Halmstad University Post-Print

Time-deterministic WDM star network for massively parallel computing in radar systems

Magnus Jonsson, Anders Åhlander, Mikael Taveniku and Bertil Svensson

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

©1996 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, Åhlander A, Taveniku M, Svensson B. Time-deterministic WDM star network for massively parallel computing in radar systems. In: Proceedings of the Third International Conference on Massively Parallel Processing Using Optical Interconnections, October 27-29, 1996, Maui, Hawaii. IEEE; 1996. p. 85-93.

DOI: <http://dx.doi.org/10.1109/MPPOI.1996.559046>

Copyright: IEEE

Post-Print available at: Halmstad University DiVA

<http://urn.kb.se/resolve?urn=urn:nbn:se:hh:diva-2733>

Time-Deterministic WDM Star Network for Massively Parallel Computing in Radar Systems

Magnus Jonsson¹, Anders Åhlander¹, Mikael Taveniku^{2,3}, and Bertil Svensson^{1,2}

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

2. Department of Computer Engineering, Chalmers University of Technology, Göteborg, Sweden

3. Ericsson Microwave Systems AB, Mölndal, Sweden

Magnus.Jonsson@cca.hh.se, Anders.Ahlander@cca.hh.se,
micke@ce.chalmers.se, svensson@ce.chalmers.se

<http://www.hh.se/cca>

Abstract

In massively parallel computer systems for embedded real-time applications there are normally very high bandwidth demands on the interconnection network. Other important properties are time-deterministic latency and services to guarantee that deadlines are met. In this paper we analyze how these properties vary with the design parameters for a passive optical star network, specifically when used in a massively parallel radar signal processing system. The aggregated bandwidth and computational power of the radar system are approximately 45 Gb/s and 100 GOPS, respectively. The analysis is focused on the medium access control protocol, called TD-TWDM, for the time and wavelength multiplexed network. It is concluded that the proposed network is very well suited to this kind of signal-processing applications. We also present a new distributed slot-allocation algorithm with real-time properties.

1 Introduction

This paper presents an analysis of several properties of a high-performance fiber-optic network for massively parallel systems. Primarily, the network is targeted for embedded real-time systems and a case study is done of such a system with very high demands on bandwidth and computational power. The sample system is a MIMSIMD (Multiple Instruction Streams for Multiple SIMD arrays) system for signal processing in future phased-array antenna radar systems. Other MIMSIMD applications that the system may be suitable for are described in [1] [2].

The interconnection network was described on a conceptual level in [3], where it was shown that a passive optical star and an electronic true crossbar switch both can use the same proposed medium access method. The medium access method has real-time properties and is based on TDM (Time Division Multiplexing). In the case where the passive optical star is used, WDM (Wavelength Division Multiplexing) [4] [5] is also a part of the network architecture. The topology used is backbone-connected star clusters, star-of-stars, where each cluster consists of either a passive optical star or an electronic true crossbar. If the crossbar is used, it is simulating a passive optical star with the same number of wavelength channels as the number of nodes in the cluster. Tutorial overviews covering high-capacity fiber-optic networks for computer communication are found in [6-8].

In this paper we focus on the passive optical star cluster with fixed-wavelength transmitters and tunable receivers, 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 [9] the network architecture can be described as FT^1-TR^1 . FT^1-TR^1 stands for one Fixed Transmitter and one Tunable Receiver per node. Other $FT-TR$ networks can be found in [10] [11] and general information on WDM star networks in [9] [12] [13]. The receivers are tunable over the whole range of channels used in the cluster. This makes the cluster a single-hop network where any node in the same cluster is reached in a single hop [6].

FT^1-TR^1 is chosen because of the naturally embedded broadcast function, obtained when every receiver is tuned to the same transmitter channel, and the excellent mapping to pipelined systems. The output from each pipeline stage

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

then has its own wavelength, as long as each pipeline stage is mapped to exactly one node in the system. Even if the mapping is not exact, analysis of the data flow can still be done rather easily under certain circumstances. By tuning the receivers the pipeline order can be reconfigured during run time. When using the deadline-guaranteeing services of the proposed MAC-protocol (Medium Access Control), the time-critical pipelined data flow is guaranteed not to be disturbed by the transfer of non time-critical status information. As a denotation for the time-deterministic protocol for the FT¹-TR¹ network we will use TD-TWDM (Time-Deterministic Time and Wavelength Division Multiple Access).

We present an analysis of how latency, bandwidth, and bandwidth utilization vary with design parameters of the network. Both single-star networks and networks with backbone-connected clusters, see Figure 2, are analyzed. We also introduce a new distributed slot-allocation algorithm.

The rest of the paper is organized as follows. In the second section a radar signal processing case and its mapping on a MIMSIMD architecture are described. An overview is given of node architecture, algorithm mapping, and demands on the communication network. The network architecture is described in Section 3 with the focus on the medium access method, while the performance analysis is presented in the fourth section. The analysis is partly based on the performance demands described in Section 2. The paper is concluded in Section 5.

2 The Radar Signal Processing System

In the sample radar system, which is a ground based one, many individual antenna elements are arranged in a linear array. By controlling the relative phase of the received pulses the main lobe bearing can be controlled without any moving parts. There are 64 receiver channels from the antennas. This fact implies that huge computational demands are set on the signal processing.

Figure 2: Multiple passive optical stars topology.

In the following subsections we give a brief description of the signal processing, which is often called space-time processing. We also present a computation-module, based on eight 8 by 8 arrays of processing elements (PEs), and show how the signal processing can be mapped on a system consisting of a number of such modules. For a more detailed description of the radar system, and the algorithm mappings, refer to [14].

2.1 Signal Processing Chain

During a *Coherent Processing Interval* (CPI), which in this system is 16 ms, a number of pulses are transmitted and pulse samples (pulse echoes) are collected. A sample corresponds to a certain pulse, receiver channel, and time of collection after a transmitted pulse, i.e., the distance to target. The data collected during a CPI is here referred to as a *batch*. Every element in the batch is a complex number with 24 bit resolution of I and Q, respectively. The data from the receiver channels is fed into the signal processing chain illustrated in Figure 3. Into each channel complex numbers are fed at a rate of 2 MHz. Below, the different steps in the signal processing chain are described.

The *digital beamformer* creates lobes from the receiver channel signals by multiplying them with complex weight factors. The weight factors are chosen so that signals in the pointing directions of the lobes are added in phase. In this system, a number of equally distributed beams are created. Two computation-modules are used for this. A 64 point

Figure 3: The signal processing chain.

Figure 4: The computation-module architecture.

DFT is split up in sixteen 8 point DFTs. Each one of the processor meshes calculates one 8 point DFT by a matrix-by-vector multiplication. The two computation-modules are interconnected in a way which utilizes the symmetry in the 64 point DFT calculation. All the calculations are carried out in a systolic fashion, and thereby new data can be fed in at full rate.

The goal of the *pulse compression* is to collect all received energy from one target into a single range bin, i.e., we get a better resolution in range. The received phase or frequency-coded signal is correlated with the same code as used in the transmitter. In this system one FIR-filter per channel is used. The FIR filter has a length of 8. Here, one computation-module is used. The rows in the module's PE-arrays act as individual linear arrays. Each PE on a row holds an element of the pulse code. The pulses are fed in via the broadcast lines, and the systolically calculated compressed pulse is delivered on the right side.

The *doppler filter bank* transforms the pulse bins to velocity bins. During a CPI, a number of pulse samples are collected. For every single lobe, the pulses corresponding to the same range are processed using an FFT. The number and size of the FFTs are dependent on the working mode of the radar, but the product of the number of FFTs and the number of pulses in each FFT will be constant. In this case the FFT is performed as a number of 8-point FFTs, individually in each row of PEs. This scheme gives the possibility to use the FFT all the way. Here, the most time consuming operation will be when the longest FFTs are to be performed, but one computation-module will always be enough.

After the pulse compression and FFT have been performed, the phase information from the sample is no longer needed. The *envelope detection* removes, by an absolute-value calculation, the phase information from the samples. One computation-module takes care of this.

The *Constant False Alarm Ratio* (CFAR) processing is intended to reduce the number of false targets in each CPI, without loss in sensitivity. This combined constraint is usually solved by, based on some statistics, setting a threshold that makes the probability of a false alarm constant over time, thus the name Constant False Alarm Ratio. Here it is assumed that the CFAR calculates the mean value of a neighborhood to every resolution cell. The mean value, multiplied with a threshold constant, is then compared to the amplitude of the cell under test. In the simplest case one module can handle all the required computations. When more advanced statistical measures are used for the thresholds, two or more modules can be used to perform the calculations in parallel. This is possible due to the spatial locality of the algorithm, because only a small surrounding of each cell affects the result. In this system four modules are used.

The *extractor*, mapped on one module, calculates the center of mass of a target detected in several resolution cells at the same time.

2.2 Computation-Module Architecture

The computation-module chosen as a candidate for all the above described computation steps is shown in a simplified form in Figure 4. The module should be a good compromise between generality and efficiency in calculations. With generality we mean that the same type of module could be used in every part in the signal processing chain, and in every possible working mode of the radar. The core of the module consists of eight 8x8 PE-meshes. X-grid inter-PE connections are used but there are also row and column broadcast lines in the meshes. The PEs work in a SIMD fashion and are all controlled by the same control unit on the module (not shown).

2.3 System Architecture

In [14] a mapping of the computations on the modules is described and evaluated. The resulting signal processing system is shown in Figure 5, where also the bandwidth demands on the inter-module communication are given. As seen in the figure, the data flow is pipelined through the chain. In addition to the computational modules, two buffer-memory modules are needed to corner-turn the batch. Although the data flow has a very simple structure there can exist control messages in other directions. Also, the algorithms, and hence the data flow, can be changed when switching to a different working mode. A general

Figure 5: Description of the data flow between the modules.

network is therefore required. For simplicity, the maximum data flow from one module is assumed to be 6.0 Gb/s, excluding, e.g., error-checking codes. The latency budget for the whole chain is 100 ms in which 10 ms is included for inter-module communication. With the ten pipeline stages shown in the figure, including the antenna, the maximum allowed latency per link is $\tau_{max} = 1.0$ ms.

3 Network Architecture

Each optically isolated cluster in the star-of-stars network (Figure 2) has the same design and MAC protocol, TD-TWDM, as the single-star network (Figure 1). Therefore, we will first concentrate on describing a single cluster. The notation used when describing the network is found in Table 1.

The passive optical star implements a fiber-optic multi-access network [13] with M nodes. All incoming messages, in the form of light pulses, to the star are distributed to all nodes in the cluster by splitting the light. The FT¹-TR¹ transceiver configuration is used and each transmitter in the cluster is assigned a unique wavelength. Hence, 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 totally independent and can work concurrently.

In addition to WDM, TDM is used, so that the access to each receiver is divided into cycles of equal length. 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 TD-TWDM network are therefore allocated in the receiver cycles where each slot will have a specific owner. This is done by a distributed slot-allocation algorithm. There is also one cycle 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, as seen in Figure 6. 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. It is then up to the transmitter to decide, in each slot, if it should transmit to the node(s) it has access to or not.

Each cycle consists of S slots where s_i , $1 \leq i \leq S$, denotes a slot. The slot length is defined as $\gamma = \alpha + \beta$, where α denotes the part of the slot where data can be transmitted and β is a gap between the slots. Furthermore, we define B as the effective bandwidth of one channel, i.e., only accounting transmission during α . 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 [3] [15]. A discussion of timing and dispersion in WDM star networks can be found in [16]. In this paper, the interconnection distances are assumed to be short and the propagation delay is hence neglected. If β includes the time to establish bit-synchronization, a fast-locking clock-recovery will reduce β . By duplicating the opto-electronic and clock-recovery parts of the receiver, the time for clock-recovery can even be eliminated in β . This is done by having one clock-recovery circuit locked to the currently used channel while the other one recovers bit-synchronization for the channel that will be used in the next slot. In this way, the tuning time of the receiver only needs to be shorter than the duration, γ , of one slot, minus the clock-recovery time.

In the next subsection the MAC layer is described, followed by 3.2 where the distributed slot-allocation algorithm, used in the MAC layer, is presented. Then, in 3.3, inter-cluster communication is discussed.

- B : Effective bandwidth of each channel
 β : Gap between two slots
 γ : Slot length, including gap
 α : Slot length, excluding gap, i.e., length of the part where data can be transmitted
 μ : Computing time of the distributed slot-allocation algorithm
 S : Total number of slots in a cycle
 s_i : The i -th slot in a cycle
 v_{ij} : Index of the high-priority owner (transmitter) of slot i in the receiver cycle in node j
 w_{ij} : Index of the low-priority owner (transmitter) of slot i in the receiver cycle in node j
 τ : Latency: delay from the moment a message arrives at the transmitter buffer until the moment the transmission begins

Single-cluster network:

- M : Number of nodes in the network
 C : Number of channels (wavelengths) in the network
 m_i : Node i

Multiple-cluster network:

- M_j : Number of nodes in cluster j
 C_j : Number of channels in cluster j
 D : Number of channels in the backbone
 L : Number of clusters, which is the same as the number of nodes in the backbone
 E : Effective bandwidth of each channel in the backbone

Table 1: Notation when describing the network architecture.

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

↓

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

Figure 6: 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 Slot 1, 9, and 10.

3.1 Medium Access Control

The main function of the TD-TWDM protocol is to allocate the time-slots to appropriate nodes based on what bandwidth demands the nodes have. These bandwidth demands are transmitted in advance on the same channels as the data is transmitted on. This type of protocols for WDM networks without a separate control channel, and where each node has a pre-allocated home channel, can be described as *pre-allocation* based protocols by the

classification scheme proposed in [17]. The home channel is either used for all incoming messages or for all outgoing messages. Protocols where a separate control channel is used to reserve access to the data channels are classified as *reservation* based. Other networks using pre-allocation based protocols are found in [11] [18] [19], while networks using reservation-based protocols are found in [17] [20-22].

As mentioned above, the access to each receiver is divided into equally sized cycles. The cycle time is expressed as $S(\alpha + \beta) = S\gamma$. In Figure 7 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. Control information that is sent in a control slot informs the other nodes about the bandwidth demands for the next cycle. When information from all the nodes has been gathered, allocation of the data slots in next cycle can be calculated using the distributed slot-allocation algorithm described in the next section.

To reduce the latency, the control slots are placed as late as possible in the cycle. 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 the figure, the best-case latency for node m_i , $1 \leq i \leq M$, will be

$$\tau_{min} = (M - i + 1)\gamma + \mu + (i - 1)\gamma = M\gamma + \mu \quad (1)$$

The first term in the middle expression is the delay, from the beginning of the own control slot, until all control slots

are collected. The third term is the delay, from the start of the next cycle, until the first owned data slot appears. By fine-grain interleaving of the slots, where each node is assured to have one of the first M slots, we can minimize this delay 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.

The worst-case latency is

$$\tau_{max} = S\gamma + M\gamma + \mu = (S + M)\gamma + \mu \quad (2)$$

where the first term in the middle expression is the worst-case delay before the own control slot appears, i.e., one cycle as shown in the figure. As described below, each cycle consists of M^2 slots. Equation 2 can therefore be rewritten as

$$\tau_{max} = (M + 1)M\gamma + \mu \quad (3)$$

to express the worst-case latency.

3.2 Distributed Slot-Allocation Algorithm

In dynamic distributed real-time systems, messages may be classified into two categories [23], best effort messages and guarantee-seeking messages. While best effort messages normally have soft deadlines, where the system will 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 become aware of it immediately.

When using the TD-TWDMa protocol, each node can reserve a number of slots in the receiver cycle of each other node, for guaranteeing real-time services. This gives a known minimum number of slots to transmit data in. The medium access control layer can then offer the higher layers, not described here, services for guarantee-seeking messages. If the reserved bandwidth is sufficient for the message to meet its deadline, then a guarantee is given. If not, the message will be rejected immediately and the owner will have time to handle the situation.

If the owner of a reserved slot does not need to use it, the slot will be temporarily released for the current cycle,

i.e., the cycle where the data should have been sent. This is done by a release message contained in the owner's control slot. The released slots are then, according to a predetermined scheme, given to other nodes. 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. The change from one mode to another can then take place by just changing an offset pointer. If the reservation is done by higher layers, a reservation request to the node in question is sent in a control slot.

The slot-allocation algorithm is based on a predetermined allocation scheme that can be partly overloaded, using the reservation method mentioned above. As an example, the allocation scheme for a four-node system is shown in Figure 8. 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 are 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 released, just as described above for the reserved slots. 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.

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

$$v_{ij} = (i - 1) \bmod M + 1 \quad (4)$$

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

Figure 7: A cycle is partitioned into data slots and control slots, where the control slots are transmitted one cycle in advance related to the data slots that they are carrying information about.

<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 8: Allocation scheme for the receiver cycles in a four-node system.

$$w_{ij} = ((i - 1) \text{div } M + j) \bmod M + 1 \quad (5)$$

When a specific node is the only transmitter in the network cluster, the bandwidth it gets is $B(M - 1)/M$. This bandwidth is then divided into $B(M - 1)/M^2$, which can be used for broadcasting, and $B(M - 1)/M^2$ for point-to-point communication to each of all the other nodes.

A maximum of $M(M - 2)$ slots, slot 5 to 12 in the example, 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 in the receiver cycle, or cycles if multicasting is used, are exchanged with the index of the reserving node. The minimum high-priority bandwidth for a node with no reserved slots is B/M^2 where all bandwidth can be used for broadcasting. For a node with maximum allowed reservation in another node's receiver cycle, the high-priority bandwidth to that node will be $B(M^2 - 2M + 1)/M^2$.

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. Other motivations to use a distributed algorithm can be found in [24]. Smarter algorithms can be implemented to get higher bandwidth utilization, but at the cost of complexity. With this simple algorithm, updating of the table entries can start as soon as the first control slot is received. Only table indexing is then used in the data slots, in transmitters to choose between buffered messages, and in receivers to choose channel tuning. Double-buffered memories are implemented in the transmitters to prevent queuing problems. Despite the simple algorithm, and the fact that the computations can start immediately, μ is not zero. With a fast communication controller, a gap of one single slot can be enough, but in

the analysis here we assume that μ increases with one slot time, γ , for each extra control slot to compute on:

$$\mu = M\gamma \quad (6)$$

3.3 Inter-Cluster Communication

In a star-of-stars network there is still the same number of channels in each cluster as the number of nodes, $C_j = M_j$ for $1 \leq j \leq L$ where L is the number of clusters. The backbone cluster has the same MAC layer as the other clusters and consists of L gateway nodes and $D = L$ channels. We let E denote the bandwidth of each channel in the backbone. We also assume that node synchronization only exists inside a cluster. Messages are buffered between the clusters and the worst-case latency for each cluster on the route, Equation 3, should be added together. No additional latency appears in the gateway node as long as reserved slots are used. For communication between two end nodes, the worst-case latency will then be

$$\tau_{\max} = ((L + 1)L + (M_i + 1)M_i + (M_j + 1)M_j)\gamma + 3\mu \quad (7)$$

for the star-of-stars network where M_i and M_j are the numbers of nodes in the two largest clusters. In networks where $M_i = L$, $1 \leq i \leq L$, the worst-case latency is

$$\tau_{\max} = 3((L + 1)L\gamma + \mu) \quad (8)$$

4 Performance Analysis

Here we will further analyze how latency and bandwidth vary with the network design parameters. We derive several of the parameters from the demands of the radar system. For simplicity, we investigate networks with N^2 nodes. Because the network should be scalable, network sizes of $3 \leq N \leq 10$ are investigated, both single-star networks and star-of-stars networks where $M_i = L = N$, $1 \leq i \leq L$. The number of nodes is therefore set to N^2 instead of 13, but still with a data flow of 6.0 Gb/s from each node. Even if some nodes have lower requirements, they are all designed for the same capacity. To guarantee a node bandwidth of 6.0 Gb/s, maximum slot reservation is made for these pipeline stages. This gives that

$$6.0 \text{ Gb/s} = BM(M - 2) / M^2 \quad (9)$$

which, e.g., for $M = 16$ leads to $B = 6.9 \text{ Gb/s}$. The network size of $M = 16$ implies that the calculation holds for the thirteen nodes in the radar system, including the antenna, plus three spare nodes. With the calculated value of B we will, in addition to the reserved bandwidth, have a high-priority bandwidth of $B/M^2 = 27 \text{ Mb/s}$. This bandwidth is contained in the part of the cycle that is not allowed to be

Figure 9: Latency versus the number of nodes in the network.

reserved and can be used for control and status messages. According to the above calculations we have guaranteed bandwidths for both the dataflow and for control messages in the radar system.

In a purely pipelined system, mapped on a star-of-stars network, the required bandwidth of the backbone star will be

$$E = B(L - 1) / L \quad (10)$$

when $M_i = L$, $1 \leq i \leq L$. It is here assumed that all nodes have the same bandwidth requirements, except for the end node where the output dataflow is neglected.

Combining Equation 3 and 6 gives the worst-case latency

$$\tau_{max} = (M + 2)M\gamma \quad (11)$$

for a single-star network where we, for the radar system, can set $\tau_{max} = 1.0$ ms assuming a constant length of the pipeline chain independent of the number of nodes. With a slot length of $\gamma = 1.0$ μ s the maximum number of nodes is then $M_{max} = 30$ nodes. By using Equation 6 and 8 we get the worst-case latency

$$\tau_{max} = (L + 2)3L\gamma \quad (12)$$

for a star-of-stars network where $M_i = L$, $1 \leq i \leq L$. This gives a maximum total number of nodes $M_{max} = 289$ nodes, including gateway-nodes, where $M_{max} = L_{max}^2$ and $L_{max} = 17$. These figures are actually higher since only part of the communication is inter-cluster communication. We can also increase M_{max} by decreasing γ since the number of bits in one slot is rather high. At an effective bandwidth of 6.0 Gb/s there will be 6000 bits in each slot, which can be compared to ATM with its 424-bit cells. Both single-cluster and multiple-cluster worst-case latency, as functions of the number of nodes in the network, are shown in Figure 9. With sizes of the radar system of up to

Figure 10: Bandwidth utilization versus the number of nodes in the network.

30 nodes we only need a single cluster network while a star-of-stars topology scales up to a size of nearly 300 nodes.

The maximum bandwidth utilization depends both on the relation between β and γ , and on the number of nodes. In a single-star network the maximum bandwidth utilization will be

$$BU_{max} = \frac{M-1}{M} \cdot \frac{\gamma-\beta}{\gamma} \quad (13)$$

The left factor is the fraction of slots that are not control slots, i.e., data slots. In a star-of-stars network where $M_i = L$, $1 \leq i \leq L$, we have

$$BU_{max} = \frac{\sqrt{M}-1}{\sqrt{M}} \cdot \frac{\gamma-\beta}{\gamma} \quad (14)$$

where M is the total number of nodes in the network. We assume that the gap between the slots is $\beta = 0.1$ μ s. In Figure 10 the maximum bandwidth utilization is plotted versus the number of nodes, for both single-star and star-of-stars networks. For the assumed values, the rightmost factors in both Equation 13 and Equation 14 limit the bandwidth utilization to 90%. Since a node can receive from one of only two possible source nodes per non-reserved slot, reservation may be needed to achieve such a high bandwidth utilization. With this in mind, the figure shows how a high bandwidth-utilization is reached already at moderate network sizes both for single-star and star-of-stars networks.

5 Conclusions

A passive optical star network with a time-deterministic protocol, TD-TWDMA, has been presented. The MAC

layer utilizes a distributed slot-allocation algorithm and can give guarantees for both latency and bandwidth. We have shown that the network meets the latency demands for a sample pipelined radar-system even with rather long time-slots. The network architecture scales with the requirements of the radar system up to nearly 300 nodes, each consisting of hundreds of processors, without breaking the latency budget. Also, the maximum bandwidth utilization is high, despite the long gap between the slots assumed in the analysis.

6 Acknowledgement

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

7 References

- [1] 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.
- [2] 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.
- [3] 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.
- [4] G. R. Hill, "Wavelength domain optical network techniques," *Proceedings of the IEEE*, vol. 77, no. 1, pp. 121-132, Jan. 1989.
- [5] P. E. Green and R. Ramaswami, "Direct detection lightwave systems: why pay more?," *IEEE LCS*, pp. 36-49, Nov. 1990.
- [6] R. Ramaswami, "Multiwavelength lightwave networks for computer communication," *IEEE Communications Magazine*, vol. 31, no. 2, pp. 78-88, Feb. 1993.
- [7] P. E. Green, "The future of fiber-optic computer networks," *Computer*, pp. 78-87, Sept. 1991.
- [8] A. S. Acampora and M. J. Karol, "An overview of lightwave packet networks," *IEEE Network*, pp. 29-41, Jan. 1989.
- [9] B. Mukherjee, "WDM-based local lightwave networks part I: single-hop systems," *IEEE Network*, pp. 12-27, May 1992.
- [10] C. A. Brackett, "On the capacity of multiwavelength optical-star packet switches," *IEEE LTS*, pp. 33-37, May 1991.
- [11] 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.
- [12] 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.
- [13] D. J. G. Mestdagh, *Fundamentals of Multiaccess Optical Fiber Networks*. Artech House, Inc., 1995, ISBN 0-89006-666-3.
- [14] M. Taveniku, A. Åhlander, 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.
- [15] L. Bengtsson, K. Nilsson, and B. Svensson, "A high-performance embedded massively parallel processing system," *IEEE and Euromicro Conference, MPC'S'94*, Ischia, Italy, May 1994, pp. 201-206.
- [16] G. Semaan and P. Humblet, "Timing and dispersion in WDM optical star networks," *Proc. INFOCOM'93*, San Francisco, CA, USA, 1993, pp. 573-577.
- [17] 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.
- [18] A. Ganz and Z. Koren, "WDM passive star - protocols and performance analysis," *Proc. INFOCOM'91*, Bal Harbour, FL, Apr. 1991, pp. 991-1000.
- [19] 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.
- [20] 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.
- [21] 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.
- [22] 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.
- [23] 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.
- [24] C. Bohm, M. Hidell, P. Lindgren, L. Ramfelt, and P. Sjödin, "Fast circuit switching for the next generation of high performance networks," *IEEE Journal on Selected Areas in Communications*, vol. 14, no. 2, pp. 298-305, Feb. 1996.