Two fiber-ribbon ring networks for parallel and distributed computing systems

Magnus Jonsson, MEMBER SPIE
Halmstad University
Center for Computer Systems Architecture
Box 823
S-301 18 Halmstad
Sweden
E-mail: magnus.jonsson@cca.hh.se

Abstract. Ring networks made of fiber-ribbon point-to-point links are proposed. The first network is a control-channel based network in which one fiber in each link joins with others to form a control-channel ring. This ring improves performance of the network by sending medium access control information immediately before the data transmissions. High throughputs can be achieved in the network due to pipelining, i.e., several packets can travel through the network simultaneously but in different segments of the ring. The network can meet tough performance demands in, e.g., massively parallel signal processing systems, which is shown by example. Also, real-time demands can be met using slot reserving. The network, called CC-FPR (control-channel based fiber-ribbon pipeline ring), can be built today using off-the-shelf fiber optic components. The increasingly good price/performance ratio for fiber-ribbon links indicates a high potential for the success of the proposed kind of networks; a prototype is currently under development. The second network is similar to first except that it divides the network into two subnetworks, one for packet-switched traffic and one for circuit-switched traffic. When the main data flow in the network does not change rapidly, this is a good choice for a simple but powerful network. [S0091-3286(98)00312-2]

Subject terms: fiber-optic communication; ring network; fiber ribbon; parallel processing; real-time communication.

Paper FIB-03 received Apr. 10, 1998; accepted for publication June 1, 1998.

1 Introduction

Parallel and distributed computing systems place increasingly higher demands on the networks that interconnect their processors or processing nodes. Fiber optic networks are foreseen to be a natural choice in such systems in the future, especially when the nodes are physically separated. With colleagues I have presented, in earlier papers, computing systems with computational modules that function as stand alone single instruction, multiple data SIMD stream computers, in which all processing elements work together closely. We use these computational modules as building blocks when building larger, highly parallel computer systems. At the global level, these systems are *multiple SIMD* computers with one instruction stream per computational module. The systems have been developed as part of a joint project between Halmstad University, Ericsson Microwave Systems AB, and Chalmers University of Technology.

Application examples are future radar signal processing systems, distributed multimedia systems, satellite imaging and other image processing methods. A typical example is the radar signal processing system described in Refs. 1 and 2. Often, these systems are classified as real-time computer systems. In a real-time computer system, correct function depends both on the time at which a result is produced and on its accuracy.[3] In many real-time systems, timing must be guaranteed to avoid life-threatening situations. An example is control systems for nuclear power plants. Other real-time systems include those for flight control, radar, robotics, and industrial control. In distributed real-time systems, the interconnection network is a very important part of the computer system. Often, guaranteeing real-time services is much more important in these systems than performance, e.g., average latency.

Since each module itself can have a sustained data output rate of several gigabits per second, a powerful interconnection network is required. In 1996, we presented a wavelength division multiplexing (WDM) star network for high-performance distributed real-time systems and analyzed how it functions in a massively parallel radar signal processing system.[1] Although the WDM star architecture is very attractive and scales well to hundreds of nodes when configured as a star-of-stars network, systems that require only a few tens of nodes can alternatively, and less expensively, be realized using optical fiber-ribbon links. Fiber-ribbon links offering an aggregate bandwidth of several gigabits per second have already reached the market.[4] The price performance ratio is very promising

In this paper, the two ring networks presented are suitable for different situations and both of them are based on fiber-ribbon links. We describe these two networks and, as a case study, show how the data flow in a massively parallel radar signal processing system is mapped on the networks.

The proposed networks are pipeline ring networks based on optical fiber-ribbon point-to-point links. In a pipeline ring network, several packets can travel through the network simultaneously, thus achieving an aggregate through-

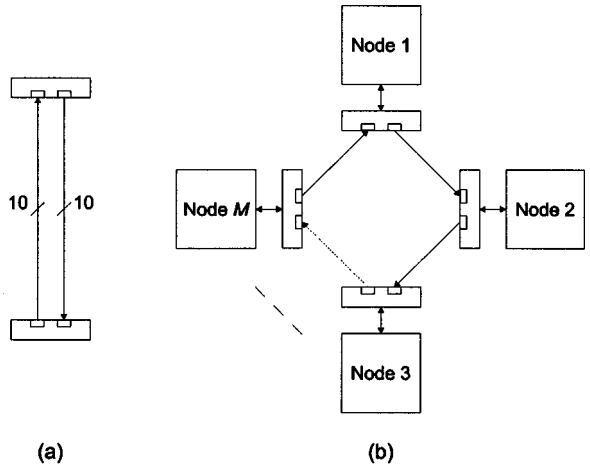


Fig. 1 (a) Bidirectional fiber-ribbon link and (b) unidirectional ring network built with $M/2$ bidirectional links.

put higher than the capacity of a single link. Motorola OPTOBUS™ bidirectional links[5] with 10 fibers per direction are used, but the links are arranged in a unidirectional ring architecture (Fig. 1), where only $M/2$ bidirectional links are required to close a ring of M nodes (assuming that M is an even number).

The first network is called the control-channel based fiber-ribbon pipeline ring (CC-FPR). The physical ring network is divided into two rings: a data ring and a control ring. In each fiber-ribbon link, eight fibers carry data and one fiber is used to clock the data, byte for byte. Together, these fibers form a data channel that carries data packets. The access is divided into slots as in an ordinary time division multiple access (TDMA) network. The tenth fiber is dedicated to bit-serial transmission of control packets that are used for the arbitration of data transmission in each slot. The clock signal, on the dedicated clock fiber, that is used to clock data also clocks each bit in the control packets.

The node synchronization requirement is more relaxed than for a traditional TDMA network and the network is somewhat similar to a slotted ring network (but without the requirement of a central controller). This is because the access to the network circulates among the nodes according to the physical order of the nodes in the ring. In addition, the ring can dynamically (for each slot) be partitioned into segments to obtain a pipeline ring network where several transmissions can take place simultaneously. Even simultaneous multicast transmissions are possible when the multicast segments do not overlap. Also, slot reserving is used to obtain guaranteed bandwidth in real-time computer systems.

Other high-performance ring networks include the WDM passive ring[6] and the hierarchical WDM ring,[7] which are more closely related to the WDM star network and star-of-stars network that were proposed in Refs. 8 and 1. Other pipeline ring networks are described in Refs. 9, 10, and 11 and more references are available in Ref. 9. Advantages of the CC-FPR network over these other networks include the use of high-bandwidth fiber-ribbon links and the close relation between a dedicated control channel and a data channel without disturbing the flow of data packets. In other words, control and data are overlapped in time.

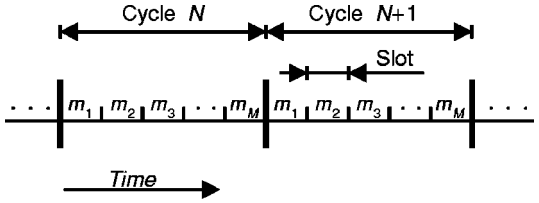


Fig. 2 Role of being slot-initiator is cyclically repeated. Each of the M nodes is the slot-initiator in one slot per cycle.

With less header overhead in the data packets the slot length can be shortened, to reduce latency, without sacrificing too much in bandwidth utilization. Also, the separate clock and control fibers simplify the transceiver hardware implementation; this is shown, among other things, by a prototype now being developed.

The network described by Jafari et al. also relies on a separate control channel but requires a central control node that brings additional cost in hardware and in latency when waiting for response from the central control node.[10] The CC-FPR network is insensitive to propagation delay in the sense that no feedback is required from other nodes or from a central controller between control-packet and data-packet transmissions.

The physical ring of the second network is subdivided into two networks, which carry different kinds of traffic. Nine of the fibers are used for time multiplexed circuit-switched traffic, eight fibers are for data and one for clocking. The tenth fiber is dedicated to packet-switched traffic using, for example, a token ring protocol. This fiber also carries control messages to reconfigure the TDMA schedule, (i.e., circuit establishment) for the other nine fibers. This network is a good choice when the main data flow in the network does not change rapidly.

The rest of the paper is organized as follows. The CC-FPR network is presented in Section 2. In Section 3, the network supporting both packet and circuit switched traffic is described. Then, in Section 4, implementation aspects are discussed, while a case study is offered in Section 5 to show the efficiency of the networks. This is followed by conclusions in Section 6.

2 CC-FPR Network

The CC-FPR protocol is described in the first subsection. Then, in Section 2.2, performance aspects related to protocol implementation are discussed. Throughout these two subsections, it is assumed that slot reserving, as described in Section 2.3, is not used.

2.1 *CC-FPR Protocol*

Before the CC-FPR protocol arbitration mechanism is explained, a description of how data packets travel on the ring is given. The access to the network is cyclic; each cycle consists of M time slots, where M is the number of nodes. Each node is denoted m_i, $1 \leq i \leq M$. In each slot, one node is always responsible for initiating the traffic around the ring. This node is called the slot initiator. Each node is slot-initiator in one slot per cycle, as shown in Fig. 2. At the end of the slot, the role of being slot-initiator is a syn-

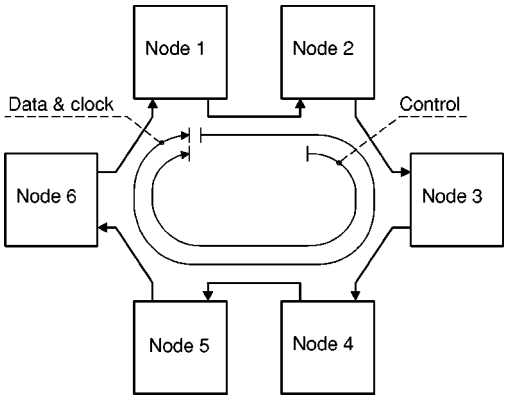


Fig. 3 Node succeeding the slot-initiator initiates the control-packet transmission.

chronously handed over to the next node downstream. This can be done implicitly simply by sensing the end of the slot, i.e., the last bit.

The CC-FPR medium access protocol is based on the use of a control packet that, for each slot, travels almost one round (over $M-1$ links) on the control-channel ring, as shown in Fig. 3. The node that will be the slot-initiator in the next slot initiates the transmission of the control packet, as shown in the figure. In the time domain, the control packet always travels around the ring in the time slot preceding the one for which it controls the arbitration (see Fig. 4). Accordingly, the control packet always passes each node one time slot before the data packet to which it is related.

The contents of the control packet are shown in Fig. 5. The control packet consist of a start bit followed by an M bit long link-reservation field and an M bit long destination field, where M is the number of nodes. Each bit in the link-reservation field tells whether the corresponding link is reserved for transmission in the next slot. In the same way, each bit in the destination field tells whether the corresponding node has a data-packet destined to it in the next slot. Additional information, such as node insertion, could also be included in the control packet; for clarity, this is not shown in the figure.

Each node succeeding the slot-initiator checks the control-packet as it passes to determine (1) if it will receive a data packet in the next slot, which is indicated by the node's bit in the destination field, and (2) if a data packet will pass the node in the next slot, which is indicated by the bit in the link-reservation field corresponding to the outgoing link of the node. If no data packet is to pass the node, i.e., the rest of the ring back to the slot-initiator is free, then the node can transmit a data packet in the next time slot in this part of the ring.

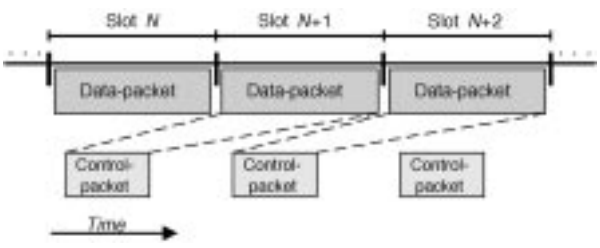


Fig. 4 In each slot, a node passes/transmits one control packet and one data packet, where the control packet is used for the arbitration of the next slot.

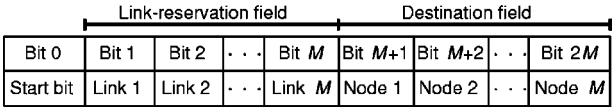


Fig. 5 Control packet contains a start bit, a link-reservation field, and a destination field.

When a node has a packet ready for transmission, it prepares, in advance, new link-reservation and destination fields to reserve needed links and notify destination node(s). In this way, the node can immediately change the control packet when it passes, provided the bit in the link-reservation field, corresponding to the outgoing link of the node, is set to zero. Since there is no data packet that will pass the node, succeeding nodes have no use for the overwritten information in the control packet.

Because all of the nodes succeeding the slot-initiator repeat the procedure of checking the control packet, multiple transmissions in different segments of the ring could occur in the same slot. An example of how the control packets travel around a five-node network is shown in Fig. 6. The arbitration results in two concurrent data-packet transmissions in the next slot, one single-destination and one multicast packet, as shown in Fig. 7. Node m_1 is the slot-initiator in the example; therefore it initiates the control-packet transmission described in Fig. 6. It reserves link 1 and link 2 for transmission to node m_3 and informs this node by setting the corresponding bits in the link-reservation field and the destination field, respectively, that it will have a data packet destined to it in the next slot. While node m_2 and node m_3 do not change the control packet, they check it to see if there will be any data packets destined to them in the next slot. Node m_4 reserves link 4 and link 5 for a multicast transmission to node m_5 and node m_1. Node m_5 then receives the control packet and removes it from the ring.

The reason the control packet travels only among the first $M-1$ links after the slot-initiator is that the clock signal is interrupted before the last link to avoid interfering with itself (see Fig. 3). The node that initiated the transmission of the control packet does not return the packet. Con-

	Outgoing control-packet		
Node	Link	Dest.	Transmission allocated
1	1 1 0 0 0	0 0 1 0 0	To Node 3
2	1 1 0 0 0	0 0 1 0 0	Could not allocate
3	1 1 0 0 0	0 0 1 0 0	Could allocate transmission to Nodes 4, 5, and 1 but had nothing to send
4	0 0 0 1 1	1 0 0 0 1	Multicast to Node 5 and 1
5	0 0 0 1 1	1 0 0 0 1	Could not allocate

Fig. 6 Control packet travels around a network with five nodes. Node 1 is the slot-initiator.

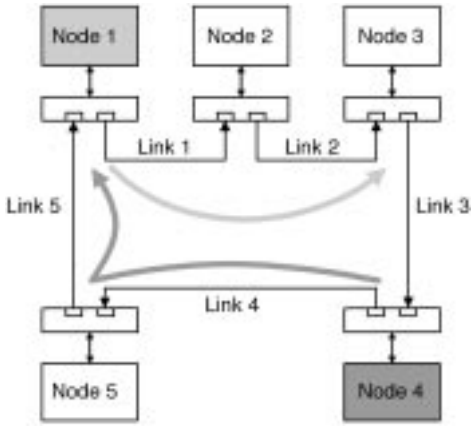


Fig. 7 Example: node 1 sends a single-destination packet to node 3, while node 4 sends a multicast packet to node 5 and node 1.

sequently, it will not be informed of whether or not there is a data packet destined to it in the next slot. However, the node will receive either a packet destined to the node or an empty packet.

Each transmitter has $M-1$ queues, one for each possible destination (the node itself excluded). When a multicast packet arrives for queuing, it is put in the queue corresponding to the multicast destination farthest away from the source node downstream. In this way, multicast packets are treated in the same way as single-destination packets and multiple multicast packets can travel in the network at the same time whenever possible.

2.2 *Performance Aspects*

It is essential to desirable performance that the delay of the control packet in each node it bypasses be minimal, especially in large networks. One method is to organize the bits in the link-reservation field in the control packet, for each slot, so that they appear in the same order as the control packet travels. In other words, the first bit corresponds to the outgoing link from the slot-initiator. Thus, when a node wants to change the contents of a control packet, it does not have to store the whole packet before checking and possibly overwriting it. Instead it can retransmit the packet bit by bit and exchange the remaining part of the packet (if transmission is possible) after reading the bit in the link-reservation field corresponding to its outgoing link. The node's bit in the destination field in the incoming control packet must, however, be checked before it is thrown away. Using this method, the delay in each node can be reduced to only one or a few bits.

As indicated in Fig. 8, the bandwidth utilization depends on the ratio of the total propagation delay around the ring to the cycle length. This is an effect related to the asynchronous passing mechanism of the slot-initiator assignment. Also, the bandwidth utilization depends on the average number of segments that can be utilized in each slot. The maximum aggregated throughput of the network that is made possible by the asynchronous slot-synchronization method $S_{\max}$ is:

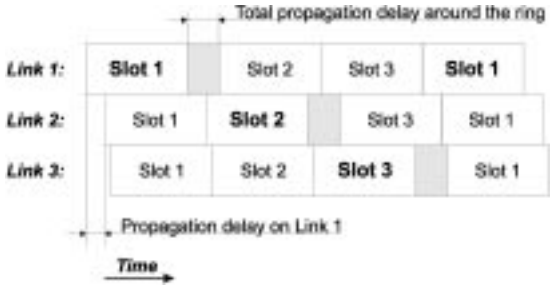


Fig. 8 Bandwidth utilization depends on the ratio of the total propagation delay around the ring to the cycle length. The boxes with bold text show the link through which each slot first propagates.

$$S_{\max}=\frac{MPT_{\text{slot}}}{MT_{\text{slot}}+T_{\text{prop}}}, \tag{1}$$

where M is the number of nodes, P is the average number of packet transmissions in each slot, T_{slot} is the duration of one slot, and T_{prop} is the total propagation delay around the ring. As an example we get a throughput of $S_{\max}=1.9$ when $M=16$, $P=2$, $T_{\text{slot}}=1\ \mu$s, and $T_{\text{prop}}=1\ \mu$s (200-m fiber).

The latency grows linearly with distance, measured in the number of hops (repeating latency in each node). Contributions to the latency also include the propagation delay between source and destination node, queuing delay, and the delay until the first available slot for transmission. By distributing tasks in such a way as to minimize the number of hops, latency is reduced and remaining bandwidth is improved.

Since a separate control channel is used, the data-packet header can be very short. Therefore, reception of data packets is simplified and large software overheads are eliminated. Another positive consequence, in combination with the asynchronous slot synchronization, is that the slot length can be relatively short without significant reduction in bandwidth utilization. In turn, short slot lengths decrease the latency and provide a finer resolution for splitting messages into packets, which can increase the bandwidth utilization. Due to limited space, a performance analysis of the network is to be published elsewhere.

2.3 *Slot Reserving*

Many computer systems have real-time demands for which the network must offer guaranteed bandwidth for certain communication patterns. This can be done in the network using slot reserving. Either the whole ring is reserved for a specific node in a slot, or several segments of the ring are dedicated to some specific nodes.

When slot reservation is allowed, the cycle is prolonged to contain $Q=M+R$ slots, where R is the number of slots used for reservation. The value of R is chosen when the system is designed and remains unchanged during operation of the network, provided the system function does not change radically. For example, there could be a mode change in a radar system, such as switching from the task of scanning the whole working range to that of tracking a certain object. For fairness, the M ordinary slots are not allowed to be reserved. Figure 9 shows how the cycle from Fig. 8 is prolonged by a fourth slot where node m_3 is the

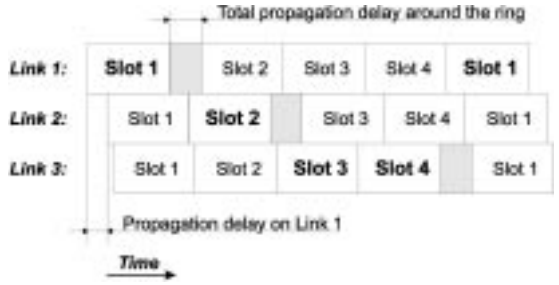
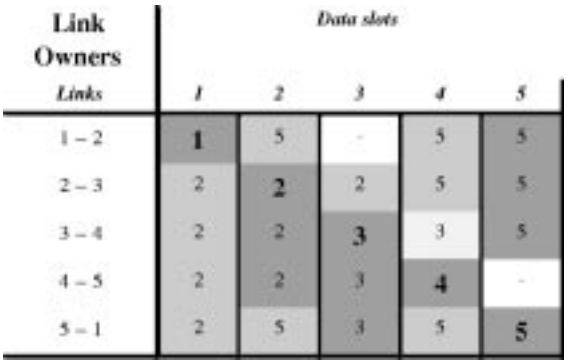


Fig. 9 Bandwidth utilization is improved using slot reserving. In this example, the cycle is prolonged by a fourth slot.



Link Owners / Links	Data slots 1	2	3	4	5
1 – 2	**1**	5	-	5	5
2 – 3	2	**2**	2	5	5
3 – 4	2	2	**3**	3	5
4 – 5	2	2	3	**4**	-
5 – 1	2	5	3	5	**5**

Fig. 10 Example of an allocation scheme for the links in a five-node system. The slot-initiators are in bold type and different segments have different background shading.

slot-initiator. However, any node in the network can try to reserve a segment of the ring in the fourth time slot.

When a node is going to reserve a slot, it searches for slots where the required links are free, so allocation of a new segment can be made. First, the node's own slots (i.e., where the node itself is the slot-initiator) are searched. When too few slots (actually only a segment in each slot) for the reservation can be allocated, the search is extended to other slots. In that case, the node broadcasts a data packet containing a request to all other nodes to allocate the desired segment in their slots. The packet contains information about the links required and the number of slots needed. Each node then checks its own slots for the required free links. All of the nodes send a packet back to the requesting node to notify which slots, if any, have been allocated. When the requesting node has received the answers, it decides if the number of allocated slots is sufficient. If not, it sends a release packet. Otherwise, it can start using the reserved slots immediately. However, if more slots than needed were allocated, a release packet is sent out.

All slots in a cycle, where a node is the slot-initiator, must be in sequence to avoid disturbing the efficiency of the asynchronous slot synchronization method. The bandwidth utilization is, at the expense of higher latencies, improved when using slot reserving, as indicated in Fig. 9. The maximum aggregated throughput of the network made possible by the asynchronous slot-synchronization method $S_{\max}$ now becomes:

$$S_{\max}=\frac{(M+R)PT_{\text{slot}}}{(M+R)T_{\text{slot}}+T_{\text{prop}}}, \tag{2}$$

where M is the number of nodes, R is the number of slots for reservation, P is the average number of packet transmissions in each slot, T_{slot} is the duration of one slot, and T_{prop} is the total propagation delay around the ring.

The advantage of this slot reservation method over circuit-switching is that when a node does not require its reserved slot, the slot can be used by other nodes in the segment. For example, suppose that node m_1 has a segment reserved containing the four links between itself and node m_5. If node m_1 does not require the slot in a cycle, the other nodes in the segment are informed of that when the control packet passes in the slot before. Node m_2 will have the first chance to take over, followed by nodes m_3 and m_4. Multiple nodes can even reuse the same slot when the communication demands of the other nodes in the segment allow for that.

3 Packet- and Circuit-Switched Ring Network

Compared to the CC-FPR network, the network for both packet- and circuit-switched traffic is slightly simpler at the expense of somewhat reduced support for dynamic traffic patterns. However, in many systems only a fraction of the traffic is irregular.

Circuit and packet-switched traffic are discussed in Sections 3.1 and 3.2, respectively. Then, in Section 3.3, circuit establishment is described.

3.1 *Circuit-Switched Traffic*

For circuit-switched traffic, the first nine fibers in each link form a high-speed channel. All of the high-speed channels, together, form a high-speed ring network for circuit switched traffic. The access is divided into slots as in an ordinary TDMA network. However, in each slot the network can be divided into segments as in the CC-FPR network. Also, for each slot there is always a slot-initiator node. The same kind of asynchronous slot-synchronization method is also used.

The access is cyclic and each cycle consists of K slots. In a typical case, K is a multiple of M, where M is the number of nodes, and each node is the slot-initiator in K/M slots. An example of an agreed schedule for a network with $K=M=5$ slots per cycle is shown in Fig. 10. Each column represents one time slot and contains information on how the ring is segmented in that slot. Each number in a column is the node index of the owner of the corresponding link. The bold-typed numbers indicate the current slot-initiator. In each segment and slot, one, and only one, node can be the owner of the links, and hence has the right to use the segment links for transmission. In the first slot in the example, node m_1 (slot-initiator) owns the link between itself and node m_2. Hence, it can transmit to node m_2 but not to any other node. In the same slot, node m_2 can transmit to any of nodes m_3, m_4, m_5, or m_1. The choice is made by the process that owns the circuit (logical connection) to which the slot segment is associated. A multicast to two or more of these nodes is also possible.

In the third slot, the link between nodes m_1 and m_2 is free. Although the link is free, node m_1 must not disturb the asynchronous slot synchronization technique. Therefore, it transmits an empty packet to node m_2. In the fifth slot, node m_5 has the capability of transmitting a broadcast packet (a packet to all other nodes in the ring).

The same reasoning about bandwidth utilization and latency for the CC-FPR network also holds for this network. The bandwidth utilization is:

$$S_{\max}=\frac{KPT_{\text{slot}}}{KT_{\text{slot}}+T_{\text{prop}}}. \tag{3}$$

3.2 *Packet-Switched Traffic*

The tenth fibers from each of the links are combined to form a ring network totally dedicated for packet-switched traffic. An ordinary ring protocol can be used. However, there are two requirements: (1) it must be possible to halt the protocol when special packets for circuit establishment are to be transmitted (see Section 3.3) and (2) the latency must be upper bounded to ensure transmission of the packets for circuit establishment. When using, e.g, a token ring protocol on the packet network, this network will support low latency communication for sporadic packets at moderate traffic rates. At the same time, it is ensured that the circuit switched traffic (often real-time traffic) is not disturbed by packet-switched traffic.

3.3 *Circuit Establishment*

When a node is to establish a new circuit, it searches for slots where the required links are free so allocation of a new segment can be made. First, the node's own slots (i.e., where the node itself is the slot-initiator) are searched. When too few slots (actually only a segment in each slot) for the circuit can be allocated, the search is continued in other slots. In that case, a special *request packet* is transmitted on the packet network to ask other nodes to allocate the desired segment in their slots. This packet is immediately followed by a *collect packet* to collect information on the success of the slot segment allocations.

The request packet, which is broadcast to all other nodes, contains information about the links required and the number of slots required. Each node then checks if any of its own slots have the required free links. If so, it prepares to modify the collect packet when it arrives (before forwarding it), to notify the requesting node of which slots have been allocated. However, if any of the previous nodes have already allocated slot segments and modified the collect packet, the number of slots required is decreased by the corresponding number of allocated slots. The number of slots still required is indicated by a dedicated field in the collect packet. In this way, allocation of more slots than required is avoided. However, several nodes can each allocate some of the slots required and information about all of these allocations is added to the same collect packet.

When the requesting node receives the collect packet after one round, it decides if the number of allocated slots is sufficient. If not, it sends a release packet. Otherwise, it can start using the established circuit immediately.

4 Implementation Aspects

In addition to the high bandwidth offered by a fiber-ribbon cable, a 10-fold increase in packing density compared to electrical cables, resulting in less rigid cables, is also offered.[12] Further on, the designer must not be concerned about electromagnetic emissions. These properties make possible new components such as the single-chip optoelectronic switch core reported in Ref. 13. In addition to the switch function the chip eliminates 32 OPTOBUS 800 Mbits/s per fiber transceivers. This translates to an aggregated bandwidth of 204 Gbits/s through the switch when 8 of the 10 fibers on each link are used for data. Such a switch can connect multiple ring clusters when building large networks. The high bit rate of a fiber-ribbon link makes it possible to reduce the slot duration in the proposed networks, and still maintain the same number of bits in a packet as on a slower link based on a electrical cable. This reduces the latency without offering too much in bandwidth utilization.

When scaling up the bandwidth of a fiber-ribbon link where a dedicated fiber carries the clock signal, the main problem is channel-to-channel skew. The skew is mainly due to differences in propagation delay between different fibers and variations of lasing delay time among different laser diodes.[14] The 400 Mbits/s OPTOBUS has a specified maximum skew of 200 ps excluding the fiber-ribbon cable for which 6 ps/m is assumed for standard ribbons.[5] Since the data stream passing a node in the ring network is, at least, passing a pipeline register, the channel-to-channel skew is not accumulated over several nodes. The limited distance between two adjacent nodes, due to skew, is of the same order of magnitude as current LANs (a few hundred meters). In parallel and distributed computer systems, so called system area networks (SANs), the maximum required distance is normally lower than this limit. It might, however, be hard to stress the bit rate to several gigabits per second per fiber without reducing the distance significantly. It can also be argued that networks with more physically distributed nodes should be possible since this could be valuable in some applications. The latency of the network is not dependant on the distance except that the propagation delay is added and the throughput is reduced, as mentioned. Because of these motivations, techniques to reduce the effect of the skew are discussed next.

One technique is to actually reduce the skew, either by using low-skew ribbons or by employing skew compensation. Fiber ribbons with about 1 ps/m skew have been developed,[15] which essentially increases the possible bandwidth-distance product. All the fibers in the same ribbon were sequentially cut to reduce the variation of refractive index among the fibers. In the fiber-ribbon link described in Ref. 16, a dedicated fiber carries a clock signal used to clock data on 31 fibers. The transmitter circuitry for each channel has a programmable clock skew adjustment to adjust the clock in 80 ps increments.

Another technique is to extract the clock signal from the bit flow on each fiber instead of using a separate fiber carrying the clock signal. The disadvantage is increased hardware complexity when adding a clock recovery circuit and a buffer circuit for each channel in the receiver. A hybrid solution is to skip the separate clock channel and encode clock information on the data channels, but still send in

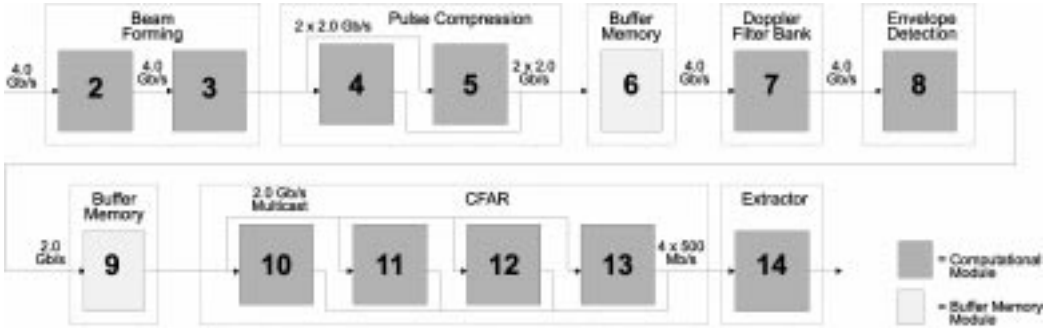


Fig. 11 Data flow between the modules in the radar signal processing chain.

bit-parallel mode as reported in Ref. 17. In this case, a deskew unit relying on first in, first out (FIFO) registers ensures that parallel data words that are output from the receiver are identical with those that were sent. A possible ± 15-ns deskew was reported.

The techniques mentioned here introduce either increased hardware complexity or a more sophisticated manufacturing process of the fiber ribbons. If the manufacturing process allows for adding more fibers in each ribbon, this might be a cheaper alternative. For example, 12 channel links with 1 Gbit/s per channel[12] and 2 Gbit/s per channel[18] have been reported, and array modules supporting 12×2.4 Gbits/s for, e.g., fiber-ribbon links were described in Ref. 19. A fiber-ribbon link with 32 fibers, each with a bit rate of 500 Mbits/s, was described in Ref. 16, and researchers at NEC developed a module where 8×2 lasers are coupled to two fiber ribbons.[20] Instead of fiber ribbons, fiber imaging guides (FIGs) with thousands of pixels can be used. In the system described in Ref. 21, both a 14,000-pixel FIG and a 3500-pixel FIG was coupled to an 8×8 vertical-cavity surface-emitting laser array in different setups. More references to reports on fiber-ribbon links are found in Ref. 22.

5 Case Study

A typical application with high throughput requirements and a pipelined data flow between the computational modules is future radar signal processing systems.[1,2] In Fig. 11, a signal processing chain, similar to those described in Refs. 1 and 2, is shown together with its bandwidth demands. Each computational module in the figure contains multiple processors. The chain is a good example containing both multicast, one-to-many, and many-to-one communication patterns. The aggregated throughput demand is 30 Gbits/s. Only the throughput requirements are treated here; all details of the chain are covered in Refs. 1 and 2. The data flow must not be disturbed by, for example, status information that the network must also transport. Slot reserving is therefore a good choice for the data flow of the signal processing chain when using the CC-FPR network. If the other network proposed here is used, circuit-switching is used for the data flow. We begin the case study by concentrating on the CC-FPR network and then discuss the circuit- and packet-switched network.

We assume links with 10 fibers and 800 Mbits/s per fiber in the case study. In the CC-FPR network, this translates to a bandwidth of 6.4 Gbits/s for data traffic on eight of the fibers. For simplicity we assume an efficient bandwidth of 6.0 Gbits/s after, for example, check sums have been excluded and Equation (2) has been used. In Fig. 11, there are 13 nodes. In addition, the antenna is seen as one node (feeds the first node in the chain with data) and there is one master node responsible for supervising the whole system and interacting with the user. We denote the antenna as node m_1, the modules shown in the figure as node m_i, $2\leqslant i\leqslant 14$, and the master node as m_{15}. The numbers of the modules are indicated in the figure also. The number of ordinary slots, hence, is 15, but the cycle is prolonged to contain also 30 slots for reservation. Accordingly, there are 45 slots in a cycle, where one slot per cycle corresponds to a bandwidth of 133 Mbits/s at an total efficient bandwidth of 6.0 Gbits/s.

A feasible allocation scheme of the slots is shown in Fig. 12. For clarity, all of the reservation slots are placed after the ordinary slots. In a real implementation, however, the reservation slots are spread out so each node is first a slot-initiator in the ordinary slot and then, immediately afterward, in two reservation slots. Care must be taken, however, when distributing the reservation slots because when there are intermediate nodes between the source and destination nodes, allocation is not possible in those slots where one of the intermediate nodes is the slot-initiator.

The maximum data flow from one module is 4.0 Gbits/s, which corresponds to having a segment of the ring reserved in all of the 30 slots for reservation. Slots for both of the two 2 Gbits/s data flows to the pulse compression nodes can be allocated, since one of the two data flows is tapped before adding the data flow produced from the same node. The incoming data flow to the constant false alarm ratio (CFAR) nodes is multicasted to all of these nodes. Although this multicast data flow must remain unchanged until the last CFAR node, it can coexist with the data flow produced from the CFAR nodes. The reason for this is that the multicast bandwidth is only 2 Gbits/s. The rest of the data flows are pure pipeline flows and fit easily on the network as long as the calculations are mapped on the nodes according to the pipeline order.

In the case of the second network, each cycle can be divided into $K=45$ slots per cycle for circuit-switched traffic. The allocation scheme in Fig. 12 then holds for this network too, leaving slots 1 through 15 free. Another possibility is to have $K=12$ slots per cycle. In that case, one slot corresponds to 500 Mbits/s and all bandwidths in Fig. 11 are divisible by the slot bandwidth.

	Data slots																							
Links	*1*	*2*		*15*	*16*	*17*	*18*	*19*	*20*	*21*	*22*	*23*	*24*	*25*	*26*	*27*	*28*	*29*	*30*	*31*	*32*	*33*		*45*
1 – 2	1				1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		1
2 – 3		2			2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2		2
3 – 4					3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3		3
4 – 5					4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	3	3	3		3
5 – 6					4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	5	5	5		5
6 – 7					6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6		6
7 – 8					7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7		7
8 – 9			.....		8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	–	–	–	.....	–
9 – 10					–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	9	9	9		9
10 – 11					10	10	10	10	–	–	–	–	–	–	–	–	–	–	–	9	9	9		9
11 – 12					10	10	10	10	11	11	11	11	–	–	–	–	–	–	–	9	9	9		9
12 – 13					10	10	10	10	11	11	11	11	12	12	12	12	–	–	–	9	9	9		9
13 – 14					10	10	10	10	11	11	11	11	12	12	12	12	13	13	13	13	–	–		–
14 – 15					–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–		–
15 – 1				15	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–		–

Fig. 12 One cycle of 45 slots where slots 16 through 45 are reserved. For slots 16 through 45, each number in the table indicates the owner of the corresponding link. Neighboring segments, in each slot, have different background shading. The slot-initiators are indicated in slots 1 through 15.

6 Conclusions

Two ring networks based on fiber-ribbon links were presented. Very high throughputs can be achieved in the networks, especially in systems for which a pipelined data flow between the nodes exists. The CC-FPR network relies on the use of a dedicated control subnetwork that is easy to implement using a dedicated fiber on each fiber-ribbon link. Guaranteed bandwidth is supported in the CC-FPR network by slot reserving, a method that allows for slot reuse when the guaranteed bandwidth is temporarily not in use. In a typical system, slot reserving can be used for time-critical data flows, guaranteeing that they are not disturbed by, e.g., control information. In the second network, circuit-switching is offered to support guaranteed bandwidth, while a packet-switched subnetwork at the same time supports flexible traffic. Note also that the network can be built today using fiber-optic off-the-shelf components and this work is in progress.

Acknowledgments

This work is part of the REMAP project, financed by NUTEK, the Swedish National Board for Industrial and Technical Development, and of the PARAD project, financed by the KK Foundation in cooperation with Ericsson Microwave Systems AB.

References

1. M. Jonsson, A. Åhlander, M. Taveniku, and B. Svensson, ''Time-deterministic WDM star network for massively parallel computing in radar systems,'' in *Proc. Massively Parallel Processing using Optical Interconnections, MPPOI'96*, pp. 85–93, IEEE Computer Society, Lahaina, HI (1996).
2. M. Taveniku, A. Åhlander, M. Jonsson, and B. Svensson, ''A multiple SIMD mesh architecture for multi-channel radar processing,'' in *Proc. Int. Conf. on Signal Processing Applications and Technology, ICSPAT'96*, pp. 1421–1427, Miller Freeman, Inc., Boston, MA (1996).
3. J. A. Stankovic, ''Misconceptions about real-time computing,'' *Computer* **21**(10), 10–19 (1988).
4. D. Bursky, ''Parallel optical links move data at 3 Gbits/s,'' *Electron. Design* **42**(24), 79–82 (1994).
5. D. B. Schwartz, K. Y. Chun, N. Choi, D. Diaz, S. Planer, G. Raskin, and S. G. Shook, ''OPTOBUS™ I: performance of a 4 Gb/s optical interconnect,'' in *Proc. Massively Parallel Processing using Optical Interconnections (MPPOI'96)*, pp. 256–263, IEEE Computer Society, Maui, HI (1996).
6. M. I. Irshid and M. Kavehrad, ''A fully transparent fiber-optic ring architecture for WDM networks,'' *J. Lightwave Technol.* **10**(1), 101–108 (1992).
7. A. Louri and R. Gupta, ''Hierarchical optical ring interconnection (HORN): scalable interconnection network for multiprocessors and multicomputers,'' *Appl. Opt.* **36**(2), 430–442 (1997).
8. M. Jonsson, K. Nilsson, and B. Svensson, ''A fiber-optic interconnection concept for scaleable massively parallel computing,'' in *Proc. Massively Parallel Processing using Optical Interconnections, MPPOI'95*, pp. 313–320, IEEE Computer Society, San Antonio, TX (1995).
9. P. C. Wong and T.-S. P. Yum, ''Design and analysis of a pipeline ring protocol,'' *IEEE Trans. Commun.* **42**(2/3/4), 1153–1161 (1994).
10. H. Jafari, T. G. Lewis, and J. D. Spragins, ''Simulation of a class of ring-structured networks,'' *IEEE Trans. Comput.* **29**(5), 385–392 (1980).
11. M. Xu and J. H. Herzog, ''Concurrent token ring protocol,'' in *Proc. INFOCOM'88*, pp. 145–154, IEEE Conf. on Computer Communication (1988).
12. H. Karstensen, C. Hanke, M. Honsberg, J.-R. Kropp, J. Wieland, M. Blaser, P. Weger, and J. Popp, ''Parallel optical interconnection for uncoded data transmission with 1 Gb/s-per-channel capacity, high dynamic range, and low power consumption,'' *J. Lightwave Technol.* **13**(6), 1017–1030 (1995).
13. T. H. Szymanski, A. Au, M. Lafrenière-Roula, V. Tyan, B. Supmonchai, J. Wong, B. Zerrouk, and S. T. Obenaus, ''Terabit optical local area networks for multiprocessing systems,'' *Appl. Opt.* **37**(2), 264–275 (1998).
14. T. Kurokawa, S. Matso, T. Nakahara, K. Tateno, Y. Ohiso, A. Wakatsuki, and H. Tsuda, ''Design approaches for VCSEL's and VCSEL-based smart pixels toward parallel optoelectronic processing systems,'' *Appl. Opt.* **37**(2), 194–204 (1998).
15. S. Siala, A. P. Kanjamala, R. N. Nottenburg, and A. F. J. Levi, ''Low skew multimode ribbon fibres for parallel optical communication,'' *Electron. Lett.* **30**(21), 1784–1786 (1994).
16. Y.-M. Wong, D. J. Muehlner, C. C. Faudskar, D. B. Buchholz, M. Fishteyn, J. L. Brandner, W. J. Parzygnat, R. A. Morgan, T. Mullally, R. E. Leibenguth, G. D. Guth, M. W. Focht, K. G. Glogovsky, J. L. Zilko, J. V. Gates, P. J. Anthony, B. H. Tyrone, Jr., T. J. Ireland, D. H. Lewis, Jr., D. F. Smith, S. F. Nati, D. K. Lewis, D. L. Rogers, H. A. Aispain, S. M. Gowda, S. G. Walker, Y. H. Kwark, R. J. S. Bates, D. M. Kuchta, and J. D. Crow, ''Technology development of a high-density 32-channel 16-Gb/s optical data link for optical interconnec-

tion applications for the optoelectronic technology consortium (OETC),'' *J. Lightwave Technol.* **13**(6), 995–1016 (1995).
17. T. Yoshikawa, S. Araki, K. Miyoshi, Y. Suemura, N. Henmi, T. Nagahori, H. Matsuoka, and T. Yokota, ''Skewless optical data-link subsystem for massively parallel processors using 8 Gb/s×1.1 Gb/s MMF array optical module,'' *IEEE Photon. Technol. Lett.* **9**(12), 1625–1627 (1997).
18. H. Karstensen, ''Parallel optical links—PAROLI, a low cost 12-channel optical interconnection,'' in *Proc. LEOS'95*, Vol. 1, pp. 226–227, IEEE Lasers and Electro-Optics Society (1995).
19. R. G. Peall, ''Development in multi-channel optical interconnects under ESPRIT III SPIBOC,'' in *Proc. LEOS'95*, Vol. 1, pp. 222–223, IEEE Lasers and Electro-Optics Society (1995).
20. K. Kasahara, ''Optical interconnects speed up networks,'' *Photon. Spectra* **32**(2), 127–128 (1998).
21. Y. Li, T. Wang, and S. Kawai, ''Distributed crossbar interconnects with vertical-cavity surface-emitting laser-angle multiplexing and fiber image guides,'' *Appl. Opt.* **37**(2), 254–263 (1998).
22. F. A. P. Tooley, ''Optically interconnected electronics—challenges and choices,'' in *Proc. Massively Parallel Processing using Optical Interconnections (MPPOI'96)*, pp. 138–145, IEEE Computer Society, Maui, HI (1996).

Magnus Jonsson received his BS and MS degrees in computer engineering from Halmstad University, Sweden, in 1993 and 1994, respectively. He then obtained a Licentiate of Technology degree in computer engineering from Chalmers University of Technology, Gothenburg, Sweden, in 1997. He has been working at Halmstad University since 1993, where he now is an assistant professor. He has authored and co-authored more than 10 scientific papers, most of them in the area of fiber-optic communication protocols for parallel and distributed processing systems.