte

where one cycle is 50ns. Hence, $^{b}_{hdr}T = 5.10\mu s$. This corresponds to a maximum throughput of $BFPS_{port} = 196,078$ frames per second per port. We could consider increasing the performance to $BFPS_{port} = 256,410$ by not validating the SMDS source address and further to $BFPS_{port} = 333,333$ by assuming that all arriving encapsulated FDDI frames are destined for this ring. However, as can be seen from Figure 7, these measures are not needed unless the average incoming encapsulated FDDI frame contains less than $^{b}_{min}L = 23$ bytes of data.

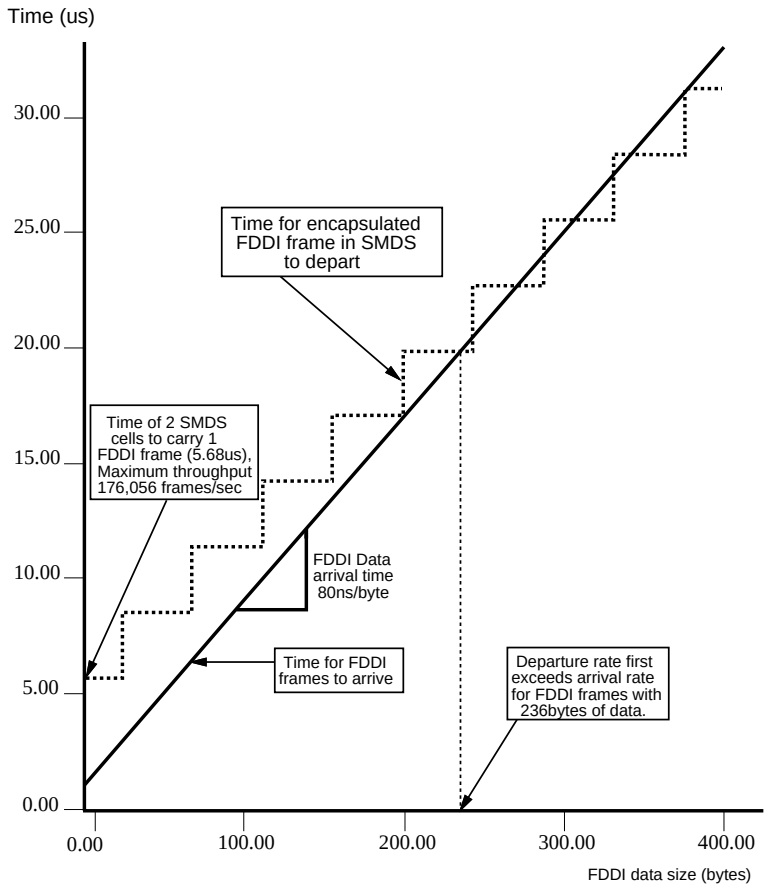


Figure 5: Overhead of bridging one FDDI frame over SMDS requires at least two cells because of the large overhead of the SMDS frame header and the LLC/SNAP fields.

For bridging, we will also deduce the minimum size frame that may be bridged *back-to-back* continuously without filling the system buffers. This will be denoted L^b_{min} . Similarly for routing we will define L_{min} for IP packets[4].

Note that for routing we will not take into account packets of unknown destination that must be sent to the host. The performance of this path is not known at this time.

4.2 Bridging Performance

We will consider the performance of The Bay Bridge operating as a MAC-layer bridge in two configurations: first as an FDDI-SMDS bridge, then as an FDDI-FDDI bridge.

4.2.1 FDDI-SMDS Bridging

The bridging performance of The Bay Bridge operating in this configuration will be different in each direction. We shall first consider the performance from FDDI to SMDS, but first it is interesting to compare the maximum rate of frame arrivals from FDDI and departures over SMDS operating at 155Mbps. Assuming minimum length FDDI frames of 20 bytes, the maximum arrival rate is 625,000 frames per second. An SMDS frame (L3_PDU) containing an encapsulated, minimum length FDDI frame is at least 65 bytes long (excluding optional CRC for SMDS and any padding bytes). This requires two 44-byte cells corresponding to a maximum encapsulated frame rate over SMDS of just 177,305 frames per second. Figure 5 shows how this overhead becomes less significant for longer FDDI frames. But it is not until the FDDI frames reaches 236 bytes that the departure rate matches the arrival rate. Furthermore, it is not until the arriving FDDI frames are longer than 450 bytes that the departure rate will always exceed the arrival rate.

The time for The Bay Bridge to bridge one FDDI frame to SMDS, encapsulating the frame into an SMDS L3_PDU is made up of the following components:

Protocol Converter Set-Up 33 cycles

Address Lookup 30 cycles

Address Learning 25 cycles

[4] Clearly, if the average packet length is larger than L_{min}, the finite buffer space may still overflow. But this will depend on the incoming traffic patterns and is not considered here.

3.4 Host and SBus DMA Card

The Host is a Sun SPARCStation and acts primarily as a monitor and manager; it is *not* involved in bridging data packets between the two networks.

4 System Performance

4.1 Performance Characterization

In order to understand the performance of a bridge or router, the per-packet processing may be broken down into its constituent components: header processing and data copying. For bridging, header processing includes making a destination port decision and may include encapsulation. Data copying consists of copying the MAC-layer frame. For routing, header processing includes the next-hop routing decision, changing header parameters and may include fragmentation control and the recalculation of checksums. Data copying consists of copying the network-layer packet. In most systems, the data copying rate is faster than the data rate of the network ports and so does not provide a bottleneck. The bottleneck is almost invariably the *header processing* time. This is the case for The Bay Bridge. The data copying is always 1 byte per 50ns cycle, whereas the header processing depends on the complexity of the protocol.

To analyze the performance of The Bay Bridge we break down the per-frame or per-packet processing time into the header processing time, (T^b_{hdr} for bridging and T^r_{hdr} for routing) and the data copying time. We will further divide T_{hdr} into the decision time by each protocol converter operating in parallel, T_{pc} the address-table lookup time of the shared address tables, T_{at}, and where relevant, the encapsulation time, T_{enc}.

From the header processing time we will infer the maximum throughput. For bridging, the *maximum number of bridged frames per second per port* is denoted $BFPS_{port}$ and for routing the *maximum number of routed packets per second per port* is denoted $RPPS_{port}$. These are given by,

$$BFPS_{port} = \frac{1}{T^b_{hdr}}$$

$$RPPS_{port} = \frac{1}{T^r_{hdr}}$$

Aggregate throughput of both ports operating simultaneously will be denoted $BFPS_{agg}$ and $RPPS_{agg}$ respectively. The calculation of $BFPS_{agg}$ and $RPPS_{agg}$ depends on the particular configuration and is described later.

messages, the source address of the remote FDDI station is stored in the Address Table, along with the associated remote bridge address (E164) (case 2 above).

2) Hardware Routing Assistance.

Hardware routing assistance is provided to the host by the Protocol Converter. The Address Tables are used to hold a cache of Network Layer Addresses (we shall assume here that the network layer is IP). These are added as static entries by the host. When a frame arrives the Protocol Converter consults the Address Tables to see if the IP address is known. If it is, the Protocol Converter makes any necessary changes to the IP header, i.e., decrements the *Time To Live* field and updates the *Checksum*, and builds the new MAC frame required by the destination port. Fragmentation may also be performed by the Protocol Converter. If the Network Address is *not* known the frame is sent to the host. The host consults its address tables and network to physical address resolution cache and determines the destination address for the frame. After forwarding the frame, the host updates the Address Tables so that in future the Protocol Converter will route the frame in hardware. Note that the Protocol Converter does not conform to the regular model in which the routing decision is made in two stages: first the IP address of the next-hop is determined and then the IP address is bound to a physical address. Higher performance is achieved by carrying out both stages at once, but requires the host to update the address tables accordingly. Also note that header prediction is *not* performed. Header prediction with a cache of one header could be readily performed by the Protocol Converter but has not been investigated.

3.2 SMDS Interface

The SMDS Interface to The Bay Bridge is described in detail in [12] and is not discussed here.

3.3 FDDI Board

The FDDI board is a standard design and is not described in detail here. We use the AMD FDDI Supernet chipset to control access to the FDDI ring and a 64Kword packet buffer between the FDDI MAC and the Bridge/Router Board.

consults the Address Tables. The FDDI Destination Address (DA) is used as an index into the Address Tables. If a match occurs, the *Copy Bits* are read from the table to decide where to send the frame. The copy bits may indicate that:

1. The frame is to be copied to the host.
2. The encapsulated frame is to be copied to the SMDS interface. The SMDS message header is built by the Protocol Converter and sent to the *Other Port Channel*. The SMDS destination address is read from the *Associated Data Table* in the Address Tables. The FDDI frame is copied to the *Other Port Channel* followed by the SMDS message trailer.
3. The frame is to be copied to both ports. The SMDS message header is sent to the *Other Port Channel*, the FDDI frame is copied to both channels and the SMDS message trailer is sent to the *Other Port Channel*.
4. The frame is not to be copied to either port. This is because the frame is destined for another station on the same FDDI ring and therefore it is not forwarded.

If a match does *not* occur in the Address Tables, then the destination is unknown. A multicast SMDS frame is built and the encapsulated frame is sent to every bridge in the *Bridge Group*.

SMDS messages arrive across Port B and are received by Protocol Converter B (PCB). First of all PCB may perform optional address validation of the SMDS source address and the FDDI destination address. The SMDS address is verified in the address tables to be part of the bridge group. This is an optional security measure to check that the frame was received from a legal source. The FDDI destination address is checked in the address tables to check whether it should be delivered to the local FDDI ring. If the copy bits indicate that the frame should be copied only to the SMDS port, then we may deduce that the encapsulated FDDI frame was delivered in error and is dropped. Following the validation, the FDDI frame is removed from the SMDS frame. The (FDDI source address, SMDS source address) pair are loaded into and learnt by the address tables. The frame is then forwarded to the FDDI ring.

Learning:
The Protocol Converter updates the Address Tables to *learn* the addresses of stations on the local and remote networks. PCA learns the source address (IEEE 48-bit format) of all passing FDDI frames and updates the Address Tables indicating that frames destined to this address should not be copied from the FDDI ring (case 4 above). When PCB decapsulates FDDI frames from SMDS

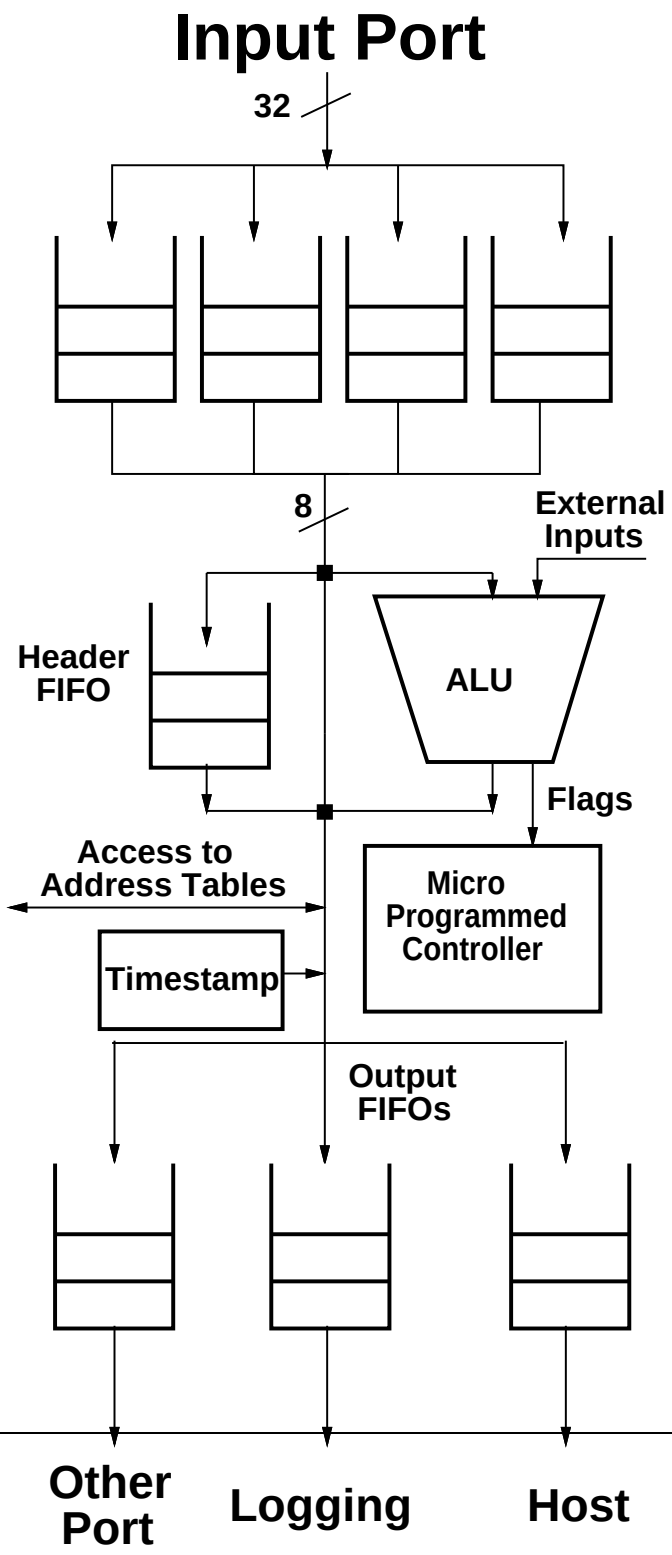


Figure 4: Block diagram of Protocol Converter: a programmable device for converting between network protocols.

3.1.2 Protocol Converter

This is a custom built, microcoded unit, optimized for fast conversion between network protocols. A block diagram is shown in Figure 4.

For a simple, flexible and fast design, a horizontal micro-instruction is used. As a result, a wide range of micro-instructions are available to the programmer for performance optimization. The microcode is developed using a user-definable instruction set and a custom assembler. The assembled microcode is downloaded to each Protocol Converter during initialization.

During the processing of a frame, the converter searches for patterns in protocol headers (e.g. MAC headers, LLC, SNAP, IP) using the ALU and stores the frame header in the Header FIFO. The Protocol Converter arbitrates for and consults the Address Tables to make forwarding and filtering decisions. When a decision has been made, the output frame is built and forwarded to one or more output channels. Frames destined for the other network interface are forwarded to the *Other Port* channel, frames destined for the host are forwarded to the *Host* channel. Logging frames may also be sent to the local host to log performance statistics or to provide call-billing information. Call logging is user defined as part of the downloaded program to the Protocol Converter. For example, a logging frame could be sent to the host each time a frame is forwarded to facilitate billing. The logging frame could contain a timestamp (one clock cycle resolution), the source address, destination address and frame length. For performance monitoring, an ALU register could be used to count the total number of frames forwarded. Every time the register overflows, a logging frame could be sent to the host containing an identifier and a timestamp.

We now describe how the Protocol Converter is used for (1) MAC Level Bridging and (2) Hardware Routing Assistance:

1) MAC Level Bridging: Forwarding, Filtering and Learning

Forwarding and Filtering:
Referring to Figure 2, assume that Port A is connected to an FDDI network interface and that Port B is connected to an SMDS interface. All FDDI frames from the ring are passed to the Bridge/Router Board across Port A, delimited by a Tag bit. Protocol Converter A (PCA) is located at the *input* to Port A. When a frame arrives, PCA reads the header one byte at a time, checking each header field as it proceeds. The first byte of an FDDI frame will contain the Frame Control (FC) field indicating whether the frame is for LLC or SMT, contains 16 or 48-bit addresses, etc. If it is an SMT frame, then it is sent to the host via the *Host* channel.

If the frame is not an SMT frame, the Protocol Converter arbitrates for and

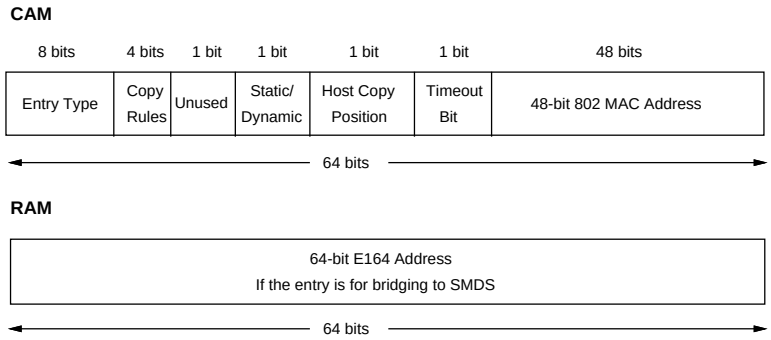


Figure 3: Format of Address Tables entries.

3.1 Bridge/Router Board

The Bridge/Router Board is responsible for making per-frame decisions. For bridging, this includes forwarding and filtering of frames. For routing, this includes the calculation of header checksums, decrementing the *Time To Live* field, and next-hop determination. This is carried out by a pair of Protocol Converters and the Address Tables.

3.1.1 Address Tables

The address table provides storage for 4096 addresses (expandable with a daughterboard). Each entry consists of 64 bits of associative memory (CAM) and 64 bits of non-associative memory (RAM). The format of the entries is determined by the Protocol Converter microcode, i.e., the hardware does not restrict the format. As an example, the format used for bridging between FDDI and SMDS is shown in Figure 3. The unrestricted format enables physical and network addresses to co-exist in the address tables. The address type is indicated in each table entry.

The Address Tables may be accessed by either Protocol Converter or the attached workstation. Access is arbitrated on a round-robin basis. The RAM is accessed by first obtaining a match in the CAM, and then loading the index of the matching CAM entry into a memory address register.

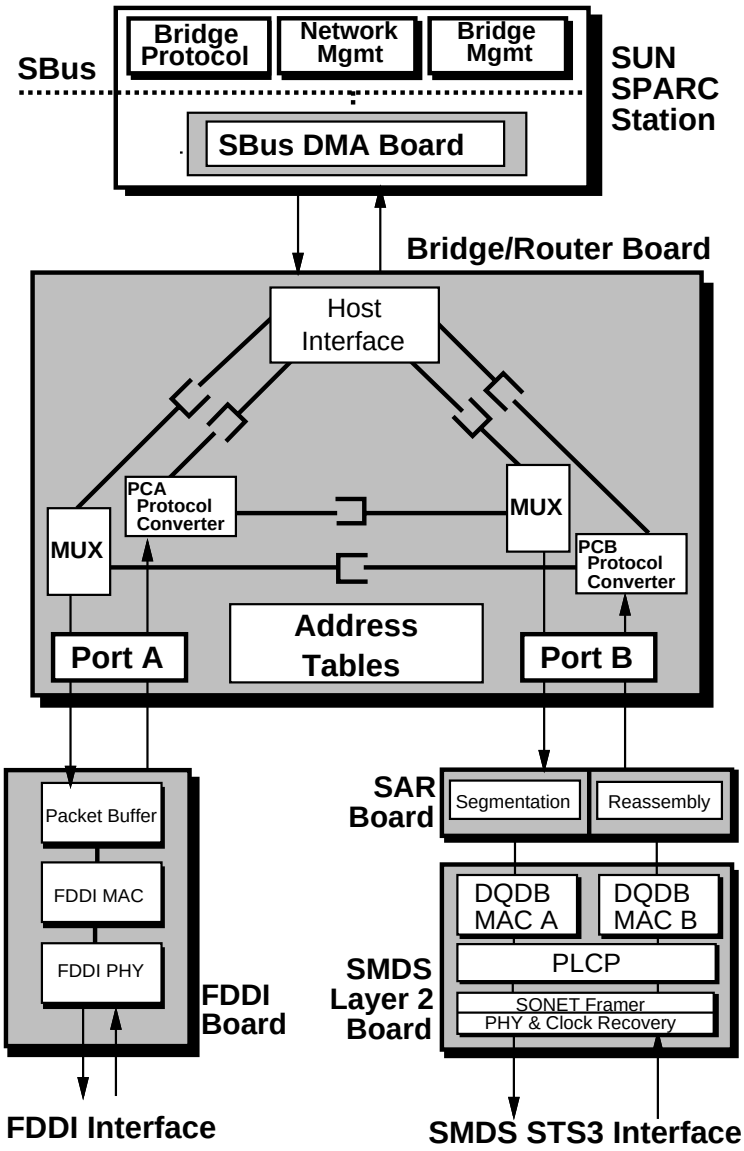


Figure 2: System Architecture showing main functional blocks. The five shaded boxes indicate each circuit board.

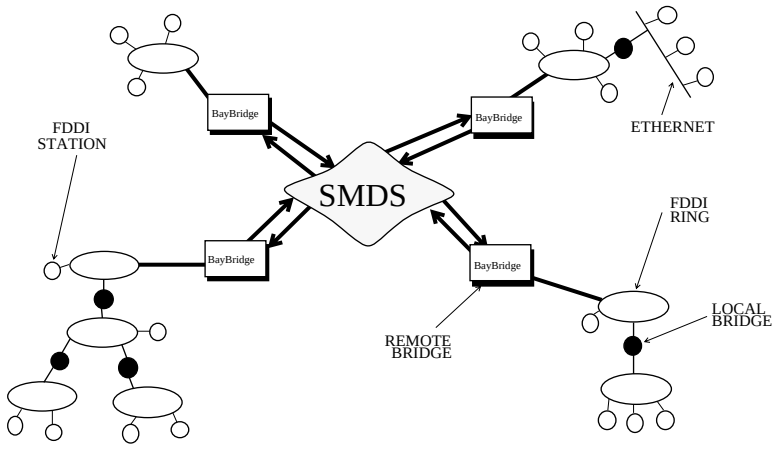


Figure 1: A typical topology interconnecting FDDI rings. Each FDDI ring may contain local bridges conforming to the IEEE 802.1D Standard.

and enables a network manager to set local and remote addresses as well as monitor network performance parameters. For routing, the host is also responsible for receiving packets of unknown destination and updating the address tables (see Section 3.1.2) and running the usual Interior and Exterior Gateway Protocols [6, 8, 7, 13]. For bridging, the workstation runs the Spanning Tree Bridge Topology algorithm[9, 10].

3 Architecture

Figure 2 is a block diagram of The BayBridge architecture [11]. The system consists of four main blocks: the Bridge/Router Board, the SMDS Interface, the FDDI Interface and the Host Interface.

The interfaces between blocks (Port A, Port B and the Host Port) are identical but *separate* busses. This eliminates the bottleneck of a single shared system bus. Each interface consists of two unidirectional data busses and a control bus for reading and writing configuration registers. Each data bus is 32-bits wide with a "Tag-bit" to delimit protocol data units, a write control line to indicat that the data is valid and a full-flag for flow control. This common interface allows the system to be configured as an FDDI-SMDS, FDDI-FDDI or SMDS-SMDS bridge/router. Alternatively, the FDDI Board or SAR Board may be connected directly to the SBus DMA card, or may be used to test each board using one or more of the SBus DMA cards.

2 System Overview

2.1 Objectives

A platform, known as The Bay Bridge[1], has been built to investigate high performance bridging and routing in hardware, based on a programmable bridge/router board connected to a host workstation. As examples of two high speed networks, FDDI and SMDS were chosen. However, the architecture is not limited to these network protocols.

The first implementation was designed with the following objectives:

- High Throughput and low per-frame[2] processing delay.
- SMDS interface at SONET STS-3c 155 Mbps and DS3 45 Mbps using custombuilt DQDB MAC chip.
- Expandable Address Tables.
- Configurable as FDDI-SMDS or FDDI-FDDI, bridge or router.
- Investigate Programmable Protocol Conversion.

2.2 Functional Overview

For bridging, the system complies with the proposed IEEE 802.1g Remote Bridge Draft Standard and the proposals of the IEEE 802.6 Multiport Bridge Committee [3, 9]. Hardware routing assist is achieved by the Protocol Converter. This is described in Section 3.1.2. When routing, the system complies with the recommendations for the transmission of IP packets over FDDI and SMDS [4, 5].

A typical application and topology is shown in Figure 1. FDDI frames are encapsulated into a single SMDS message and transmitted across the Subscriber Network Interface[3] to the Metropolitan Switching System (MSS). The SMDS message is then delivered to one or more remote Bay Bridges. A host interface provides an FDDI and SMDS interface to a local workstation. The local host runs the Network Management agent, the FDDI Station Management Process

[1] The project initially focussed on high speed bridging, but later enhanced the design to include hardware routing. Hence the name "The Bay Bridge"

[2] Throughout this paper we will use the word "frame" to describe a MAC Layer protocol data unit and the term "packet" to refer to a Network Layer protocol data unit.

[3] The Subscriber Network Interface is the DQDB protocol of IEEE 802.6

1 Introduction

As the number of high speed LANs increase, so does the need to connect these networks within a metropolitan or wide area, thus expanding the boundary of LANs beyond a building or campus. To maintain the quality of service offered by the LANs, a high performance Metropolitan Area Network (MAN) is required. The MAN used to connect the high speed LANs must aim to *maximize the throughput* and *minimize the delay per packet*. *Switched Multi-megabit Data Service (SMDS)* [1], proposed by *Bell Communications Research, Inc. (Bellcore)* and offered by *Regional Bell Operating Companies*, enables LANs to be interconnected beyond the customer premises, across a metropolitan or wide area. SMDS offers a high throughput connectionless datagram service at DS1 (1.5Mbps), DS3 (45Mbps), and STS-3c (155Mbps). LANs may be interconnected over SMDS with a MAC layer bridge or network layer router (or "gateway"). Bridging is suitable for a small number of sites within an organization. Routing is more suitable for larger networks and between different organizations.

For effective bridging and routing between a high speed LAN such as FDDI and a high speed MAN such as SMDS it is important that the bridge/router operates at high throughput and low delay. Although a high performance router between FDDI and ATM has been described elsewhere [2], the design and performance of a high speed bridge or router between FDDI and SMDS has not been reported.

The Bay Bridge was built to investigate high performance bridging and routing between FDDI and SMDS. The STS-3c rate for SMDS was chosen because it is a near match to the FDDI rate and enables high speed bridging and routing to be investigated.

In the remainder of this paper, we shall describe the architecture operation and performance of The Bay Bridge. In Section 2 we shall consider the objectives of the work and an overview of the functional operation of The Bay Bridge. We then describe the architecture of the system in Section 3 with emphasis on the *Programmable Protocol Converter*. In Section 4 we consider the throughput performance and per-packet processing time of The Bay Bridge operating as an FDDI-SMDS bridge/router and as an FDDI-FDDI bridge/router.

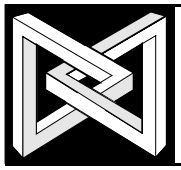
Architecture and Performance of The BayBridge: A High Speed Bridge/Router Between FDDI and SMDS

Nick McKeown Richard Edell My T Le

Abstract

The BayBridge is a high performance bridge/router designed for high throughput bridging and routing in hardware between two network ports. The first prototype may operate as an encapsulating two-port remote bridge between FDDI and SMDS. FDDI rings are interconnected via the public SMDS network operating at the SONET STS-3c or DS3 rate. High throughput is achieved by a specialized processor: *The Protocol Converter*, a programmable device for translating between network protocols and making forwarding/routing decisions.

In this paper, we first present an overview of the system followed by a description of the architecture with special emphasis on the *Protocol Converter*. Finally the performance of The Bay Bridge is described in detail, first as a bridge and then as a router operating between FDDI and SMDS or between FDDI and FDDI. With interfaces to FDDI and SMDS, The Bay Bridge has a maximum bridging throughput of over 300,000 frames per second and a maximum combined bridging/routing throughput of over 250,000 packets per second. Methods are discussed for increasing the throughput in particular applications.



The Bay Bridge

Project Report: 13 Revision: 2.0
Department of Electrical Engineering and Computer Sciences
University of California at Berkeley