C. Bergenhem, M. Jonsson, B. Gördén, and A. Åhlander, "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.

Heterogeneous Real-Time Services in High-Performance System Area Networks - Application Demands and Case Study Definitions

Carl Bergenhem[1], Magnus Jonsson[1], Bengt Gördén[2], and Anders Åhlander[3]

Technical report no. IDE263

1. School of Information Science, Computer and Electrical Engineering, Halmstad University, Halmstad, Box 823, S-301 18, Sweden. {Carl.Bergenhem, Magnus.Jonsson}@ide.hh.se, http://www.hh.se/ide

2. SUNET/KTH-NOC, Stockholm, Sweden

3. Ericsson Microwave Systems, Mölndal, Sweden

Abstract

To be able to verify the feasibility of high-performance networks, it is essential to evaluate them according to specific application requirements. At the same time, specifications of quite general, or understandable, application requirements are needed for the ability to make repeated analyses on different networks. Especially, heterogeneous real-time requirements must be defined to be able to analyze networks to be used in future applications. In this report, we introduce two application fields where system area networks (SANs) supporting heterogeneous real-time services are highly desirable if not required: radar signal processing and large IP routers. For each application field, a case study with heterogeneous real-time communication requirements is defined. No case studies are presented in this report. Instead, they are defined for later evaluations to determine how suitable networks are for applications with heterogeneous real-time communication requirements.

1. Introduction

To be able to verify the feasibility of high-performance networks, it is essential to evaluate them according to specific application requirements. At the same time, specifications of quite general, or understandable, application requirements are needed for the ability to make repeated analyses on different networks. Especially, heterogeneous real-time requirements must be defined to be able to analyze networks to be used in future applications.

In this report, we introduce two application fields where system area networks (SANs) supporting heterogeneous real-time services are highly desirable if not required: radar signal processing and large IP routers. For each application field, a case study with heterogeneous real-time communication requirements is defined. No case studies are presented in this report. Instead, they are *defined for later evaluations to determine how suitable networks are for applications with heterogeneous real-time communication requirements.*

An example of other applications with heterogeneous real-time communication requirements is information and entertainment systems in vehicles connected by a high-performance network. The bandwidth demands are lower than the formerly mentioned applications. However, since new applications, with real-time requirements hard to predict in advance, might be added to the system by the vehicle owner or on his/her request, the network must support heterogeneous real-time services.

The rest of the report is organized as follows. Radar signal processing is introduced in Section 2, followed by related case study definitions in Section 3. In the same manner, large IP routers are introduced in Section 4, followed by a related case study definition in Section 5. The report is then summarized in Section 6.

2. Radar signal processing applications

Modern radar systems are often based on the use of a phased array antenna, i.e., an antenna with multiple fixed antenna elements (and digital beam forming) instead of a moveable antenna. The 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 such 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. After calculation of one algorithmic step in a radar signal processing chain the data should be transferred to the node(s) responsible for the next algorithmic step and/or be exchanged (many-to-many communication to corner-turn the data cube) between a number of nodes cooperating on several algorithmic steps. 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.

A number of different working modes are possible in a radar system. The task of one mode can, for example, be to scan the whole working range, while the task of another mode can be to track a certain object. Normally, the algorithm mapping and communication patterns are different for two different modes. Support for dynamic changes of the communication pattern is therefore needed. Those changes are, however, not made more often than that the reestablishment of circuit-switched channels might be fast enough. The maximum distance between two modules is typically below 2 m (intra-rack communication), but it might be valuable if it is possible to include a near-antenna system a little bit longer away.

In addition to a pure data delivery, powerful services like low-level support for many-to-many communication patterns, real-time debugging, reliable transfer, and process synchronization are valuable. This is especially addressed by the radar industry since they are striving for engineer-efficiency [Åhlander and Taveniku]. Examples of motivations are: (*i*) long development time for a low number of sold systems today, (*ii*) updates of the systems or product families are made several times during their life-time, and (*iii*) it is often hard to employ enough good engineers.

3. A radar signal processing case

The 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.

	Communication pattern	Delay Bound	Traffic amount	Arrival model
Control traffic	Out: broadcast In: many-to-one (Master / Slave)	100 µs 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 1: Assumptions for the traffic in the RSP case study.

- 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 1, 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.

*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 (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.

Table 1 specifies parameters of the traffic in the RSP network. Parts of the assumptions presented in Table 1 are also found in [Jonsson et al. 1996] [Agelis et al. 2002]. 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 [Jonsson et al. 1996]. 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. [Jonsson et al. 1996] [Agelis et al. 2002].

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 1500 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.

3.1 Communication patterns

In Table 1, 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 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 discussed in the following sections.

3.2 Irregular pipeline

Figure 1 depicts an example of an irregular pipeline. The difference between the two mentioned pipelines in [Jonsson et al. 1996] and [Agelis et al. 2002] 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 1 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 1 where Node 7 multicasts to Nodes 9 through 11. Also, a many-to-

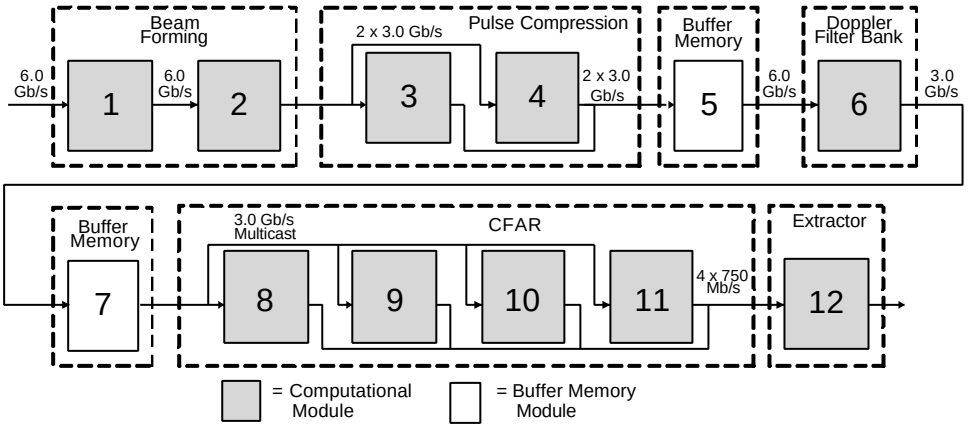

Figure 1: An irregular pipeline chain. Data flow between the modules in the radar signal processing chain.

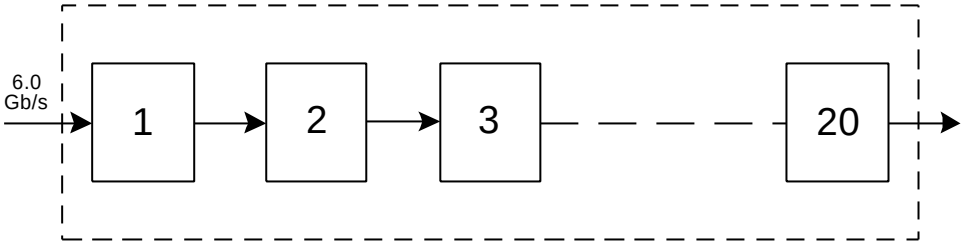

Figure 2: The straight pipeline

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.

The requirements on communication capacity between each processing step can be seen in Figure 1. At most the required capacity from one node is 6.0 Gb/s, which corresponds to the incoming data cubes. The maximum total communication latency for one data cube is 10 ms.

3.3 Straight Pipeline

For the straight pipeline, we assume that the bulk of the data communication is to the next neighbour (see Figure 2). 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.

3.4 SPMD

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 3. 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.

In the RSP algorithm, in a certain step, the radar data may be processed simultaneously as modules communicate. For a following step the data may also be processed in parallel but sometimes parallel in another dimension. Between such steps, the data must be redistributed. This is called corner turning and is, mathematically speaking, a matrix transposition. Depending on the type of the following step, requirements for data communication in an SPMD processing model varies. At best it is the same data that the module will process during the next step. At worst the data cube must be corner turned. The complete RSP algorithm contains two corner turns per data cube. For simplicity, we neglect other radar data traffic.

The latency for the SPMD model is the duration from when the data cube arrives until the result of the RSP algorithm is available. Because a new data cube is generated every 16 ms, the latency of the RSP must be less or equal to this. We assume that there is no overlap of communications and processing and that the 16 ms latency is composed of two corner turns and also processing. It is further assumed that communication and processing share the latency equally. This implies that the processing modules will be idle half of the time and also that the communications network will be idle half of the time. Therefore we over dimension the processing modules by a factor of two (compared to the straight pipeline case), which gives a total of 40 modules. The processing time, t_{proc}, is assumed to be 8 ms. Moreover, t_{proc} is

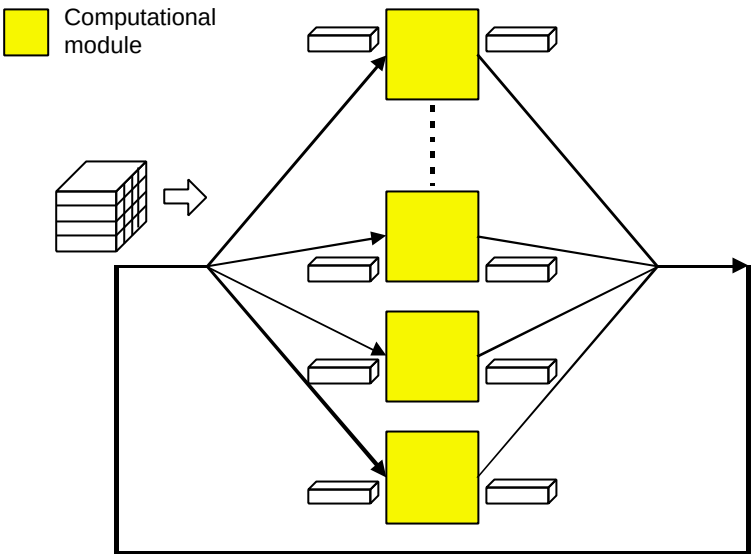

Figure 3: If the SPMD model is used, all modules work together on one stage in the signal processing chain at a time.

shared between the processing steps in the RSP algorithm. The time for one corner turn, t_{CT}, is 4 ms. By assuming no overlap the communication requirements are sufficiently over dimensioned. More about over dimensioning networks for SPMD can be found in [Teitelbaum 1998].

The case in Figure 1 states data cubes will be of size 96 Mbit. A corner turn with 40 modules requires 1 600 message transfers (also accounting for communication to the same module) with a payload of 60 kbit. Since corner turns account for the bulk of utilisation in the communication network, this gives an indication for dimensioning the payload of each data transfer.

3.5 Timeliness requirements

A broad reflection about the different processing models is that the two pipelined models have a total latency that is 100 ms, thus processing of data cubes is overlapped. With SPMD, the latency is equal to the period of new incoming data cubes (16 ms), thus all processing must be completed before a new data cube arrives and processing may not be overlapped.

From Table 1 three classes of timeliness requirements can be identified: hard deadlines, soft deadlines, and no deadline. So why cannot all these traffic classes be served by a single guaranteed hard real time service? The answer is that it would be uneconomical, concerning network capacity. The service that offers guaranteed service might for example rely on an analysis in which an assumption is made about the minimum network throughput. However, the actual network throughput may, on average during runtime, be higher than the level specified for guaranteed traffic. This is to say that even though the network is at full capacity with guaranteed traffic, there may still be capacity in the network that can be used for traffic classes that don't require guarantees.

4. Large IP routers

When building large IP routers (or similar data or telecommunication equipment), possibly with hundreds or even thousands of input and output ports, multiple racks are used. 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. It can also be the case that each rack

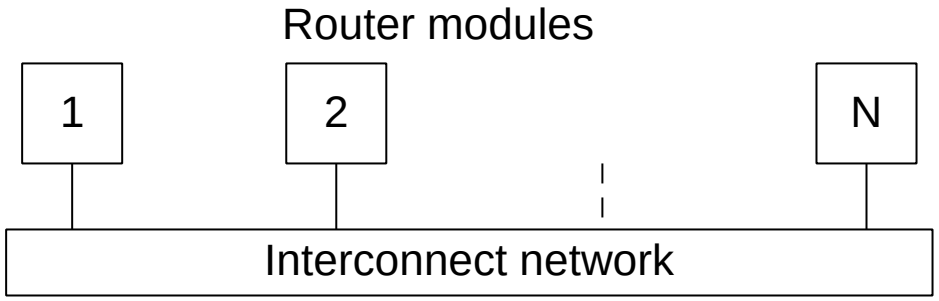

Figure 4: A block diagram of a cluster of router modules that form the "super router"

consists of both line cars and routing cards. Anyhow, the racks must be interconnected. For flexibility in design, operation, and maintenance of the system, a general SAN with real-time support is desirable. The complete system takes the form of a distributed router.

The main traffic through the SAN, i.e., IP datagrams segmented into fix-sized cells, should have a low average latency at normal loads, while still not be too large at higher loads. In addition to the normal data traffic, the SAN to coordinate the operation of the distributed router must carry a lot of control traffic. The traffic through the SAN has heterogeneous real-time demands in two senses: (*i*) different traffic classes that the IP datagrams belong to, and (*ii*) different kinds of control traffic coexisting with the data traffic. The distance between two nodes in the SAN of a distributed router can be assumed to be less than 10 m.

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 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 2, 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

	Communication pattern	**Delay bound**	**Traffic amount**	**Arrival model**
Control traffic	Evenly distributed	10 µs 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 µs max and 20 µs 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 2: The various traffic types in the network.

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 [Leland et al. 1994]. 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 1, 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 1, 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.

6. Summary

Two applications for SANs, with support for heterogeneous real-time communication, has been described. For each application, a case study has been defined. This gives the opportunity for the research community to evaluate SANs in terms of their suitability for this kind of applications, and for heterogeneous real-time communication in more general terms as long as the requirements are somewhat similar.

References

[Agelis et al. 2002] 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.

[Jonsson et al. 1996] 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.

[Leland et al. 1994] 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.

[Teitelbaum 1998] K. Teitelbaum, ”Crossbar tree networks for embedded signal processing applications,” *Proc. Massively Parallel Processing using Optical Interconnections (MPPOI'98),* Las Vegas, NV, USA, June 15-17, 1998, pp. 200-207.

[Åhlander and Taveniku] A. Åhlander and M. Taveniku, “Engineer efficient parallel systems - a preliminary requirements specification,” *Technical Report 1/0363-FCP 104 825 Uen, Ericsson Microwave Systems, Mölndal, Sweden*, 2002.