Halmstad University Post-Print

Dynamic time-deterministic traffic in a fiber-optic WDM star network

Magnus Jonsson, Klas Börjesson and Magnus Legardt

N.B.: When citing this work, cite the original article.

©1997 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

Jonsson M, Börjesson K, Legardt M. Dynamic time-deterministic traffic in a fiber-optic WDM star network. In: Proceedings : Ninth Euromicro Workshop on Real Time Systems, June 11-13, 1997, Toledo, Spain. IEEE; 1997. p. 25-33.

DOI: <http://dx.doi.org/10.1109/EMWRTS.1997.613760>

Copyright: IEEE

Post-Print available at: Halmstad University DiVA

<http://urn.kb.se/resolve?urn=urn:nbn:se:hh:diva-2765>

Dynamic Time-Deterministic Traffic in a Fiber-Optic WDM Star Network

Magnus Jonsson, Klas Börjesson¹, and Magnus Legardt²

Centre for Computer Systems Architecture, Halmstad University, Halmstad, Sweden

Magnus.Jonsson@cca.hh.se, klasb@erv.ericsson.se, ehsmale@aom.ericsson.se, <http://www.hh.se/cca>

Abstract

A number of protocols for WDM (Wavelength Division Multiplexing) star networks have been proposed. However, the area of real-time protocols for these networks is quite unexplored. In this paper, a real-time protocol, based on TDM (Time Division Multiplexing), for fiber-optic star networks is presented. By the use of WDM, multiple Gb/s channels are achieved. Services for both guarantee-seeking messages and best-effort messages are supported for single destination, multicast, and broadcast transmission. Slot reserving can be used to increase the time-deterministic bandwidth, while still having an efficient bandwidth utilization due to a simple slot release method. The deterministic properties of the protocol are analyzed and simulation results presented.

1 Introduction

The WDM (Wavelength Division Multiplexing) technique offers extremely high aggregated bandwidth by the use of multiple wavelength channels in fiber-optic communication systems, and is foreseen to have a key role in future computer networks. A number of protocols for WDM star networks have been proposed. However, the area of real-time protocols for these networks is quite unexplored, with a few exceptions, e.g., [1].

This paper presents a high-performance fiber-optic network for distributed hard real-time systems. Typical target applications include future radar signal processing [2] and applications using artificial neural networks [3] [4]. Those applications will need multiple computation modules, each parallel itself, and a flexible real-time interconnection network with a bandwidth of multiple gigabit per second.

¹ The author is now with Ericsson Mobile Data Design AB, Göteborg, Sweden.

² The author is now with Ericsson Hewlett-Packard Telecommunications AB, Mölndal, Sweden.

The proposed interconnection network is a passive optical star network which utilizes WDM [5] [6] to implement multiple Gb/s channels. Each channel has a specific wavelength, i.e., color, of the light. A WDM system with 100 channels, each with a bit-rate of 622 Mb/s, has been demonstrated [7] and systems with 1000 channels are possible [8]. The passive optical star implements a fiber-optic multi-access network [9] where all incoming messages, in the form of light pulses, to the star are distributed to all nodes in the cluster by splitting the light. Tutorial overviews covering high-capacity fiber-optic networks for computer communication are found in [10-12].

In dynamic distributed real-time systems, messages may be classified into two categories [13]: best effort messages and guarantee-seeking messages. While best effort messages normally have soft deadlines, such that the system need only try its best to meet the deadlines, guarantee-seeking messages have harder timing constraints. If the communication system cannot guarantee the timing constraints of a guarantee-seeking message, the owner of the message should be aware of it immediately.

The TD-TWDM (Time-Deterministic Time and Wavelength Division Multiple Access) medium access method [14] is used in the proposed network, in a way that implements services for *both best effort messages and guarantee-seeking messages*. By the use of TDMA (Time Division Multiple Access) the access to each channel is divided into cycles of time-slots. Each node has a number of guaranteed slots to support guarantee-seeking messages. However, if there is no guaranteed messages in a node the slots will be released for best effort messages from other nodes (or the same node) according to a predetermined scheme.

Fixed-wavelength transmitters and tunable receivers are used in the network, see Figure 1. A fixed unit is always tuned to one specific wavelength channel, while a tunable unit can be tuned to an arbitrary wavelength channel. Each transmitter has a specific wavelength and, using the classification scheme given in [15], the network architecture can be described as FT¹-TR¹. FT¹-TR¹ stands

Figure 1: Passive optical star network with fixed transmitters and tunable receivers.

for one Fixed Transmitter and one Tunable Receiver per node. Other FT-TR networks can be found in [16] [17] and general information on WDM star networks in [15] [18] [9]. The receivers are tunable over the whole range of channels used in the network. This makes the network a single-hop network where any node is reached in a single hop [10]. FT¹-TR¹ is chosen because of the naturally embedded broadcast function, important in many computer systems, obtained when every receiver is tuned to the same transmitter channel.

The main function of the TD-TWDMA protocol is to allocate time-slots for either guarantee-seeking messages or best effort messages. The allocation is done using a deterministic distributed slot-allocation algorithm. The algorithm temporarily changes the predetermined scheme according to the current slot demands from each node. These slot demands are transmitted in advance on the same channels as the data is transmitted on, and contain information on which guaranteed slots to keep and which

to release. This type of WDM networks without a separate control channel are denoted as *non control channel* based networks. Networks where a separate control channel is used to reserve access to the data channels are denoted as *control channel* based networks. Other non control channel based networks are found in [17] [19] [20], while control channel based networks are found in [21-24].

For the sake of scalability we propose the star-of-stars topology when building larger networks [25] [26]. However, the TD-TWDMA protocol can still be used in each cluster and in the backbone [14].

The rest of the paper is organized as follows. In the second section the protocol is described. A presentation is given of the TD-TWDMA protocol, and the implementation of real-time services, in a network using TD-TWDMA, is described. The use of slot reservation to increase the amount of time-deterministic bandwidth is also discussed. The analysis of the network is concentrated to Section 3 and Section 4, respectively, presenting deterministic performance and simulation results. Section 5 is a conclusion and summary.

2 Protocol description

The notation used when describing the network is found in Table 1. Because each transmitter in the network has a specific wavelength, the number of wavelengths, C , hereafter denoted as channels, equals the number of nodes, $C = M$. The transmitter and receiver parts of the transceiver are independent and can work concurrently.

There are $2M$ queues in each transmitter, M for best effort messages and M for guarantee-seeking messages. For each of the two types of messages, one queue is for

M :	Number of nodes in the network
C :	Number of channels (wavelengths) in the network
m_i :	Designation of node i
B :	Effective bandwidth of each channel in the network
γ :	Slot length, including gap
μ :	Computing time of the distributed slot-allocation algorithm
S :	Total number of slots in a cycle
P :	Number of cycles deadline guarantees can be forecast
K :	Number of best effort multicast/broadcast packets to be sent in next cycle
s_i :	Designation of the i -th slot in a cycle
v_{ij} :	Index of the default high-priority owner (transmitter) of slot i in the receiver cycle in node j (v_{ij} is an element in matrix V)
w_{ij} :	Index of the default low-priority owner (transmitter) of slot i in the receiver cycle in node j (w_{ij} is an element in matrix W)
x_{ij} :	Element in the control slot matrix X , sent from one node, where status is given of slot i in the receiver cycle in node j
y_i :	Tells which queue to take a sub-message from for transmission in slot i (y_i is an element in matrix Y)
z_i :	Tells which channel to tune in for receiving in slot i (z_i is an element in matrix Z)
g_{ij} :	Number of high-priority slots in the i -th next cycle, where j is the corresponding queue (g_{ij} is an element in matrix G)
h_i :	Current number of sub-messages in each guarantee-seeking queue (h_i is an element in matrix H)
τ :	Packet latency: delay from the moment a message arrives at the transmitter buffer until the moment the transmission begins

Table 1: Notation when describing the network architecture and protocol.

Receiver cycles	Data slots											
	1	2	3	4	5	6	7	8	9	10	11	12
Node 1	2	2	3	4	3	2	3	4	4	2	3	4
Node 2	1	3	3	3	4	4	3	4	1	1	3	4
Node 3	1	4	4	4	1	2	1	4	1	1	2	4
Node 4	1	2	1	1	2	2	3	2	1	2	3	3

	Data slots											
	1	2	3	4	5	6	7	8	9	10	11	12
Transmitter cycle in node 1	2	-	4	4	3	-	3	-	2	2	-	-
	3								3	3		
	4								4			

Figure 2: The transmitter cycle, lower table, is filled by taking all owned slots in the receiver cycles, upper table, in all other nodes. Note that a multicast is possible in Slots 1, 9, and 10.

broadcast and $M - 1$ queues are for single destination messages (one for each of the other nodes). The broadcast queues are used for both true broadcast messages and for multicast messages (messages destined to more than one node but not to all). We define the size of an entry in the queues as that of a packet, i.e., a part of a message corresponding to one slot.

The TDM method used requires synchronized nodes. Harder synchronization will decrease the size of the necessary gap between the time slots. A method to achieve harder synchronization in TDMA networks (Time Division Multiple Access) by measuring the node-to-star propagation-delay is described in [25] [27]. A discussion of timing and dispersion in WDM star networks can be found in [28]. In this paper, the interconnection distances are assumed to be short and the propagation delay is hence neglected.

A method to decrease the effect of clock-recovery and tuning latencies in the receivers is presented in [14]. For this reason we assume that the tuning latency in the receivers can be neglected.

In the next subsection the function of the cycles is

Figure 3: A receiver cycle is partitioned into data slots and control slots, where the control slots are received one cycle in advance related to the data slots that they are carrying information about.

explained. In 2.2, the TD-TWDMA medium access method is described, followed by 2.3 where the implementation of real-time services is presented. Then, in 2.4, slot reserving is discussed.

2.1 Transmitter and receiver cycles

Because every transmitter has its own home-channel the only conflict that can appear is when two or more nodes want to transmit to the same node at the same instant. To prevent conflicts, slots in the network are therefore allocated in the receiver cycles where each slot will have a specific owner. This allocation is done by the distributed slot-allocation algorithm.

There is also one cycle running in each transmitter, but only to tell when and to whom the node is allowed to transmit. The transmitter cycle reflects the slots that the node owns in the receiver cycle of every other node (exemplified in Figure 2). In the receiver cycles, shown in the upper table in the figure, each slot in each receiver cycle can only be assigned to one transmitter at the same time. The lower table shows how the slots of a transmitter cycle are built up by copying its entries in the corresponding slots in each receiver cycle.

In Figure 3 it is shown how a receiver cycle is partitioned into data slots and control slots. Each node m_i , $1 \leq i \leq M$, is assigned one of the M control slots where it broadcasts control information to all other nodes m_j , $1 \leq j \leq M$ and $j \neq i$. The control slots are therefore identically assigned in every receiver cycle. When control slots have been gathered from all other nodes, allocation of the data slots in next cycle can be calculated using the distributed slot-allocation algorithm described in the next subsection.

To reduce the latency, the control slots are placed as late as possible in the cycle. However, there must be time for the distributed slot-allocation algorithm to be calculated before the beginning of the next cycle.

2.2 The TD-TWDMA protocol

The TD-TWDMA protocol consists of three steps:

1. Each node transmits a control slot

2. Each node separately runs the distributed slot-allocation algorithm
3. The nodes transmit and receive data slots

How the contents of the control slots is calculated and how incoming messages and buffers are handled, partly determines the real-time services and is described in the next subsection. When describing the distributed slot-allocation algorithm it is assumed, for simplicity, that all broadcasted control slots are received before the algorithm start. However, this is not a requirement in a real implementation because for each slot, the outcome of the distributed slot-allocation algorithm only depends on the control slot information from one node.

The slot-allocation algorithm is based on a predetermined allocation scheme that can be partly overloaded. As an example, the allocation scheme for a four-node system is shown in Figure 4. We do not call this scheme "reservation", because that term is used when describing the overloading of the scheme. Only data slots are shown and even if the control slots might be in the middle of the cycle we here assume they have index 13 to 16.

The total number of slots in a cycle is set to $S = M^2$ and the number of data slots is $M(M - 1)$. Each pair of rows represents one receiver cycle, where each number is the index of the transmitter that owns the corresponding slot. The high-priority row is the default scheme, but if the high-priority slot owner does not need the slot it is temporarily released for the current cycle, i.e., the cycle where the data should have been sent. This is done by a release message (described below) contained in the owner's control slot. The low-priority owner will then get the slot. If neither the high-priority nor the low-priority owner needs the slot, it will be unused. This is the cost of having a *simple* algorithm. However, compared to a static

TDM system an *efficient* bandwidth utilization is achieved due to the slot-release method.

In the predetermined allocation scheme, v_{ij} in matrix V identifies the index of the high-priority owner of slot i in the receiver cycle in node j . In the same way, w_{ij} in matrix W identifies the index of the low-priority owner. The high-priority owners are determined by

$$v_{ij} = (i - 1) \bmod M + 1 \quad (1)$$

for $1 \leq i \leq M(M - 1)$, $1 \leq j \leq M$ and $i \neq j$, while the low-priority owners are determined by

$$w_{ij} = ((i - 1) \div M + j) \bmod M + 1 \quad (2)$$

The release message mentioned above is actually a specified value in the control slot matrix, X , that each node transmits in the control slot. The matrix has the same size as matrix V and each element in the matrix gives information about the corresponding high-priority slot. All elements x_{ij} , $1 \leq i \leq S$, $1 \leq j \leq M$, in the matrix are set to zero except for those corresponding to high-priority slots, belonging to the node, which should not be released. Those elements are instead set to one. This means that a zero in the position of a high-priority slot, belonging to the node will release the slot.

When all the X matrices (one from each node) have been gathered by a node, two new matrices are composed, Y and Z . Each element in Y is used to determine what queue to take a packet from, for transmission in a slot, while an element in Z determines what channel to tune in at the beginning of a slot. The two matrices have elements corresponding to both data slots and control slots during one cycle, but for clearness the control slot elements are not treated here. The way the elements in Y is set and used to determine the current queue is described in the next subsection. When running the distributed slot-allocation algorithm, an element tells the kind of slot and is defined as:

$$y_i = \begin{cases} -1 - j & \text{Low - priority slot to node } j \\ 0 & \text{No transmission in slot } i \\ 1 & \text{High - priority multi - /broadcast} \\ 1 + j & \text{High - priority slot to node } j \end{cases} \quad (3)$$

where $1 \leq i \leq M(M - 1)$. Each element is set using the following algorithm for node m_k :

1. Let $y_i = -1 - j$ where j refers to the element $w_{ij} = k$ in matrix W , i.e., the receiver where the node has a default low-priority slot
2. If $x_{ij} = 1$ in matrix X from node m_l , where j refers to the element $w_{ij} = k$ in matrix W and $l = v_{ij}$, then set $y_i = 0$, i.e., the slot was not released by the high-priority owner

<i>Receivers</i>	<i>Data slots</i>											
<i>Priority</i>	1	2	3	4	5	6	7	8	9	10	11	12
1: High	-	2	3	4	-	2	3	4	-	2	3	4
1: Low	2	2	2	2	3	3	3	3	4	4	4	4
2: High	1	-	3	4	1	-	3	4	1	-	3	4
2: Low	3	3	3	3	4	4	4	4	1	1	1	1
3: High	1	2	-	4	1	2	-	4	1	2	-	4
3: Low	4	4	4	4	1	1	1	1	2	2	2	2
4: High	1	2	3	-	1	2	3	-	1	2	3	-
4: Low	1	1	1	1	2	2	2	2	3	3	3	3

Figure 4: Allocation scheme for the receiver cycles in a four-node system.

3. If $\sum_{j=1}^M x_{ij} > 1$ for matrix X from node m_i , then set $y_i = 1$, i.e., a high-priority multicast/broadcast slot belonging to the node
4. If $\sum_{j=1}^M x_{ij} = 1$ for matrix X from node m_i , then set $y_i = 1 + l$, where l refers to the element $x_{il} = 1$ in matrix X from node m_k , i.e., a high-priority single destination slot belonging to the node

Each element z_i , $1 \leq i \leq M(M-1)$, in matrix Z is set using the following algorithm for node m_k :

1. Let $z_i = w_{ik}$, i.e., the default low-priority owner
2. If $x_{ik} = 1$ in matrix X from node m_j , where $j = v_{ik}$, then set $z_i = v_{ik}$, i.e., the slot was not released by the high-priority owner

Since each node independently can perform the computations of the slot-allocation scheme we call it a distributed algorithm. No extra latency is required to return the result of the algorithm, which had been the case if, e.g., a master had calculated it.

2.3 Implementation of real-time services

In the description of how guarantee-seeking and best effort messages are passed through the transmitter, the following parts will be explained:

1. Treatment of arriving guarantee-seeking messages
2. Treatment of arriving best effort messages
3. The moment of transmitting a control slot
4. The moment when all control slots have been received
5. Action at the beginning of a data slot

The ability to give deadline guarantees to a message relies on knowing when we have guaranteed slots in the forthcoming cycles. Element g_{ij} , $1 \leq i \leq P$, $1 \leq j \leq M+1$, in matrix G holds the number of high-priority slots, belonging to the node, in the i -th cycle next to the currently running one, where $j=1$ for broadcast slots and $2 \leq j \leq M+1$ for single destination slots to node m_{j-1} . P is chosen at system design and tells us how far away in the future, relative to current time, deadlines can be guaranteed. Broadcast and multicast do not have to be separately treated here since the protocol does not allow the default scheme to be changed (by reservation) to have high-priority multicast to less than all other nodes.

By scanning g_{ij} , starting with $i = 1$ and with j set to the actual kind of an arriving message, we can see if there is enough high-priority slots, before deadline, to guarantee

the message to be sent in time. If not, the message will be rejected immediately and the owner will have time to handle the situation. If, instead, a guarantee can be given, each element in G corresponding to required slots is decremented by the number of required slots for that element, i.e., the sum of all decrements equals the number of packets in the message. In the case of a single destination message with too few available slots, the broadcast elements can also be scanned. Packets are always put in the guarantee-seeking queue corresponding to the element in G that was decremented, and in order of transmission to ease reassembling of the message.

Best effort messages might be transmitted in order according to, e.g., the *earliest deadline first* algorithm, but in the scope of this paper they are assumed to be transmitted in order of arrival. An arriving best effort message is hence just put in the right best effort destination queue. Multicast messages are put in the broadcast queue.

When it is time to send a control slot the following algorithm is executed in node m_k , $1 \leq k \leq M$:

if $g_{11} > 0$

$K = \text{MIN}(i, g_{11})$, where i is the number of packets in the best-effort broadcast queue

endif

count the packets in each guarantee-seeking queue to set

h_i , $1 \leq i \leq M+1$, $i \neq k+1$

$x_{ij} = 0$, $1 \leq i \leq M$, $1 \leq j \leq M(M-1)$

for $j = 1$ to M , $j \neq k$

scan x_{ij} starting with $i = 1$, and set $x_{ij} = 1$ in h_{j+1} positions (if possible), where the corresponding element in V is $v_{ij} = k$ and $v_{il} \neq k$, $1 \leq l \leq M$, $l \neq j$

endfor

do the same scan of x_{ij} for broadcast high-priority slots, first for the h_1 guarantee-seeking packets and then for the K best effort packets, setting $x_{ij} = 1$ for each corresponding destination that is set in the address field of a multicast (all destinations for broadcast), i.e., releasing slots to not specified destinations

send control slot

$g_{ij} = g_{i-1,j}$, $1 \leq i \leq P-1$, $1 \leq j \leq M+1$

g_{ij} , $i = P$, $1 \leq j \leq M+1$, are set to the corresponding number of default (after reservation) high-priority slots for each corresponding kind of destination

where K is the number of best effort multicast/broadcast packets that are to be sent in the next cycle. The elements in matrix H tell the current number of packets in each guarantee-seeking queue, where h_1 is the broadcast queue and h_i , $2 \leq i \leq M+1$, $i \neq k+1$, is the single destination queue to node m_{i-1} .

When all control slots have been received, the elements in Y are calculated as described in the last section. However, each element is modified to tell what queue to take a packet from when transmitting in the corresponding slot, and is, for Slot i , defined as:

$$y_i = \begin{cases} -1-j & \text{Best effort to node } j \\ -1 & \text{Best effort broadcast} \\ 0 & \text{No transmission in slot } i \\ 1 & \text{Guarantee-seeking broadcast} \\ 1+j & \text{Guarantee-seeking to node } j \end{cases} \quad (4)$$

where $1 \leq i \leq M(M-1)$. K of the elements in matrix Y , where $y_i = 1$, should be set to $y_i = -1$, beginning to scan the elements with $i = M(M-1)$, i.e., in descending order.

At the beginning of each data slot, y_i is used to determine which queue to take a packet from. By the definition of the protocol, there will always be a packet in the addressed queue when it is a guarantee-seeking queue or the best effort broadcast queue. However, the best effort single destination queues might be empty. In that case an empty packet is sent, only telling the receiver that there was nothing to send.

2.4 Slot reserving

A node can reserve slots to increase its guaranteed bandwidth. A maximum of $M(M-2)$ slots per cycle, slot 5 to 12 in the example in Figure 4, are allowed to be reserved. Slots s_i , $1 \leq i \leq M$, are not allowed to be reserved and slots s_i , $M^2 - M + 1 \leq i \leq M^2$, are control slots. When reserving slots, the corresponding high-priority entry (element in matrix V) in the receiver cycle, or cycles if broadcast is used, are exchanged with the index of the reserving node. To keep complexity down in the transmitters the corresponding slots in all receivers must be reserved (i.e., for broadcast) if the slots should be used for multicast messages.

The assignment of reserved slots can be changed during run-time either by higher layers, by a development system,

or by having slot-assignment schemes for several working modes stored in the nodes. If the reservation is done by higher layers, a reservation request to the nodes in question is sent in a control slot (together with matrix X).

3 Deterministic performance

The real-time services rely on time-deterministic communication. The deterministic bandwidth and the worst-case packet latency for this guaranteed bandwidth is analyzed below.

When a specific node is the only transmitter in the network, the bandwidth it gets is $B(M-1)/M$, where B is the channel bandwidth. This bandwidth is then divided into $B(M-1)/M^2$, which can be used for broadcasting, and $B(M-1)^2/M^2$ for point-to-point communication, i.e., $B(M-1)/M^2$ to each of all the other nodes. When competing with other nodes the broadcast bandwidth is always guaranteed and can be used for either single-destination, multicast, or broadcast messages.

If maximum allowed reservation in another node's receiver cycle is possible, the deterministic bandwidth is increased to $B(M^2 - 2M + 1)/M^2$ including the broadcast bandwidth. When some of the reserved slots are reserved in all receiver cycles, the deterministic broadcast bandwidth is also increased. A node that leaves all slots allowed for reservation, to other reserving nodes, will have the minimum guaranteed bandwidth possible: B/M^2 . However, all this bandwidth has broadcast capability.

The worst-case latency for a single packet gives an essential performance measurement of networks for real-time systems and will therefore be analyzed for the proposed network. The time it takes to run the slot-allocation algorithm, μ , sets the limit of how late in the cycle the control slots can be placed. As shown in Figure 5, the worst-case latency for node m , $1 \leq i \leq M$, is

$$\tau_{max} = S\gamma + M\gamma + \mu = (M+1)M\gamma + \mu \quad (5)$$

which is built up of: (i) the worst-case delay before the own control slot appears, i.e., one cycle as shown in the

Figure 5: Latency calculation.

Figure 6: Real-time performance plotted as fraction of messages that miss their deadlines, versus traffic intensity.

figure, (ii) the delay from the control slot till the first owned data slot. By fine-grain interleaving of the slots, where each node is assured to have one of the first M slots, we can minimize the second part of the latency so that node m_i , $1 \leq i \leq M$, never will have a slot later than slot s_i as its first slot in the cycle.

4 Simulation results

In addition to the worst-case analysis (deterministic performance), the average performance of the network has been analyzed through computer simulations for general traffic. Single star networks with 8, 16, and 32 nodes were simulated. Other assumptions were:

- A gap of one data slot between the last control slot and the next cycle, i.e., $\mu = \gamma$. To give an example of a real system, the slot duration was set to $\gamma = 1 \mu s$.
- All guarantee-seeking messages have a deadline 5 ms from the moment of generation.
- Uniform traffic was assumed, i.e., all nodes had equal probability of message generation and uniformly distributed destination addresses. Messages were generated according to a Poisson process and all messages were of single-destination type.
- Message lengths were exponentially distributed between 1 and 10 slots (discrete number of slots), with a length of 1 slot as highest probability.
- For simplicity, the propagation delay was neglected and no slot reservation was done.
- Infinite queue lengths were assumed.
- Packet generation rate is the message generation rate through the mean message length.

Figure 7: Latency for guarantee-seeking messages plotted versus traffic intensity.

- Latency is defined as the time elapsed from the arriving moment of a message until the last packet of the message leaves the transmitter.

First, the deadline missing percentage versus packet generation rate of guarantee-seeking messages is plotted in Figure 6. At moderate traffic intensities, no messages miss their deadlines for the given assumptions. The deterministic bandwidth fraction is $(M-1)/M^2$. This gives 10.9, 5.9, and 3.0 percent for 8, 16, and 32 nodes, respectively. As shown in the figure, guarantee-seeking messages begin to be partly rejected around these values. The plot is independent of packet generation rate of best effort messages, since guarantee-seeking messages always are prioritized.

In Figure 7, the mean latency for guarantee-seeking messages is plotted against packet generation rate. Again, the performance is independent of the amount of best effort traffic. Because messages that can not be guaranteed to meet their deadlines are discarded, the guarantee-seeking latency is upper bounded. The latency is rather uniform at low traffic intensities, but starts to grow earlier for larger networks because of the lower fraction of deterministic bandwidth.

The plot in Figure 8 shows the latency for best effort messages versus total packet generation rate (guarantee-seeking plus best effort). The traffic consisted of 10 % guarantee-seeking messages and 90 % best effort messages. The fraction of the bandwidth available for data packets is $(M-1)/M$, which gives 87.5, 93.8, and 96.9 percent for 8, 16, and 32 nodes respectively. However, when the guarantee-seeking traffic gets saturated, some of the total number of generated packets are discarded guarantee-seeking messages. Below saturation, the best effort latency is almost uniform.

Figure 8: Mean latency for best effort messages in a network with 10 % guarantee-seeking and 90 % best effort messages.

The mean best effort latency is plotted again in Figure 9, but versus the bandwidth utilization. The same ratio between guarantee-seeking messages and best effort messages was used. The plot looks very similar to the previous best effort latency plot in Figure 8. This is a consequence of the almost linear relationship between packet generation rate and bandwidth utilization below saturation.

5 Conclusions

A medium access protocol for WDM star networks in distributed real-time computer systems has been proposed. For the sake of scalability, the network architecture can be extended to a star-of-stars configuration. At the same time as dynamic real-time properties are supported, the protocol gives almost uniform latency nearly up to the theoretical saturation point. Deadline guarantees are supported where the underlying deterministic bandwidth can be changed dynamically through slot reserving. An efficient bandwidth utilization is obtained due to a simple slot release method.

6 References

[1] A. Yan, A. Ganz, and C. M. Krishna, "A distributed adaptive protocol providing real-time services on WDM-based LAN's," *Journal of Lightwave Technology*, vol. 14, no. 6, pp. 1245-1254, June 1996.

[2] M. Taveniku, A. Åhländer, 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.

Figure 9: Mean latency for best effort messages plotted versus bandwidth utilization. The traffic consisted of 10 % guarantee-seeking and 90 % best effort messages.

[3] E. W. Davis, T. Nordström, and B. Svensson, "Issues and applications driving research in non-conforming massively parallel processors," *Proc. of the New Frontiers, a Workshop of Future Direction of Massively Parallel Processing*. Scherson Ed., McLean, Virginia, pp. 68-78, 1992.

[4] B. Svensson and P.-A. Wiberg, "Autonomous systems demand new computer system architectures and new development strategies," *IECON '93: 19th annual Conference of the IEEE Industrial Electronics Society*, Maui, Hawaii, USA, Nov. 15-19, 1993, pp. 27-31.

[5] G. R. Hill, "Wavelength domain optical network techniques," *Proceedings of the IEEE*, vol. 77, no. 1, pp. 121-132, Jan. 1989.

[6] P. E. Green and R. Ramaswami, "Direct detection lightwave systems: why pay more?," *IEEE LCS*, pp. 36-49, Nov. 1990.

[7] H. Toba, K. Nakanishi, K. Oda, K. Inoue, and T. Kominato, "A 100-channel optical FDM six-stage in-line amplifier system employing tunable gain equalizers," *IEEE Photonics Technology Letters*, vol. 5, no. 2, pp. 248-251, Feb. 1993.

[8] T. S. Wailes and D. G. Meyer, "Multiple channel architecture: a new optical interconnection strategy for massively parallel computers," *Journal of lightwave technology*, vol. 9, no. 12, pp. 1702-1716, Dec. 1991.

[9] D. J. G. Mestdagh, *Fundamentals of Multiaccess Optical Fiber Networks*. Artech House, Inc., 1995, ISBN 0-89006-666-3.

[10] R. Ramaswami, "Multiwavelength lightwave networks for computer communication," *IEEE Communications Magazine*, vol. 31, no. 2, pp. 78-88, Feb. 1993.

[11] P. E. Green, "The future of fiber-optic computer networks," *Computer*, pp. 78-87, Sept. 1991.

[12] A. S. Acampora and M. J. Karol, "An overview of lightwave packet networks," *IEEE Network*, pp. 29-41, Jan. 1989.

- [13] K. Arvind, K. Ramamritham, and J. A. Stankovic, "A local area network architecture for communication in distributed real-time systems," *Journal of Real-Time Systems*, vol. 3, no. 2, pp. 115-147, May 1991.
- [14] M. Jonsson, A. Åhlander, 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*, Lahaina, HI, USA, Oct. 27-29, 1996, pp. 85-93.
- [15] B. Mukherjee, "WDM-based local lightwave networks part I: single-hop systems," *IEEE Network*, pp. 12-27, May 1992.
- [16] C. A. Brackett, "On the capacity of multiwavelength optical star packet switches," *IEEE LTS*, pp. 33-37, May 1991.
- [17] N. R. Dono, P. E. Green, K. Liu, R. Ramaswami, and F. F. Tong, "A wavelength division multiple access network for computer communication," *IEEE Journal on Selected Areas in Communications*, vol. 8, no. 6, pp. 983-994, Aug. 1990.
- [18] C. A. Brackett, "Dense wavelength division multiplexing networks: principles and applications," *IEEE Journal on Selected Areas in Communications*, vol. 8, no. 6, pp. 948-964, Aug. 1990.
- [19] A. Ganz and Z. Koren, "WDM passive star - protocols and performance analysis," *Proc. INFOCOM'91*, Bal Harbour, FL, USA, Apr. 1991, pp. 991-1000.
- [20] A. Ganz and Y. Gao, "A time-wavelength assignment algorithm for a WDM star network," *Proc. INFOCOM'92*, Florence, Italy, May, 1992, pp. 2144-2150.
- [21] K. Bogineni and P.W. Dowd, "A collisionless multiple access protocol for a wavelength division multiplexed star-coupled configuration: architecture and performance analysis," *Journal of Lightwave Technology*, vol 10, no. 11, pp. 1688-1699, Nov 1992.
- [22] I. M. I. Habbab, M. Kavehrad, and C. W. Sundberg, "Protocols for very high-speed optical fiber local area networks using a passive star topology," *Journal of Lightwave Technology*, vol. 5, no. 12, pp. 1782-1794, Dec. 1987.
- [23] M. Chen, N. R. Dono, and R. Ramaswami, "A media-access protocol for packet-switched wavelength division multiaccess metropolitan area networks," *IEEE Journal on Selected Areas in Communications*, vol. 8, no. 6, pp. 1048-1057, Aug. 1990.
- [24] R. Chipalkatti, Z. Zhang and A. S. Acampora, "High speed communication protocols for optical star coupler using WDM," *Proc. INFOCOM'92*, Florence, Italy, May 1992, pp. 2124-2133.
- [25] M. Jonsson, K. Nilsson, and B. Svensson, "A fiber-optic interconnection concept for scaleable massively parallel computing," *Proc. Massively Parallel Processing using Optical Interconnections, MPPOI'95*, San Antonio, TX, USA, Oct. 23-24, 1995, pp. 313-320.
- [26] M. Jonsson and B. Svensson, "On inter-cluster communication in a time-deterministic WDM star network," *Proc. 2nd Workshop on Optics and Computer Science (WOCS)*, Geneva, Switzerland, Apr. 1, 1997.
- [27] L. Bengtsson, K. Nilsson, and B. Svensson, "A high-performance embedded massively parallel processing system," *IEEE and Euromicro Conference, MPCS'94*, Ischia, Italy, May 1994, pp. 201-206.
- [28] G. Semaan and P. Humblet, "Timing and dispersion in WDM optical star networks," *Proc. INFOCOM'93*, San Francisco, CA, USA, 1993, pp. 573-577.