Loop Concatenation and Loop Replication to Improve *Blazelan* Performance

ZYGMUNT HAAS, MEMBER, IEEE

***Abstract*—*Blazelan* is a proposed "almost-all" optical local-area network that can support Gb/s throughput with very low delay (on the order of tens of hundreds of microseconds). The basic idea behind the *Blazelan* design is the use of the fiber links as storage for packets in transit, thus eliminating the need for switch memory, simplifying the switch design, and featuring fast, "on the fly" switching. Because of the distribution of storage throughout the network, congestion control and flow control up to the network layer are inherently provided at the physical layer in *Blazelan*. Furthermore, the use of *source routing* simplifies the routing operation. Finally, because of lack of conventional memory and the simple switching node design, *Blazelan* lends itself to photonic implementation. *Blazelan* is an example of an "extended bus" local-area network that provides high-throughput low-latency communication and can be used for distributed and parallel processing systems. In this paper, we present the basic network design and introduce two techniques to increase the network capacity: loop concatenation and loop replication. We show that with the two techniques, the *Blazelan* capacity approaches that of the output queueing system.**

I. Introduction and Motivation

THE motivation behind the *Blazelan* design is to provide a local-area network that supports high-speed communication for bursty traffic. Because of the inexpensive bandwidth in the local-area environment, the major challenge is to provide low-delay delivery of packets (as opposed to wide-area, usually bandwidth limited networks, in which the major challenge is to provide high-throughput characteristics). High-performance (low-delay) communication can be implemented by sacrificing the relatively cheap bandwidth available in fiber-based local-area networks. *Blazelan* is targeted at the future computing environment, which will be characterized by parallel processing, distributed processing, and distributed databases. In such an environment, many machines operate with some degree of coupling, from the loosely coupled mode in the distributed processing case to the closely coupled mode in the parallel processing case. The communication model most suitable for such an environment is the transactional communication model[1] [1], where a *client* initiates a *request* to a *server*, and the server responds with a *response*.

Blazelan is targeted to provide large (in the Gb/s range) throughput with low delays on the order of tens to hundreds of μs in a local-area environment. The reason for the need for such low delay is that the performance of a distributed system is seriously degraded if long latency is associated with a transaction. The classical example is an atomic transaction.[2] Consider, for example, an atomic operation performed on remote machines, such as a distributed database. An update in the database needs, in general, to lock access to some information residing on more than one machine. If the locking operation of data in distant machines is slow because of the network speed, the concurrency of other transactions will be affected, and the performance of the whole system will be degraded. This is especially significant in a distributed environment, in which a transaction may involve many machines. (Moreover, if the network is slow, some of the timers associated with the transaction may expire and, as a result, additional overhead and delay may be required to determine the status of the transaction.) Thus, the low-delay requirement is crucial if high-performance communication is to be realized. Low delay is also required for real-time traffic such as voice and interactive video (for example, delay on the order of 50 ms is required for voice communication), and for remote control operation (for example, remote robotics or interactive games).

Some examples of the applications that *Blazelan* is targeted at include: large-scale distributed database systems in general, and file systems in particular, parallel processing systems, multimedia conferencing systems, medical imaging, and database of high-resolution images. These systems are characterized by the need for low-latency communication of randomly accessed large amounts of data.

With the introduction of optical fiber media (and photonic switching and processing) some of the functions of protocols become obsolete or unnecessary (error detection of the data link, for example). Of the remaining functionalities of these protocols, as much as possible should be implemented in hardware if fast networking is the goal. One possibility for hardware implementation of these protocols is to simply copy the software implementation into hardware. A more novel approach, however, is to use some of the features of the new media to provide the functionality of higher levels (flow control on the data link and

Manuscript received September 13, 1989; revised January 24, 1990. This paper was presented in part at the IEEE 6th International Workshop on Microelectronics and Photonics in Communications, New Seabury, Cape Cod, MA, June 7-9, 1989 and at GLOBECOM '89, Dallas, TX, November 27-30, 1989.

The author is with AT&T Bell Laboratories, Holmdel, NJ 07733.

IEEE Log Number 9036224.

[1]Also referred to as the request-response model.

[2]An atomic transaction is a sequence of operations whose execution is to be performed without interruption.

0733-8716/90/1000-1574$01.00 © 1990 IEEE

network layers, for example). We note that such an approach may violate the ISO-OSI model.

Blazelan is an example of a network that possesses some higher layer functionality in the physical layer; i.e., the congestion-control and the flow-control up to the network layer are provided in *Blazelan* by the physical layer. *Blazelan* is a multihop network, in which the fiber links provide temporary storage for packets in transit, recirculating the blocked packets on the fiber links that are structured as loops. Since the fiber loops are designed to fit approximately one packet, the order of packets is preserved as they travel through the network.[3] As the number of packets present in the network increases, the loops become more and more populated, allowing less and less new traffic to be inserted into the network. Thus, network congestion-control and flow-control up to the network layer are inherently performed by the subnet itself. Consequently, it is possible to eliminate some of the protocol processing overhead associated with the execution of these functions.

The difference between *Blazelan* and *Blazenet* (the wide-area network counterpart to *Blazelan* described in [2]–[4]) is that each one of the *Blazelan*'s loops is designed to accommodate a single packet, while in *Blazenet*, many packets can coexist at any time on a single loop. Because of this difference, the *Blazenet* throughput is a little bit higher than the basic *Blazelan*'s throughput.[4] The basic *Blazelan* design can be extended to include two features: loop concatenation and loop replication. Both of the features increase the total storage within the fiber links and, as shown in this paper, significantly improve the network throughput performance. In fact, by using the loop replication technique, the throughput-limited input queueing can be improved to have performance approaching that of the output queueing.

In this paper, we present the *Blazelan* design and some performance evaluation. Section II describes the *Blazelan* design and operation principles. Section III presents *Blazelan* performance. Section IV discusses some additional issues in *Blazelan* design. Conclusions are presented in Section V.

II. The Basic *Blazelan* Design and Operation

Blazelan is composed of switching nodes that are connected by point-to-point logical unidirectional links formed by the fiber loops. Network topology is unconstrained. An example of a loop is shown in Fig. 1. Each loop is a bidirectional channel, but serves unidirectional network traffic. Thus, two loops are required for a logical bidirectional link. Loop length (the roundtrip distance between the nodes) corresponds to approximately one packet,[5] (i.e., a loop is designed to store a single packet).

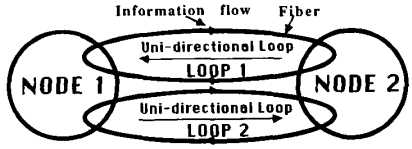

Fig. 1. *Blazelan* loop.

Thus,

$$\text{loop-length} = \frac{1}{\mu C} \cdot c \,[\text{meters}], \tag{1}$$

where $1/\mu$ is the packet size in bits, C is the transmission data rate in bits/s, and c is the speed of light in the fiber media in meters/s. For example, packets of 1000 (bits) on 1 Gb/s require loops of approximately 200 m (fiber refractive index of 1.5 was assumed). *Blazelan* is a multihop network; a packet is forwarded from a switching node to a switching node until it arrives at the destination host and is removed from the network. An example of a 16-node *Blazelan* is shown in Fig. 2. In this figure, a loop with double-sided arrow represents two loops serving traffic in opposite directions.

Several novel aspects of the design are essential to realize the *Blazelan* performance and partial implementation in photonics. First, *Blazelan* uses source routing [6] to allow fast switching and simple switching logic at each switching node. The source route is specified in each packet as a series of *hop-selects*, resulting in a packet format as shown in Fig. 3. Each *hop-select* field indicates the output link on which the packet is to be forwarded for that hop. When a packet arrives at a switching node, the first nonzero *hop-select* field in the packet is examined to determine the next output link for the packet. If that output link is available for transmission of a new packet, the *hop-select* field is zeroed[6] and the packet is immediately routed to the available output link.

When a packet is blocked (because it selects an output link that is unavailable at the packet's arrival time), the packet is routed back to the previous switching node on the return portion of the loop that the packet arrived on. Upon its arrival on the previous switching node, the returned packet is sent again to arrive at the blocking switching node one roundtrip time after its first arrival at this node. Thus, the loop effectively provides short-term storage of the packet, causing the packet to reappear at the blocking switching node a short time later. The combination of the high data rates and the decreasing cost of the fiber makes this form of storage attractive. The *loopcount* field of a packet header is decremented and examined each time a packet is returned. If the *loopcount* reaches zero,[7] the packet is removed from the network. This mechanism prevents a packet from indefinitely looping within the network under some failure or load conditions.

[3]This is in contrast with *Blazenet*, in which the packet order is not preserved.

[4]The link capacity of a *Star* topology with infinite number of inputs is 63.2% of the total link capacity for *Blazenet* and 58.6% for *Blazelan* [5].

[5]If variable packet size is used, the loop length corresponds to the maximum packet size. In this case, there is some performance degradation if the average packet size is smaller than the maximum packet size.

[6]In a practical implementation, overwriting a *hop select* with a string of ones might be a better solution.

[7]In a practical implementation, the *loopcount* may be represented as a string of ones; each time the packet is returned, one of the ones is erased.

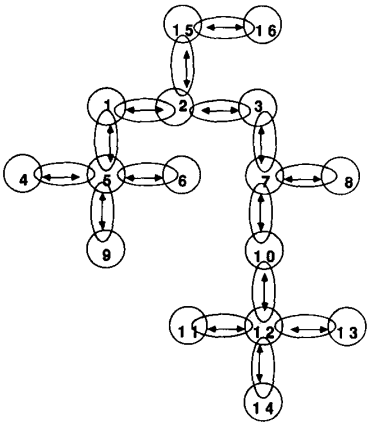

Fig. 2. An example of 16-node *Blazelan* configuration.

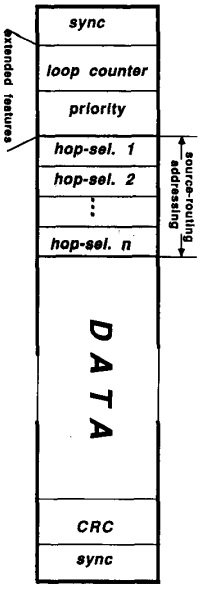

Fig. 3. *Blazelan* packet format.

This handling of blockage dramatically reduces the average packet delay through a loaded network and increases the network efficiency compared to simply dropping the packet. A dropped packet has to be retransmitted by the source after some timeout, at least one roundtrip time long. Since the probability of a packet being blocked increases with path length, as does the network investment in the blocked packet, dropping the packet seriously degrades the network performance under load for networks with a large hop span. Moreover, the *Blazelan* approach does not require memory in the switching node of the size and speed required to store all blocked packets, such as would be needed for a conventional store-and-forward design. Memory operating at 1 Gb/s would significantly increase the cost of the switching nodes and make their realization in optics unfeasible, at least in the near future. Finally, the loopback technique exerts back pressure on the link over which the packet was received, because each returned packet makes the loop less available for new packets to be forwarded on it. In the extreme, this back pressure extends back from the point of contention to one or more packet sources. Besides alerting the packet source of congestion, the back pressure provides fast feedback to the source routing mechanism, allowing it to react quickly to network load and topological changes.

The simple logic required for hop selection makes it feasible to implement the switching function in photonics and perform switching operation at gigabit data rates. (It is suggested that, with today's state-of-the-art technology, the control is built in electronics. As the photonic processing matures, the electronic control may be replaced by its photonic counterpart.) Also, the switching delay is limited to the time required to interpret the packet header, check the availability of the output link, and perform the actual switching operation (if the output link is available).

If the distance between the nodes exceeds half the loop length, two or more loops are physically concatenated by a simple switching node with one input and one output loop. If the internodal distance is shorter than half the loop length, the excessive length of the loop can be compactly stored in the switching node.

Fig. 4 shows the block design of a *Blazelan* switching node. For simplicity, only one *Input Loop* and one *Output Loop* are shown. In practice, each link connected to the switching node has one *Input Loop* and one *Output Loop* that corresponds to *Loop 1* and *Loop 2* in Fig. 1. Handling of local I/O is explained later.

Each *Input Loop* terminates with a *packet detector* circuit and a *switching delay*. The *packet detector* circuit scans the input and looks for syncs. Upon detection of a sync, the *packet detector* raises the *new-packet* signal and reports packet arrival to the *Control*.[8] *Switching delay* consists of a piece of fiber, which is long enough to contain the packet header and the number of bits corresponding to the time the control logic requires to do the actual switching.

Each *Output Loop* terminates with a *packet-detector* circuit that scans the *Output Loop* looking for a returned packet. Upon detection of a returned packet, the *returned-packet* signal is raised. A loop is considered to be free if it does not contain a packet or any part of a packet. The *Control* can uniquely determine the status of a loop using the *returned-packet* signal and a timer that can be set for a packet length.[9]

The switches that switch a packet from an input to an output loop can be implemented in a switching matrix configuration (such as Lithium-Niobate switches). How-

[8]The *Control* is not shown in the figures for simplicity.

[9]Use of precise and fast timers can be avoided by sending a signal back to the previous node that indicates a packet leaving the loop. Such a signal can be sent over the reverse portion of the loop, and can contain a truncated header with a proper indication.

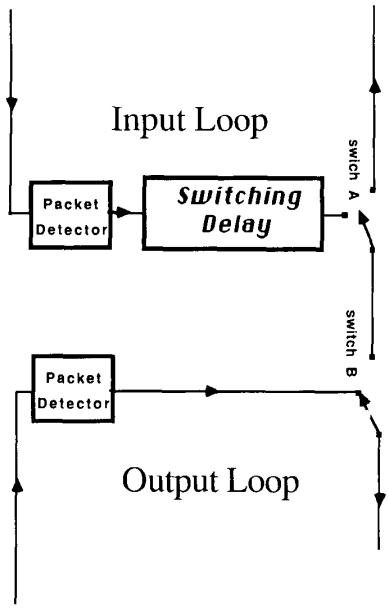

Fig. 4. The switching node design.

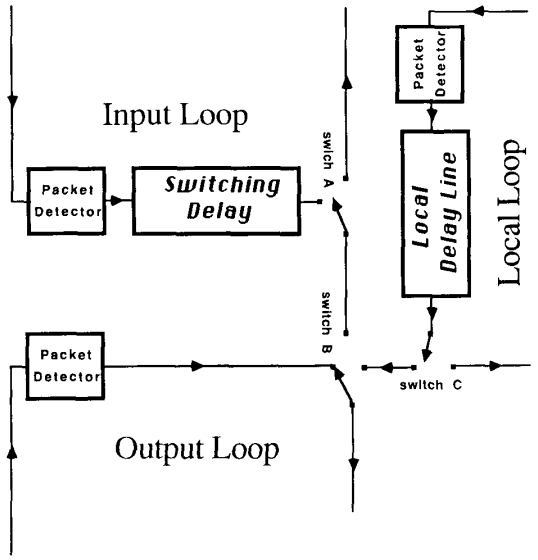

Fig. 5. Switching of local traffic.

ever, for a large number of inputs/outputs, this approach might be too expensive. In such a case, an alternate switch architecture can be used.

When a packet is to be forwarded from an *Input Loop* to an *Output Loop*, the availability of the *Output Loop* is checked. An *Output Loop* is available if no active connection of any other loop to this loop already exists and if the loop is not occupied by a returned packet. The condition that a loop is not occupied by a returned packet can be determined by the timer that is triggered by the sync of every transmitted packet. (As mentioned, a simpler implementation is possible, in which a packet sync with proper indication whether a packet is removed from the loop or not is always returned.) If the *Output Loop* is available, the packet is clocked from its *Input Loop* onto the *Output Loop* by properly setting switches *A* and *B*. If, on the other hand, the *Output Loop* is busy, the packet is blocked and is returned to its previous node by being clocked out on the same *Input Loop* it came on. (Switch *A* has to be properly operated.) In the case that more than one packet tries to enter a specific *Output Loop*, only one packet wins (the one with the higher priority,[10] or one chosen randomly in the case of equal priorities), and the other(s) are clocked out on their loops. The *Input Loop* terminates on the previous node, where it is referred to as an *Output Loop*. Upon arrival of a blocked packet at the previous node, it sets the timer to indicate that the *Loop* is busy and is clocked out to return at the blocking node one roundtrip later. A *Blazelan* switching node can be implemented as a simple interconnection of a number of photonic components, assuming the availability of fast-switching devices capable of switching within a small fraction of the duration of a header bit once the switching command has been initiated. Devices operating at rate of at least 3 GHz exist. Slower devices can be employed for lower cost by maintaining an adequate interpacket gap.

Local I/O is handled in a similar way to the forwarded traffic (see Fig. 5). A local delay line, of the length of packet size, serves as an insert register. When the current message is inserted into the network and empties the local delay line, a new message from the local host can be input to the local delay line.

Blazelan can be designed as a synchronous or asynchronous network. In the synchronous configuration, all the packets arriving at a switching node are assumed to be in phase.[11] In the asynchronous case, packets can arrive at a switch at any time. The synchronous network has slightly better performance at the expense of the need to provide packet synchronization.[12] Also, a switch operating synchronously may have simpler design.

Previous reports [2]–[4] provide a detailed description of *Blazenet*, which is the wide-area counterpart of *Blazelan*.[13] The performance analysis of the *Blaze* approach presented in the above reports indicates that the *Blazenet* delay penalty for lack of buffering is minimal, especially for low-load operation. We now proceed to present some additional throughput performance figures.

III. Performance Evaluation

The performance evaluation presented here was obtained for synchronous *Blazelan*, by simulation. (Alter-

[10]See Section IV.

[11]In order to satisfy this requirement, there may be the need to insert an additional loop on some network paths.

[12]Synchronization is impractical for wide-area *Blazenet*. However, in a local environment, one may find synchronization to be quite an attractive alternative.

[13]The main difference between *Blazelan* and *Blazenet* is that, because of the short length of loops in *Blazelan*, a single packet can be stored in a loop. This simplifies the switching node design.

natively, analytical methods that enable one to reach the same conclusions, as well as a rough estimate of *Blazelan* performance, can be used by neglecting the time dependence of the subsequent arrival of a blocked packet. In other words, if it is assumed that the probability of a successful forwarding through a switch is independent of the number of previous attempts, a simple analytical formula for throughput can be composed. Solution of such a formula results in optimistic throughput evaluation. From our experience, the error is about 10% in most cases.) The switch in the basic *Blazelan* design is an input-queueing system.[14] As shown in [5], the maximum throughput in an input-queueing switch is limited to 58.6% (63.2%, if input dependency is neglected). Unfortunately, the capacity of the basic *Blazelan* design is lower than that; it is approximately 33% (in the case of a very long path and a switch with large number of inputs/outputs). The reason for this reduction in *Blazelan* throughput is the fact that a packet on an input loop can be blocked not only by another input forwarding a packet to the same output loop, but also by a packet stored in the output loop and blocked by the next switch in line. This is illustrated in Fig. 6. However, as shown in this section, *Blazelan* throughput can be significantly improved by two design variations: loop concatenation and loop replication.

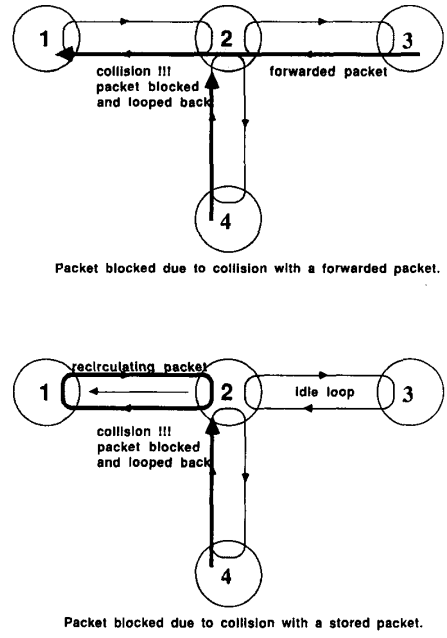

Fig. 6. Scenarios for packet blockage.

A. Loop Concatenation

Loop concatenation is done by connecting several loops in tandem by a simple one-input–one-output switching node. The switching node can, optionally, regenerate the signal, if the distance between switching nodes is sufficiently large. For an example of concatenated loops, see Fig. 7. Loop concatenation has two purposes: to connect nodes separated by a distance longer than a single loop length, and to increase the network throughput. This increase in the network throughput is obtained by reducing the *occupancy*[15] of the loops emerging from a *Blazelan* switching node. In other words, the effect of the mechanism[16] that brings down the *Blazelan* throughput, from the theoretical 58.6% of the input-queueing system to 33%, is reduced. Since a switch is a throughput "bottleneck," the closer a loop is to the next switch, the higher the occupancy[17] the loop experiences. Reduction of the occupancy of a loop located immediately after a switch lowers the probability of a packet blockage due to the loop being busy.[18] In the limit, when the number of concatenated loops increases so that the probability of a packet being blocked due to a busy output loop is negligible, the capacity of the system can be restored to that of the input-queueing system.

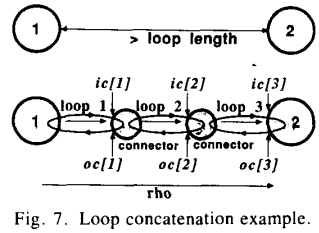

Fig. 7. Loop concatenation example.

In this subsection, we demonstrate the impact of increasing the number of concatenated loops on the capacity[19] of the system. We need to differentiate between two cases: single hop and inner hop.[20] The capacity of a single hop is shown in Fig. 8 as a function of *conditional output-removal probability*, and with the number of concatenated loops per single hop n as a parameter. The conditional output-removal probability is the probability that the sink will *not* remove a packet from the loop, given that a packet on the loop exists. Conditional output-removal probability represents the sink blocking probability. The parameter n varies from 1 to 5. In the curves in the graph in Fig. 8, it was assumed that the source is always ready to transmit a packet. Graphs in Figs. 9 and 10 illustrate how the curves change when the *conditional in-*

[14]In an input-queueing system, packets are stored before they are switched. In an output-queueing system, the buffer is located after the switching fabric.

[15]The term *occupancy* is used here to indicate the fraction of time a loop is occupied with a packet. It is assumed that all the processes are ergodic; thus, occupancy equals the probability of a loop being busy.

[16]This mechanism is the second scenario in Fig. 6.

[17]Occupancies of loop k at the loop's input and output are labeled $ic[k]$ and $oc[k]$, respectively.

[18]This phenomenon is similar to flow control in a network with finite buffers.

[19]Capacity = maximum throughput.

[20]The term *hop* refers to a switching node and all the loops emerging from this switching node. A single hop is a hop that connects a source and a sink, i.e., a single switching node between a source and a sink. An inner hop is a hop that is located "far" from any source or sink in the network.

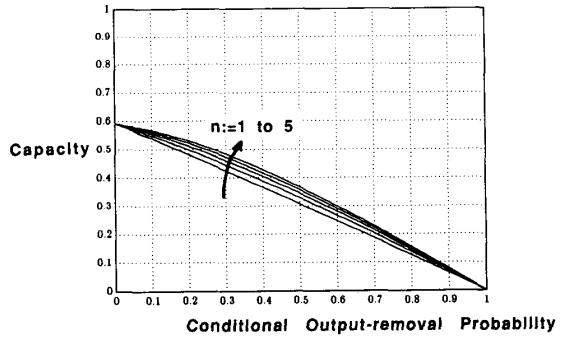

Fig. 8. Capacity of a single hop with concatenated loops, conditional insertion probability = 1.

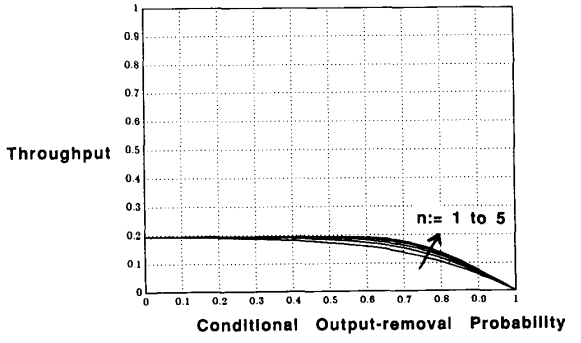

Fig. 9. Throughput of a single hop with concatenated loops; conditional insertion probability = 0.2.

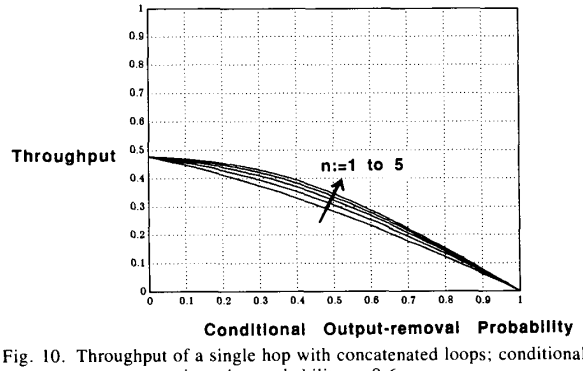

Fig. 10. Throughput of a single hop with concatenated loops; conditional insertion probability = 0.6.

put-insertion probability[21] of a source is smaller than 1, i.e., 0.2 and 0.6, respectively. All these graphs were obtained by a simulation. The graph that shows the effect of loop concatenation for an inner hop is presented in Fig. 11. In this case, the capacity is shown as a function of the actual (not conditioned) *Input-Loop Occupancy*[22] of the loop entering the switch, $ic[n]$ ($ic[3]$, for the example in Fig. 7).

[21]*Conditional input-insertion probability* is the probability that a source has a packet to transmit, given that the input loop is free.

[22]The *input-loop occupancy* is the fraction of time the loop entering the switch is busy.

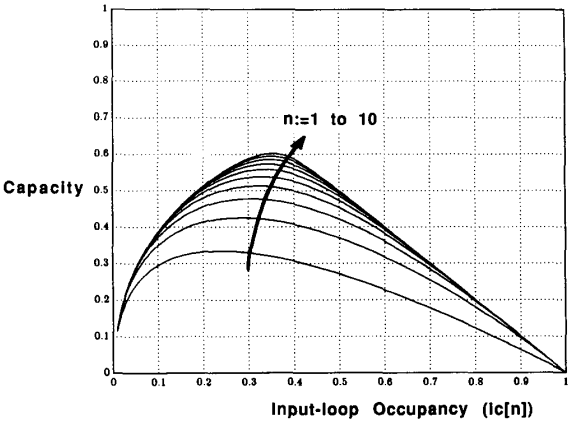

Fig. 11. Capacity of an inner hop for concatenated loops.

We draw the following conclusions from the above graphs.

- By using the concatenated loops, the *Blazelan* capacity can be restored to the value obtainable by a basic input-queueing system.
- For an inner loop, most of the improvement in capacity is achieved with $n = 5$. Further increase in n results in improvement smaller than 2%.
- The throughput improvement by loop concatenation technique for a single hop is not as dramatic as for the inner-loop case.[23] Also, in this case, increase in n above 5 yields only marginal improvement.
- As the conditional input insertion probability decreases, the gain in throughput also decreases.

B. Loop Replication

Loop replication is used as a way to reduce the input-queueing effect in *Blazelan*. In this technique, loops near the switches are replicated,[24] thus reducing the head-of-the-queue blocking phenomenon. For an example of replicated loops see Fig. 12. The replication effectively increases the amount of available (fiber-based) storage closer to the switch and, if properly designed,[25] should preserve the packet order. The required replication factor m to achieve some level of performance depends on the traffic statistics and on the network topology.

The graph in Fig. 14 shows the impressive improvement of the network capacity as a function of the loop multiplicity m. The graph was obtained for a *Blazelan* configuration presented in Fig. 13. (In this figure, a letter in the upper portion and in the lower portion of a switching node indicate the number of inputs and the number of outputs, respectively. The results in Fig. 14 were obtained for $i = m = j = 10$.) A graph, showing similar improvement for a single loop, is shown in Fig. 15. In

[23]This may suggest that the loop concatenation technique should be used for networks with long paths.

[24]The replicated loops are referred to here as *subloops*.

[25]In order to preserve the packet order, packets that are destined for the same output of the switch should be forwarded to the same subloop.

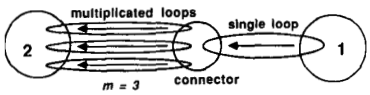

Fig. 12. An example of loop replication with factor $m = 3$.

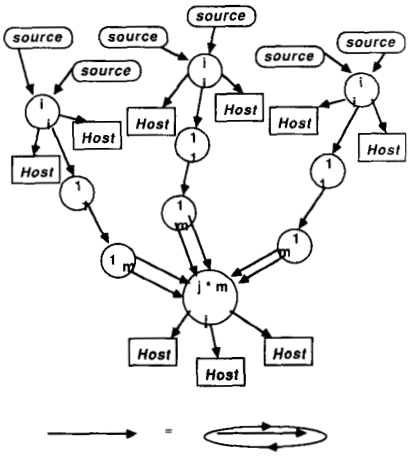

Fig. 13. *Blazelan* configuration for evaluation of inner loop throughput improvement.

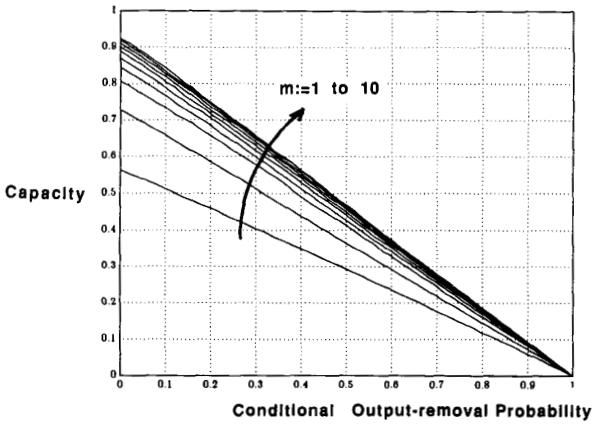

Fig. 14. Capacity of an inner loop *Blazelan* with replicated loops.

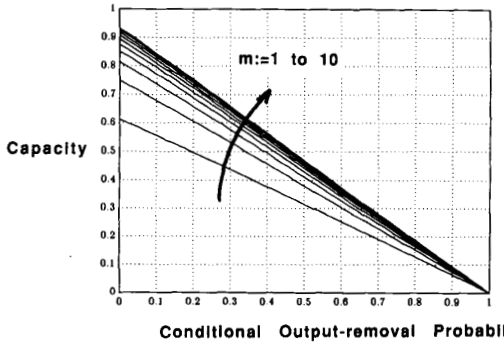

Fig. 15. Capacity of a single loop *Blazelan* with replicated loops.

both the graphs, it was assumed that the conditional input insertion probability is 1. The single loop case was evaluated for a hop with two concatenated loops ($n = 2$), and the loop replication was performed on the second concatenated loop only.

One can draw the following conclusions from the graphs.

- By using the loop replication technique, the capacity of *Blazelan* can be dramatically improved, approaching that of the output queueing system.
- Most of the dramatic improvement in capacity of an inner loop is achieved for $m \leq 5$. Larger values of m yield only marginal improvement.
- Likewise, the improvement of the single hop capacity is most significant for m smaller than or equal to 5.

The results discussed in the last two sections indicate that *Blazelan* capacity is not a limitation on the network performance.

IV. Additional Issues

A. Extended Features

Some additional features that can be incorporated into the *Blazelan* design include: priorities, time-stamping, and multicast. These are performed by including specific fields in the packet format displayed in Fig. 3. The switching node design needs to undergo some minor changes in order to accommodate the above features (see [4] for details).

B. Boundary Between Photonics and Electronics

Optical switching and processing of optical transmission open new dimensions in future networking. Photonic implementation, as opposed to a conventional electronic implementation, offers increased switching speeds [7], [8]. In addition, a network built out of optical components is less susceptible to electromagnetic interference and electromagnetic pulse and provides more secure transmission. Unfortunately, the state of the art of photonic processing is still in its infancy. Large and fast memory, in particular, appears to be a difficult component to realize photonically. However, with the progress in photonic technology, processing in light of more and more functions becomes available. Consequently, a simple network node design is of great importance, if and when the state of the art of photonic processing advances to such a degree that such full photonic implementation will be possible. *Blazelan* switching node design has limited functionality (i.e., no routing and no flow-control) and thereby lends itself more toward photonic implementation when it becomes feasible. Below, some speculations are given on a possible full photonic implementation in the future. In the future full photonic implementation of *Blazelan*, the detection of fields in the packet format (such as the *sync* or the *hop-selects* field) can be done by the optical delay-line signal processing [9], [10], setting/resetting/zeroing of fields in the packet (such as the *hop-selects*) can be performed by modulating a fast switch to either transfer the

input information or to override some of its fields, and information processing and computing can be done by photonic logic [11]–[14]. Note that there is little memory needed in the *Blazelan* switching node design; the control is composed mainly of logic. Signal regeneration (after amplification) can be done by an all optical regenerator [15].

V. Concluding Remarks

The demand for high-speed communication results in the necessity to replace slow communication software by fast hardware. However, a simple replacement of a software by a hardware implementation might not be an adequate solution in applications which require very low latency. We need to look for hardware implementations of communication networks that inherently possess higher layer functionality. *Blazelan* has been proposed as such a local-area network, in which the congestion-control and flow-control on the network and data link layers are built into its physical layer. Moreover, *Blazelan* avoids use of large and very-fast buffering, performing switching by the *hot potato*[26] switching scheme. Thus, *Blazelan* lends itself to photonic implementation. *Blazelan*, by providing multi-Gb/s throughput with low delays, can be the local-area network for future distributed and parallel processing environments.

References

[1] D. R. Cheriton and C. L. Williamson, "VMTP as the transport layer for high-performance distributed systems," *IEEE Commun. Mag.*, vol. 27, June 1989.

[2] Z. Haas and D. R. Cheriton, "*Blazenet:* A high-performance wide-area packet-switched network using optical fibers," Dep. Comput. Sci., Stanford Univ., Tech. Rep. STAN-CS-87-1185, Oct. 1987. Also, "*Blazenet:* A packet-switched wide-area network with photonic data path," to appear in *IEEE Trans. Commun.*

[3] —, "A case for packet-switching in high-performance wide-area networks," in *Proc. SIGCOMM '87 Workshop*, Stowe, VT, Aug. 11–13, 1987.

[4] Z. Haas, "Packet-switching in future fiber-optic, wide-area networks," Ph.D. dissertation, Elect. Eng. Dep., Stanford Univ., May 1988.

[5] M. J. Karol, M. G. Hluchyj, and S. P. Morgan, "Input versus output queueing on a space-division packet switch," *IEEE Trans. Commun.*, vol. COM-35, Dec. 1987.

[6] V. Singh, "The design of a routing service for campus-wide Internet transport," M.Sc. thesis, Lab. Comput. Sci., M.I.T., MIT/LCS/TR-270, Aug. 1981.

[7] P. R. Prucnal, D. J. Blumenthal, and P. A. Perrier, "Photonic switch with optically self-routed bit switching," *IEEE Commun. Mag.*, vol. 25, May 1987.

[8] F. Guterl and G. Zorpette, "Fiber optics: Poised to displace satellites," *IEEE Spectrum*, Aug. 1985.

[9] K. P. Jackson, S. A. Newton, B. Moslehi, M. Tur, C. Chapin Cutler, J. W. Goodman, and H. J. Shaw, "Optical fiber delay-line signal processing," *IEEE Trans. Microwave Theory Tech.*, vol. MTT-33, Mar. 1985.

[10] B. Moslehi, J. W. Goodman, M. Tur, and H. J. Shaw, "Fiber-optic lattice signal processing," *Proc. IEEE*, vol. 72, July 1984.

[11] T. E. Bell, "Optical computing: A field in flux," *IEEE Spectrum*, vol. 23, Aug. 1986.

[12] *Proc. IEEE*, Special Issue on Optical Computing, vol. 72, July 1984.

[13] S. D. Smith, "An introduction to optically bistable devices and photonic logic," *Phil. Trans. Roy. Soc. Lond.*, Great Britain, A-000-000, 1984.

[14] A. L. Lentine, D. A. B. Miller, J. E. Henry, and J. E. Cunningham, "Photonic ring counter differential logic gate using the symmetric self-electrooptic effect device," in *Proc. CLEO*, Anaheim, CA, Apr. 25–29, 1988.

[15] M. Jinno and T. Matsumoto, "All-optical timing extraction using a 1.5 μm self-pulsating multielectrode DFB LD," *Electron. Lett.*, vol. 24, no. 23, Nov. 1988.

[26] *Hot potato* switching scheme does not require buffers in the switching nodes. An arriving packet that cannot be forwarded to the requested output is switched to another, randomly chosen output.

Zygmunt Haas (S'84–S'86–M'88) received the B.Sc. degree in electrical engineering from the Technion, Israel, in 1979 and the M.Sc. degree in electrical engineering from Tel-Aviv University, Israel, in 1985, both summa cum laude; he received the Ph.D. degree from Stanford University, Stanford, CA, in 1988.

From 1979 to 1985, he worked for the Government of Israel. He joined AT&T Bell Laboratories, Holmdel, NJ, in 1988, where he is now a Member of Technical Staff in the Network Systems Research Department. His interests include high-speed communication, high-speed protocols, lightwave networks, and traffic integration.