Bergenhem, C., and M. Jonsson, "A pipelined fiber-ribbon ring network with heterogeneous real-time support", in ARTES – A network for Real-Time research and graduate Education in Sweden 1997-2006, Editor: Hans Hansson, pp. 373-395, Mar. 2006, ISBN 91-506-1859-8.

Chapter 16

A pipelined fiber-ribbon ring network with heterogeneous real-time support

By **Carl Bergenhem**[‡] and **Magnus Jonsson**[†]

‡Bergenhem Konsult
Email: carlb@bergenhem.com

†Laboratory for Computing and Communication
Halmstad University
Email: magnus.jonsson@ide.hh.se

This paper presents a fiber-optic ring network with support for heterogeneous real-time communication. The CCR-EDF (Control Channel Ring network with Earliest Deadline First scheduling) network is an optical fibre-ribbon pipelined ring network with a separate channel for network arbitration. The medium access protocol, that uses the control channel, provides low-level support for hard real time traffic and group communication functions such as barrier synchronisation and global reduction. The topology of the network is a pipelined unidirectional fibre-ribbon ring that supports several simultaneous transmissions in non-overlapping segments. Access to the network is divided into time-slots. In each slot the node that has the highest priority message is permitted to transmit. The novel medium access protocol uses the deadline information of individual packets, queued for sending in each node, to make decisions, in a master node, about who gets to send. This feature of the medium access protocol gives the network the functionality for earliest deadline first scheduling. Different classes of traffic are supported for the user. These are guaranteed

logical real-time channels (LRTC), best effort (BE) channels and non real-time (NRT) traffic.

16.1 Introduction

The CCR-EDF (Control Channel Ring network with Earliest Deadline First scheduling) network is an optical fibre-ribbon pipelined ring network with a separate channel for network arbitration. The medium access protocol, that uses the control channel, provides low level support for hard real time traffic and group communication functions such as barrier synchronisation and global reduction. [1]. The basic network structure is depicted in Figure 16.1. The novel medium access protocol uses the deadline information of individual packets, queued for sending in each node, to make decisions, in a master node, about who gets to send. The new protocol may be used with a previously presented network topology; the control channel based fibre ribbon pipeline ring (CC-FPR) network [2].

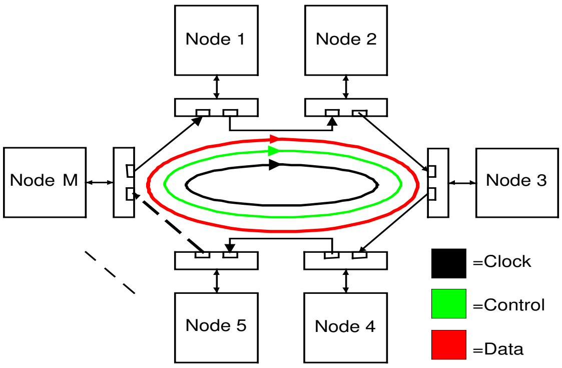

Figure 16.1: A Control Channel based Fiber Ribbon Pipeline Ring network

In addition to local area networks, the CCR-EDF network is also suitable for parallel and distributed real-time systems. Application examples are future radar signal processing systems, distributed multimedia systems, satellite imaging and other image processing appplications. A typical example is the radar signal processing system described in [3], [4]. 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 [5]. In distributed real-time systems, the interconnection network is a very important part of the computer system. Often, guaranteeing real-time services are much more important in these systems than performance, e.g., average latency.

Radar signal processing (RSP) in an airborne environment is an area of communication where the network may be applied. The computation in RSP is done in a distributed fashion. Therefore the communications network is a vital part of a RSP system. Studies have shown that the radar processing algorithms map suitably on a ring topology such as the one discussed in this report [6].

The paper also presents results of simulations where the network has been tested together with defined cases. These cases are first defined and presented. The first case is a radar signal processing (RSP) application. The RSP system has a number of different requirements depending on the application, but the algorithms comprise mainly linear operations such as matrix-by-vector multiplication, matrix inversion, FIR-filtering, DFT etc. In general, the functions will work on relatively short vectors and small matrices, but at a fast pace and with large sets of vectors and matrices. Even if the incoming data can be viewed as large 3-dimensional data cubes, it is quite easy to divide the data into small tasks that each can be executed on a separate processing element in a parallel computer. Actually, the computational demands are so high that it implies that parallel computer systems are needed. In turn, the performance of parallel computers is highly dependent on the performance of their interconnection networks, especially in data intensive applications like radar signal processing. In addition to the transfer of radar data, there are control functions and registration functions controlling and monitoring the system. Both data and control traffic have real-time demands. Results of the simulations indicate that the CCR-EDF network works well with the RSP application studied. To show the networks broad applicability, a case is also defined,not simulated, for the network used within large IP-routers. When building large IP routers multiple racks are used (hundreds or thousands of input /output –ports in total). The intelligence (to take routing decisions etc) can be implemented either in racks with line cards (interfacing to ingoing and outgoing physical channels) and/or in centralized routing racks. These racks must be interconnected to realise the router. For flexibility in design, operation, and maintenance of the system, a network such as CCR-EDF (or network with similar properties i.e. real-time support etc.) is desirable. The complete system takes the form of a distributed router.

The article is organised as follows. The network itself and the protocol are described in Sections 2 and 3, respectively. Two cases, the radar signal processing case and the large IP-router case are presented in Sections 4 and 5 respectively. The radar signal processing case, which is simulated in the later sections, is further defined together with the simulator setup, in Section 6. Three different simulations are presented and discussed in Section 7. A discussion about how throughput is defined is presented in Section 8. Finally, conclusions are drawn in Section 9.

16.2 The CCR-EDF network architecture

Motorola OPTOBUS$^{\text{TM}}$ bi-directional links (or similar) with ten fibres per direction are (in this report) assumed to be used but the links are arranged in a unidirectional ring architecture where only $\lceil N/2 \rceil$ bi-directional links are needed to close a ring of N nodes. All links are assumed to be of the same length. The increasingly good price/performance ratio for fibre-ribbon links indicates a great success potential for the proposed type of networks.

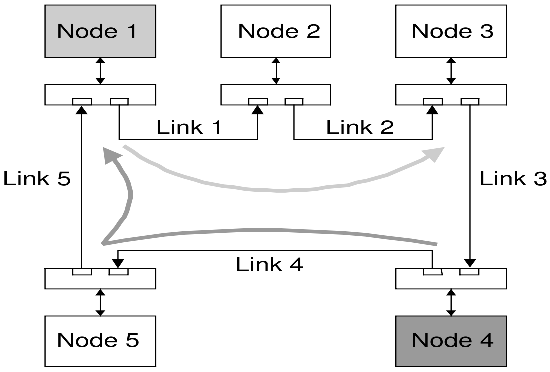

Figure 16.2: Example where Node 1 sends a single destination packet to Node 3, and Node 4 sends a multicast packet to Node 5 and Node 1.

The ring can dynamically (for each slot) be partitioned into segments to obtain a pipeline optical ring network [7]. Several transmissions can be performed simultaneously through spatial bandwidth reuse, thus achieving an aggregated throughput higher than the single-link bit rate (see Figure 16.2 for an example). Even simultaneous multicast transmissions are possible as long as multicast segments do not overlap. Although simultaneous transmissions are possible in the network because of spatial reuse, each node can only transmit one packet at a time and receive one packet at a time.

For the following discussion, the term master node is exchangeable with “the node with clocking responsibility etc.”. That is, the master node also clocks the network. The clocking strategy functions as follows. During a slot, the node that has the highest priority message, according to the arbitration process, has the responsibility to clock the network. In the following slot, the clocking responsibility is handed over to the node that has the highest priority message in that slot. This may be another node or the same one as in the previous slot. Thus clock hand-over is always done in accordance with the result of the medium access arbitration process, described further in the next section. The result of the arbitration process is knowledge of all messages at the heads of the local queues in all nodes, and therefore also knowledge about which node has the highest priority message in the entire system. The current master distributes this information to all nodes. A distribution packet is sent so that the end of the packet corresponds with the end of the slot. This implies that, when the master stops the clock at the end of the slot, all nodes have the information that they need to perform clock hand-over that takes place in the gap between slots. The node that has highest priority in the coming slot detects when the clock signal is stopped and assumes the master role. The highest priority node knows that it will be master because of the information received in the distribution phase packet.

Since the node that is master, also the node that has the highest priority message, has responsibility for generating the clock, then there cannot occur a situation where the node cannot send its message. This is because the node will at most send $N-1$ hops (where N is the number of nodes) and will never have to transmit past a master, i.e. cross the clock break at the master node. In the CCR-EDF network, access to the network is divided into time-slots. There is no concept of cycle since a cycle cannot be defined. In other cases a cycle would be e.g. when all nodes have been master once. However, in this network the master role is not shared equally among nodes but is given to the node with the highest priority message.

16.3 The CCR-EDF medium access protocol

The medium access protocol has two main tasks. The first is to decide and signal which packet(s) is to be sent during a slot. The second task is that the network must know exactly which node has the highest priority message in each slot. This is to be able to perform clock hand-over to the correct node. Therefore, this information is included as an index in the distribution phase packet.

The two phases of medium access are collection phase and distribution phase (see Figure 16.3). As can be seen, medium access arbitration occurs in the time slot prior to the actual transmission. The protocol is time division multiplexed into slots to share access between nodes. A disadvantage with the CC-FPR protocol presented in [2] is that a node only considers the time constraints of packets that are queued in it, and not in downstream nodes. As an example (see Figure 16.2), Node 1 decides that it will send and books

Links 1 and 2, regardless of what Node 2 may have to send. This means that packets with very tight deadlines may miss their deadlines. The novel network presented here does not suffer from this problem. In the collection phase, the current master initiates by generating an empty

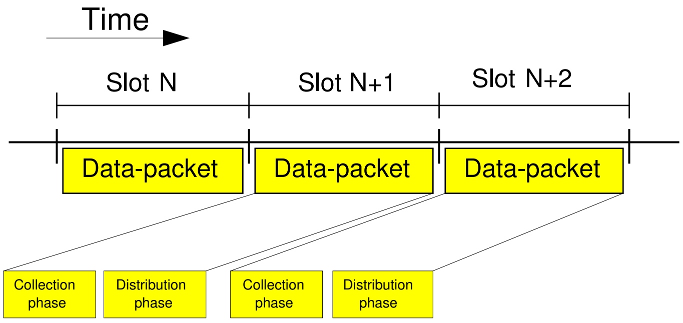

Figure 16.3: The two phases, collection and distribution, of the TCMA protocol. Notice that the network arbitration information, for data in slot $N+1$,is sent in the previous slot, slot N.

packet, with a start bit only, and transmits it on the control channel. Each node appends its own request to this packet as it passes. The master receives the complete request packet (see Figure 16.4) and sorts the requests according to priority (urgency).

Type	Start	Prio	Link reservation field	Destination field	Prio	Link reservation field	Destination field	
		node i	node i	node i	node i+1	node i+1	node i+1	etc.
Nbr of bits	1	5	N	N	5	N	N	

Figure 16.4: Contents of the collection phase packet. The figure shows that one request per node make up the complete packet. Each request consists of three fields, the primary field, the link reservation field and the destination field.

The network can handle three classes of traffic: logical real-time connection, best effort, and non-real time. Which class of traffic that a certain message belongs to, is signalled to the master with the priority field in the request (see Figure 16.4). Table 16.1 shows the allocation of the priority field to each user service in the network. The time until deadline (referred to as laxity) of a message is mapped, with a certain function, to be expressed within the limitation of the priority field size, see Table 16.1. This applies to both logical real-time connection and best effort traffic. A shorter laxity of the packet implies a higher priority of the request. The result of the mapping is written to the priority field. One priority level is reserved (0 in the proposed implementation of the protocol) and used by a node to indicate that it does not have a request. If so, the node signals this to

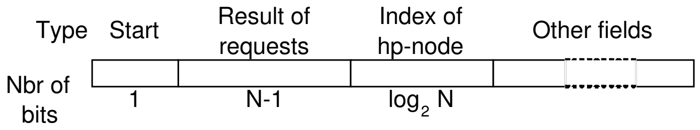

Figure 16.5: The modified TCMA distribution phase packet. The field labelled “index of hp-node” contains the index of the node that has the highest priority message.

0	Nothing to send
1	Non-Real Time
2–16	Best Effort
17–31	Logical real-time connection

Table 16.1: The allocation of priority levels to user services. A higher priority within the traffic class implies shorter laxity and a more urgent message.

the master by using the reserved priority level and also writes zeros in the other fields of the request packet. Observe that messages that are part of LRTCs (logical real-time connections) always have higher priority than any other service. However, a possible situation, considering spatial reuse, is that a best effort message uses the spatially reused capacity and may be transmitted simultaneously as a logical real-time connection message. The best effort message does not affect the logical real-time connection message. Observed locally in a node, best effort messages will only be requested to be sent if there is no logical real-time connection message queued. The same applies to non real-time message. They are only sent if there are no best effort and no logical real-time connection messages.

Request priority is a central mechanism of the CCR-EDF protocol. Deadlines are mapped with a function to priority. For the following discussion, a logarithmic mapping function is assumed. This mapping gives higher resolution of laxity, the closer to its deadline a packet gets. Further discussion of deadline to priority mapping function is out of the scope of this paper.

When the completed collection phase packet arrives back at the master, the requests are processed. There can only be N requests in the master, as each node gets to send one request per slot. The list of requests is sorted in the same way as the local queues. The master traverses the list, starting with the request with highest priority (closest to deadline) and then tries to fulfil as many of the N requests as possible, avoiding overlapping transmission segments.

The second phase, the distribution phase, is described as follows. The master sends a packet, see Figure 16.5, on the control channel that contains the result of the arbitration. This is either acceptance or denial of a node's request and also which node contains the highest priority message in that slot. All nodes read the message. A request was granted if the corresponding bit in the "request result field" of the distribution phase packet contains a "1". The protocol also has a feature to permit several non-overlapping transmissions, that is grant several requests, during one slot. This function is called spatial reuse and is used during run-time. Observe that the distribution phase packet also may contain other information such as acknowledgement for transmission, group communication etc. These are further described in [8] and will not be part of the discussion here.

The addition of an index that points to the node that has highest priority in the current coming slot, see Figure 16.4, enables all nodes to know who will have the highest priority message in the coming slot and that the node will assume the role as master and clock the network. The index field needs to be $\lceil log_2 N \rceil$ bits wide to represent numbers up to N. When all nodes have received the distribution phase packet, and hand-over has taken place, the new slot commences and data may begin to flow in the data channel.

16.4 A radar signal processing case

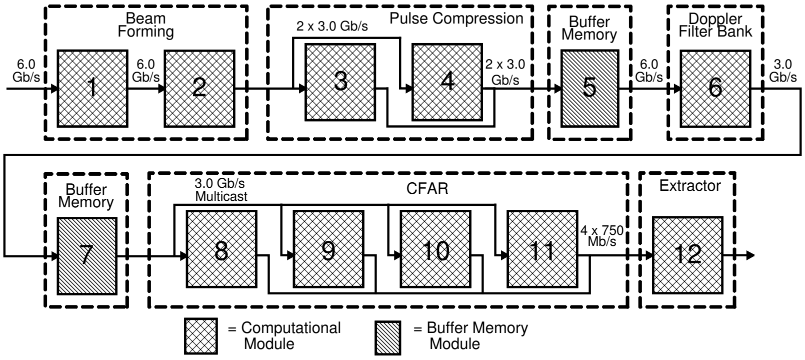

Figure 16.6: An irregular pipeline chain. Data flow between the modules in the radar signal processing chain.

In Table 16.2, there are three different communication patterns for data traffic. The difference between the cases is how the processing takes place and thus, how the bulk of the data is communicated. The

	Communication pattern	Delay Bound	Traffic amount	Arrival model
Control traffic	Out: broadcast In: many-to-one (Master / Slave)	100 ms per direction (guaranteed or highest priority)	1 kByte	Periodic @ 1 ms, for both directions
Data traffic	a)Irregular pipeline	1 ms for each packet in the data cube. *3)	96 Mbit data cubes @ 62.5 Hz *7)	A new data cube arrives every 16 ms *1)
	b)Straight pipeline	0.5 ms *4)	96 Mbit data cubes @ 62.5 Hz *7)	*1)
	c)SPMD	A corner turn must take place within 4ms(soft deadline)	96 Mbit data cubes *5) 60 kbit messages in CT *6)	*1) *2) Two corner turns will occur every 16 ms.
Other traffic	Assume broadcast for worst case	Non real-time. No bound but a certain capacity is needed.	100 Mbit/s is assumed to be representative	Periodic with an average period of 50 ms is assumed to be representative

Table 16.2: Assumptions for the traffic in the RSP case study.

communication pattern of the control traffic is independent of the data traffic; all nodes, despite processing model, are assumed to have central control. Generally, in the pipeline processing models there will be more that one data cube in processing at a time and only one at a time in the SPMD model. The three communication patterns are briefly discussed below.

Figure 16.6 depicts an example of an irregular pipeline. The difference between the two mentioned pipelines in [3] and [10] is the number of nodes in the processing chain and amount of the incoming data. Certain steps of the RSP algorithm are much more computationally intensive than others (the CFAR step in Figure 16.6 is an example). It is assumed for this processing mode that all nodes are equally powerful. Therefore some steps will be performed on more than one node and communication services to support this are required for efficiency. The irregular pipeline assumes that there will be multicast communication between certain nodes in the processing chain. Multicast communication is shown in Figure 16.6 where Node 7 multicasts to Nodes 9 through 11. Also, a many-to-one service is required when Node 12 collects the information from the previous four stages. One-to-many communication is also included in the chain.

For the straight pipeline, we assume that the bulk of the data communication is to the next neighbour (see Figure 16.7). Here it is assumed that processing is done in a pipelined fashion, such that the processing steps are divided among all modules equally. This is purified version only for evaluation purposes. For the straight pipeline we assume a network with 20 identical modules and a required latency for the whole pipeline of 100 ms. This gives 0.5 ms allocated for communication per data cube per module and a total of 90 ms processing time, per data cube.

In the SPMD (same program multiple data) processing model, all nodes take part in the same processing step at a time, where each node works on a part of the data cube. When the processing step is complete, the data is redistributed, if needed, to the appropriate nodes for the next step of processing. An SPMD processing model is depicted in Figure 16.8. It is assumed that each node has I/O capabilities for communication with external devices, e.g., for data from the antenna and for sending results that are ready.

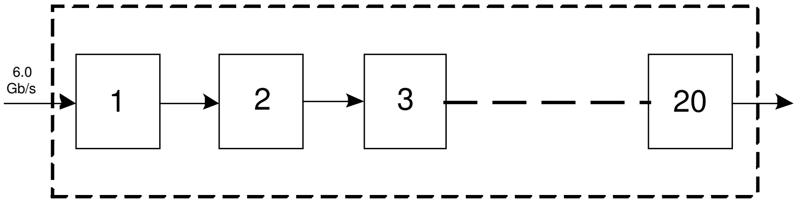

Figure 16.7: The straight pipeline

The main studied application is radar signal processing (RSP). In the main and supporting algorithms we can identify several different types of traffic. Examples of these and what they may require are:

- The radar signal data itself has soft real-time requirements, but expected performance must be determinable.
- The control traffic has, for obvious reasons, hard real-time requirements. Also real-time debugging requires a guaranteed service.
- Other general traffic such as logging and long term statistics do not have any real time constraints at all. System start-up, bootstrapping, firmware upgrading are tasks that don't require real time support. However, some kind of performance analysis is needed in order to design for desired performance.

These traffic types are further explained in Table 16.3, while some definitions and explanations are made later in this chapter. Some notes referred to in the table are:

*1) All three data traffic models will have periodic arrival of data cubes with a period of 16 ms.

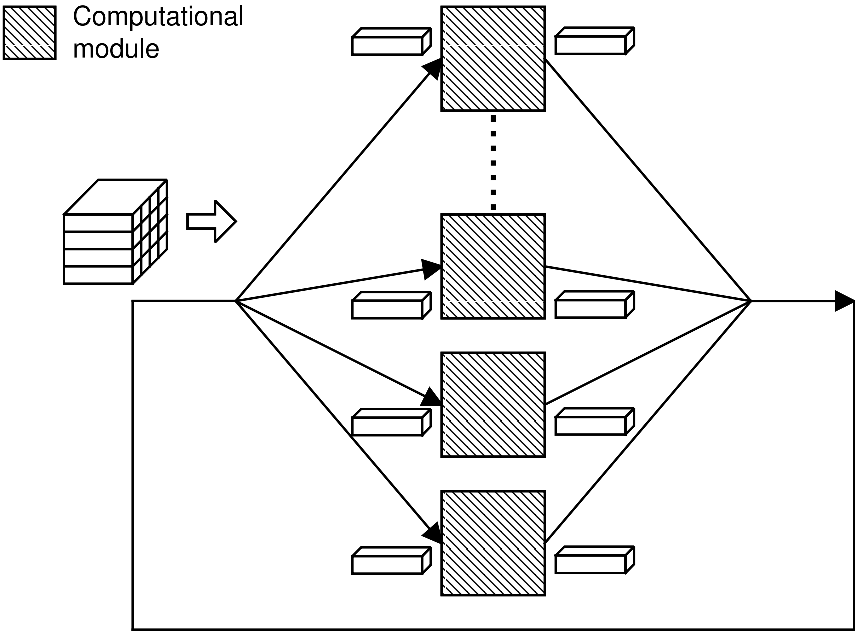

Figure 16.8: If the SPMD model is used, all modules work together on one stage in the signal processing chain at a time.

*2) New data cubes and results are received and sent on dedicated I/O.

*3) The communication delay bound (deadline) for the whole pipeline is 10 ms (10 % of 100 ms total latency including processing), with a new whole data cube arriving every 16 ms. The (soft) deadline for each packet in the data cube is 1 ms (≈ 10 ms / 12 steps). Priority based or soft deadline based service class is assumed (guaranteed service might be implemented on a higher level).

*4) The communication delay bound for the whole pipeline is 10 ms, with a new whole data cube arriving every 16 ms. The (soft) deadline for each packet in the data cube is 0.5 ms (10 ms / 20 steps). Priority based or soft deadline based service class (guaranteed service might be implemented on a higher level).

*5) 96 Mbit data cubes arriving with a rate of 62.5 Hz, i.e., a new data cube each 16 ms.

*6) 96 Mbit / (40 · 40) = 60 kbit messages in a corner turn (CT) with the all-to-all communication pattern (assuming that all 40 nodes are both source and destinations in the corner turn).

*7) It may be possible to start to send parts of the data cube before all processing is completed.

	Communication pattern	Delay bound	Traffic amount	Arrival model
Control traffic	Evenly distributed	10 ms max. latency, (Guaranteed or at least highest priority)	5-10 % of the amount of data traffic	Internet communication *1) *3)
Data traffic	Evenly distributed	*2), 100 ms max and 20 ms average latency (priority based, i.e., no deadline sorting of cells)	In the order of Tbit/s	Bursty. *1) A number of consecutive data cells will be send after one control cell.
Other low priority (e.g. routing table updates)	Evenly distributed	None, but certain capacity may be needed. Upper level protocols may handle this with time-outs.	At most 1 % of total traffic	More or less periodic

Table 16.3: The various traffic types in the large IP–router network.

Table 16.1 specifies parameters of the traffic in the RSP network. Parts of the assumptions presented in Table 16.1 are also found in [3] [10]. For both pipelined RSP chains (straight and irregular), the maximum latency of a result is 100 ms, including communications and processing. This latency is assumed to be composed of ca. 10 % communications delay. These two assumptions can be found in [3]. Both directions of the control traffic are periodic with a period of 1 ms. It is assumed that processing is co-ordinated in a master / slave fashion and that there is a node in the network that acts as master over the other nodes that do the processing. These latter nodes are referred to as slaves.

It is assumed that the RSP algorithm is the same for all the three different communication patterns for data traffic. The RSP algorithm has two corner turns per batch. It is also assumed that the incoming data rate from the antenna is 6.0 Gb/s. The amount of traffic depends on the representation of numbers, precision, etc. [3] [10].

The arrival model for data traffic case a) and b) is related to how often a new data cube is accepted and to the pipelined fashion. However, data is sent with finer granularity (minimum ca 1 500 bits per message), than a whole data cube between the nodes. Despite this, we can assume only target at a throughput that can deliver the whole data cube in time, and at a maximum delay for each packet. A new data cube arrives equally often in the SPMD case, but utilises separate I/O.

16.5 A Large IP router case

A rule of thumb concerning delay in routing is that the maximum delay from input port to output port should be 1 ms. In a router-architecture such as in Figure 16.4, the latency over the SAN is assumed to be 100 μs maximum. We assume that each router module consist of both input and output ports. A summary of the different traffic types in the interconnection network is presented in Table 16.3, where some notes referred to in the table are:

*1) We assume that each IP-datagram is split into fixed size cells at the source port and sent across the interconnection network to the destination. Therefore, there will be one control cell for each IP-datagram followed by a number of data cells that comprise the split IP-datagram. The arrival of control traffic is linked to the arrival of the IP-datagrams. Possibly, information about several IP-datagrams may be transferred in the same control cell.

*2) Data cells of IP-datagrams that exceed their deadlines are not rejected straightaway (soft deadlines). However some other action is required so that a stream is given higher priority to increase probability of keeping deadlines. The "action" may be to raise the priority of the IP-datagrams, belonging to a stream, that is missing its deadlines. Feedback to the source of data cells (in the interconnect network) may be required. Data traffic may have different priorities within the data traffic class. Priority based service-class are assumed but where guaranteed services might be implemented on a higher level in the distributed router.

*3) In general, the arrival of IP-datagrams in a LAN environment is considered to be self-similar [11]. The traffic in the studied router may be similar to this. The arrival model of the control traffic cells is more directly related to arrival of IP datagrams than the arrival model of data traffic cells.

As can be seen in Table 16.3, all nodes communicate with all other nodes with equal probability. The data units in the network are called cells and are 48 – 64 bytes long. Speed is of concern in this application. Checking deadlines of e.g. control traffic cells consumes much time. Instead we assume that it is acceptable if cells are treated in FCFS (first come, first served) order but are differentiated by the three traffic classes in Table 16.2, and by priority in the case of data traffic. More or less predictable throughput is also assumed, depending on which traffic class.

The control packets contain information about how an IP-datagram requires to be handled at the destination module and how the source and destination ports of the interconnect network should be set up for the following data packets. Information in the control cells also co-ordinate functions in the interconnect network and are therefore not necessarily directly associated with data cells. The data cells will contain some ID that associates it with a control cell that arrived previously, together with destination and source ports in the interconnect network

16.6 Case definition and simulator setup

In all simulations the channel "wiring" is the same, i.e. the same definition of logical channels are used and each channel goes to the same node(s). However, the load of a set of channels may change for each data point. In [9], the full case definition is described. All three simulations are based on the "straight pipeline" case, as described in [9]. Each slot (smallest simulated time unit) is equivalent to 1 μs and corresponds to 1 kByte of data. Table 16.2 contains values and assumptions for three cases. The "straight pipeline" RSP case is assumed in the following simulations.

The traffic in the RSP case definition consists of three main types (in decreasing order of timeliness requirements): Control traffic, data traffic and other traffic. These types are conveniently services with the three data transports services offered by the CCR-EDF network protocol. The control traffic requires guarantees of timeliness and is therefore is serviced by the LRTC service. Typical qualities of the three types of traffic in the RSP case definition, found in [9], are summarised in Table 16.2.

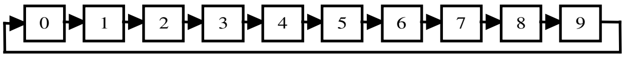

Figure 16.9: The "wiring" of BE–channels. Note that each arrow is a unicast channel.

Figure 16.9 depicts the "wiring" of the Best effort (BE) channels. These communicate the bulk of the data, in a pipelined fashion from node to node. The traffic is periodic with a period of 16 ms, i.e. 16 000 slots, and the default payload is 10 MByte, i.e. 10000 slots. During simulation, the payload may be varied from 1 MByte – 16 Mbyte. Best effort traffic has lower priority than LRTC, but higher priority than NRT traffic, see also Table 16.1. If the BE channels are used by themselves in a system, the set of BE channels have a maximum throughput of 10 packets per slot with 16 Mbyte payload. This is the theoretical limit and is also found by simulation.

Figure 16.10 depicts the wiring of LRTCs. The case study has a fixed set of LRTCs. These are organised as master / slave communication.

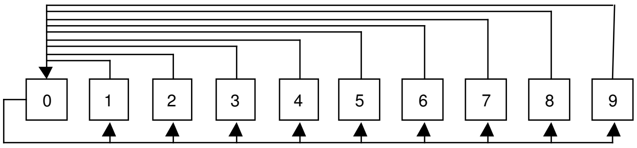

Figure 16.10: The "wiring" of the LRTCs. Node 0 is the master node and the others are slaves.

The data is control information and therefore the amount is less than BE traffic. The traffic is periodic with a period of 100g μs i.e. 100 slots, and default payload is 1 kByte, i.e. one slot. During simulation, the payload may be varied from 1 kByte to 16 kByte. The LRTC traffic class is a guaranteed service with the highest priority. When used by themselves in a system, the set of LRTCs have a maximum throughput of 1.1 packets per slot per channel with 11 kByte payload per packet. This result is found in simulation 3 performed in Section 7.3 and is also further explained here.

	[packets per slot]
Total throughput with the default load of the RSP case (Observe that the system is not saturated here)	5.97
Default LRTC throughput	0.10
Default BE throughput	5.26
Maximum throughput of LRTC traffic only	1.11
Maximum throughput of BE traffic only	10

Table 16.4: Important results, values taken from simulations. Observe that packets per slot refers to any and all types of traffic in the system unless otherwise stated. Observe also that more complex combinations such as maximum BE throughput with default LRTC load are not stated here.

The NRT (non real time) traffic is not organised in channels with fixed destinations. From each source the destination is uniformly distributed. Every node is a NRT source of equal load. The deadline of NRT traffic is fixed to 100 slots. In the simulator, NRT traffic arrival is always poisson distributed. NRT traffic always has the lowest priority. When used by itself in a system, the NRT traffic will have a maximum throughput of 2 packets per slot and will use all available capacity.

All "wirings" of channels, etc. are done according to the case study definition [9]. The case definition also states the traffic loads under normal operation. With the mix of traffic described in the case study, the total throughput (for all traffic types) is 5.97 packets per slot (see Table 16.4). In Simulation 2, Section 7.2 it will be seen that the load of the network can be increased to achieve an even higher throughput.

Each "simulation" consists of several (usually 16) data points. Each data point is an execution of the simulator with fixed parameters. To attain the curve, a parameter is varied and several runs later, the result is the data points that make up the curves in the presented figures.

Periodic traffic channels in the simulator are treated as follows. All traffic that a channel will send during one period will be generated and queued for sending at the start of the period. There is no "intelligent" functionality that smoothes the incoming traffic over the whole period. This lack of smoothing affects the maximum latency that a packet endures during a period. If the incoming traffic was smoothed, then the maximum would be closer to the average latency.

16.7 Simulations

In this section three main simulations are presented. Other simulations have also been done. These may be referred to only briefly, e.g. as numeric results, and not presented in full with diagram and discussion.

The first simulation concerns the latency and packet loss of BE traffic under increasing load of BE traffic. The second and third simulations have the mixture of traffic described in the case definition. In the second and third, the load of the BE traffic and LRTC traffic, respectively, is varied while the other is fixed. The effect on the mixture of traffic is studied. Important results from the simulations are summarised in Table 16.4. The "default" throughputs in the table are according to the load-levels found in the case study [9]. Observe that NRT traffic in the RSP case always strives to use all left over available capacity. Therefore it is meaningless to cite the throughput of this class.

16.7.1 Simulation 1

The "straight pipeline" of the case study definitions was implemented in the simulator and tested with different loads of BE traffic. The network setup for the simulation is as follows:

- All LRTCs at constant load: 1 data packet with a period of 100 slots period (the same as the RSP case definition).
- The load of each BE channel is varied in the interval 1 MByte – 16 MByte (6 % - 100 %).

- No Non Real time traffic in the system

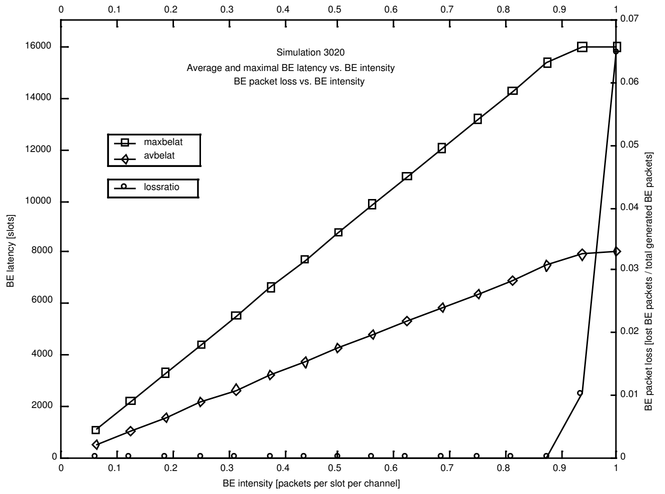

Figure 16.11: The figure shows the BE-traffic latency and BE-packet loss vs. the intensity of the BE-traffic. Regarding BE latency, observe that the dedline of the BE packets is 16000 slots.

The result of the simulation is shown in Figure 16.11. The figure shows three plots: Maximum and average BE latency vs. BE intensity and BE packet loss vs. BE traffic intensity. See Section 5.4 for a deeper discussion on traffic intensity. The maximum BE latency is the maximum latency that any BE packet was subject to during the simulation (for the respective intensity of BE traffic). As the BE traffic intensity increases, so does the maximum latency. The trend holds until the network cannot accept more BE traffic. At maximal BE intensity, 1 packet per slot per channel, the theoretical throughput (with no other traffic in the system) is 1 packet per slot per channel (total of 10 packets per slot for all BE channels). However, in the simulation, higher priority traffic also contends for capacity (LRTC traffic), and the throughput of BE traffic is therefore lower than the theoretical value. This can be observed since packets are lost at high intensity levels of BE traffic. At this point the trend of the latency curve changes and BE packets begin to be lost. The packet latency curve levels out (does not continue to increase) at a value of 16 000 slots. This is because the latency cannot be higher than the deadline of the packets, which is 16 000 slots. When a packet is queued for longer than its deadline, the packet is removed and considered lost. A maximum latency of 16 000 slots might seem to be long but one should remember that 16 Mbyte of data translates to 16 000 needed slots / packets per channel.

Regarding BE intensity, observe that the x-axis in Figure 16.11 indicates the ratio per BE channel, not total BE intensity. The total BE intensity would at its peak approach 10, i.e. the number of nodes in the system. Observe also that the average latency of the BE traffic is always roughly half that of the maximum throughput.

16.7.2 Simulation 2

In this simulation, all three types of traffic are generated during the simulation. How the traffic types effect each other is studied. In short the BE traffic is constant and the LRTC traffic is varied. The network set-up for the simulation is as follows:

- The LRTC set and behaviour of the set is the same as in simulation 1, i.e. constant.
- The set of best effort channels and behaviour of these are the same as in simulation 1, i.e. varied in the interval 1 MByte – 16 Mbyte (6 % - 100 %).
- The intensity of NRT data is enough to saturate the network. This means that although the other traffic classes have priority, NRT traffic will always be sent as soon as there is an opportunity. The intensity of the NRT traffic is constant throughout the simulation.

Figure 16.9 shows the result of the simulation. Observe that the concept of "throughput ceiling" is dependent on the traffic pattern, (see Section 8). In the figure, it can be seen that the highest priority traffic (LRTC) is not affected by the increasing level of BE traffic. Also note that the throughput of the NRT traffic decreases as the intensity of BE increases. In other words, prioritisation of different traffic classes in the simulator works as specified in the protocol.

As can be seen in the figure, the maximum total throughput (before any BE traffic is dropped) is approximately 7.7 packets per slot. The total throughput is the combined throughput of all traffic types.

16.7.3 Simulation 3

In this simulation, all three types of traffic are generated during the simulation. How the traffic types effect each other is studied. In short the BE traffic is constant and the LRTC traffic is varied. The network setup for the simulation is as follows:

- The load of the BE set is constant at 4 000 packets (4 MByte) per period of 16 000 slots (16 ms), (less load than the default RSP case).
- The load of LRTC traffic is varied between 1-16 KBytes per channel with a period of 100 slots.
- The intensity of NRT data is enough to saturate the network. This means that although the other traffic classes have priority, NRT traffic will always be sent as soon as there is an opportunity. The intensity of the NRT traffic is constant throughout the simulation.

Figure 16.13 shows the result of the simulation. Observe again that the concept of "throughput ceiling" is dependent on the traffic pattern, (see Section 8). As can be seen in the figure, the maximum total throughput (before any LRTC traffic is dropped) is approximately 2.8 packets per slot. Observe that in a real implementation of the network there would be admission control of LRTC traffic. Thus it would be impossible to reach a level where LRTC traffic is lost. The total throughput is the combined throughput of all traffic types. The maximum throughput of LRTC traffic (before packets begin to be dropped) is approximately 1.1 packets per slot per channel. This can be found in Figure 16.13. When LRTC traffic begins to be dropped, the payload is 11 kByte per period, i.e. the LRTC intensity is 0.11 packets per slot per channel (11 packets / 100 slots per channel). Here the LRTC throughput is 1.1 packets per slot.

As can be seen in the figure, the total throughput drops as the LRTC intensity increases. This is because there is decreasingly less traffic in the system that takes advantage of spatial reuse (pipelining), i.e. BE traffic. This phenomenon is further discussed in Section 8. As expected, throughput for the two lower priority traffic classes decrease as LRTC intensity is increased.

16.8 Discussion on throughput ceiling

In simulation 2 and 3 a concept of total system throughput is used. This is a measure of the number of data packets that are sent during each slot. Each data packet takes one slot to transmit. When comparing the throughput of the different traffic classes, it must be taken into account that the classes have different traffic patterns and therefore have different throughput ceilings. This "100%"-level varies depending on the traffic pattern. In a system, the throughput ceiling is constant and does not change unless the traffic pattern does so.

The following example illustrates this. The worst case is when all communication is destined one node upstream. Here the throughput

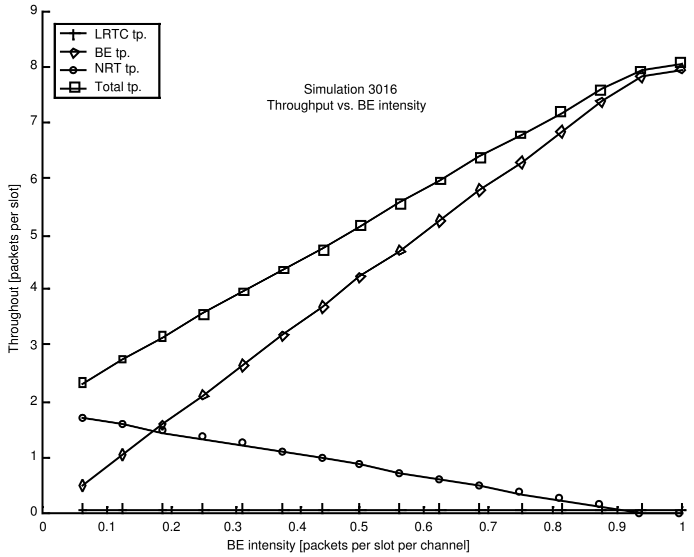

Figure 16.12: The throughput of the different traffic classes change as the BE intensity is increased. LRTC traffic load is constant.

can be at most one. I.e. all traffic is destined to one node upstream, and no traffic is destined to any other node in the network. Therefore no pipelining of non-overlapping transmissions can take place. The best case is when all communication is destined one node down stream. Here the throughput can be at most N, where N is the number of nodes in the network. In this case, the pipelining capability of the network is fully utilised. For traffic that is well-specified in channels, it is easy to find a value for throughput. However, if traffic is, e.g. poisson distributed, then the throughput can only be known statistically. The four different scenarios of data communications patterns are shown as explained below. Observe that they are discussed in increasing order of throughput.

- Scenario one occurs when all traffic is destined "furthest around the ring" i.e. to the node's upstream neighbour. This communication pattern does not suit the network topology under discussion (unidirectional pipelined ring). The level of throughput achieved is equivalent to that of a shared media network (e.g. a standard shared ring or bus), i.e. one data packet per slot.

- Scenario two occurs when the destination of all traffic is evenly distributed, i.e. on average the packets will travel half way around the

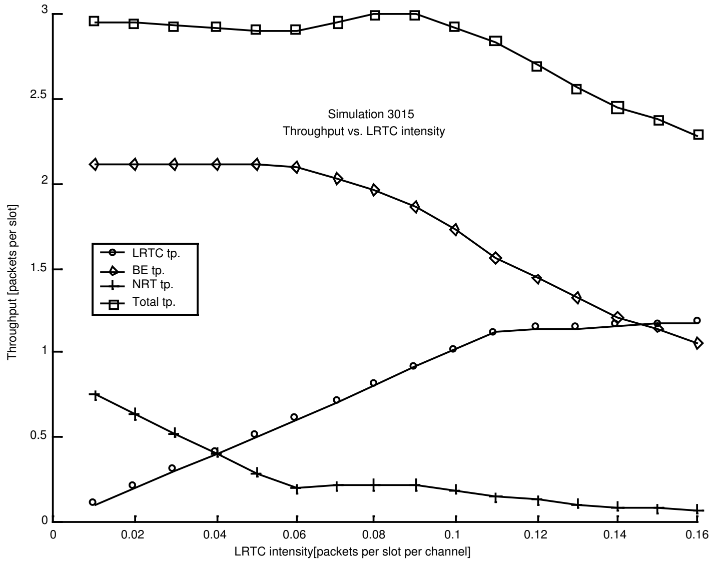

Figure 16.13: Throughput of different traffic when BE and NRT traffic stays constant as LRTC traffic varies.

ring. In this case the average throughput is two packets per slot. This is possible because of the pipelining feature of the network, where several transmissions may take place in non-overlapping segments.

- Scenario three is the radar case currently being discussed. Here, a large part of the communication is pure pipelined (the BE traffic), which is advantageous for total throughput. Therefore the total throughput will be larger than two. Observe that the total throughput depends on the traffic pattern. We have seen in the simulations that the total throughput with the radar case is 5.97 packets per slot, (see Table 16.4).

- Scenario four occurs when all traffic is destined to the next downstream neighbour. Here the pipelining feature of the network is optimally utilised. Throughput will be N packets per slot, where N is the number of nodes in the network.

16.9 Conclusions

The function of the CCR-EDF protocol has been verified by simulation. Also, the concept of having different traffic classes to differentiate between traffic has been tested in simulation, and shown to work. Results from the simulations of the radar signal processing case study show that the CCR-EDF network is an effective choice. Two applications for SANs, with support for heterogeneous real-time communication, has been described. For each application, a case study has been defined.

Acknowledgement

This work is part of M-NET, a project financed by SSF (Swedish Foundation for Strategic Research) through ARTES (A Real-Time network of graduate Education in Sweden).

Bibliography

[1] C. Bergenhem and M. Jonsson, “Fibre-ribbon ring network with inherent support for earliest deadline first message scheduling” *Proc. International Parallel and Distributed Processing Symposium., , (IPDPS 2002),* Fort. Lauderdale, FL, USA, 2002 15-19 Apr., pp. 92 - 98

[2] M. Jonsson, “Two fibre-ribbon ring networks for parallel and distributed computing systems,” *Optical Engineering,* vol. 37, no. 12, pp. 3196-3204, Dec. 1998.

[3] M. Jonsson, A. hlander, M. Taveniku, and B. Svensson, “Time-deterministic WDM star network for massively parallel computing in radar systems,” *Proc. Massively Parallel Processing using Optical Interconnections (MPPOI'96),* Maui, HI, USA, Oct. 27-29, 1996, pp. 85-93.

[4] M. Taveniku, A. hlander, M. Jonsson, and B. Svensson, “A multiple SIMD mesh architecture for multi-channel radar processing,” *Proc. International Conference on Signal Processing Applications & Technology (ICSPAT'96)*, Boston, MA, USA, Oct. 7-10, 1996, pp. 1421-1427.

[5] J. A. Stankovic, “Misconceptions about real-time computing,” *Computer*, vol. 21, no. 10, pp. 10-19, Oct. 1988.

[6] M. Jonsson, B. Svensson, M. Taveniku, and A. hlander, ”Fiber-ribbon pipeline ring network for high-performance distributed computing systems,” *Proc. International Symposium on Parallel Architectures, Algorithms and Networks (ISPAN'97)*, Taipei, Taiwan, Dec. 18-20, 1997, pp. 138-143.

[7] P. C. Wong and T.-S. P. Yum, ”Design and analysis of a pipeline ring protocol,” *IEEE Transactions on communications*, vol. 42, no. 2/3/4, pp. 1153-1161, Feb./Mar./Apr. 1994.

[8] M. Jonsson, C. Bergenhem, and J. Olsson, "Fibre-ribbon ring network with services for parallel processing and distributed real-time systems," *Proc. ISCA 12th International Conference on Parallel and Distributed Computing Systems (PDCS-99)*, Fort Lauderdale, FL, USA, Aug. 18-20, 1999, pp. 94-101

[9] C. Bergenhem, M. Jonsson, B. Gördén, and A. hlander, "Heterogeneous real-time services in high-performance system area networks - application demands and case study definitions," *Technical Report IDE - 0254, School of Information Science, Computer and Electrical Engineering (IDE), Halmstad University*, 2002.

[10] S. Agelis, S. Jacobsson, M. Jonsson, A. Alping, and P. Ligander, "Modular interconnection system for optical PCB and backplane communication," *Proc. Workshop on Massively Parallel Processing (WMPP'2002) in conjunction with International Parallel and Distributed Processing Symposium (IPDPS'02)*, Fort Lauderdale, FL, USA, April 19, 2002.

[11] W. E. Leland, M. S. Taqqu, W. Willinger, D. V. Wilson, "On the self-similar nature of Ethernet traffic," *IEEE/ACM Transactions on Networking*, vol. 2, no. 1, pp. 1-15, Feb. 1994.