Research Report IDE - 0777, School of Information Science, Computer and Electrical Engineering (IDE), Halmstad University, Sweden, Nov. 2007.

Network Component Architectures from a Real-Time Perspective

Xing Fan and Magnus Jonsson

CERES, Centre for Research on Embedded Systems
School of Information Science, Computer and Electrical Engineering, Halmstad University, Halmstad, Sweden,
Box 823, S-301 18, Sweden. {Xing.Fan, Magnus.Jonsson}@ide.hh.se, http://www.hh.se/ide

1. Introduction

In this report, we survey different architectural features of network components for packet-switched networks. Especially, we have a real-time perspective in the study. We also discuss how different architectural features vary with different implementations of two standards, RapidIO and switched Ethernet.

2. Real-Time and Dependable Application properties

Some general application properties of interest for real-time embedded applications are given in Table 1.

The real-time literature discusses two types of real-time guarantees, *deterministic or probabilistic.* If the guaranteed real-time service is deterministic, it means that it is predictable and suitable for hard real-time systems, since normally both a guaranteed minimum throughput and a bounded end-to-end delay is offered. A guaranteed probabilistic service, on the other hand, is said to only guarantee to meet a specified QoS with a certain probability. It is more difficult to provide deterministic guarantees by a real-time system including communications, because many factors, for example, routing and difficulties in having centralized control/scheduling, make the real-time analysis more complex.

A real-time application is typically constrained by a set of explicitly expressed QoS parameters, namely *delay, throughput and jitter.* Deterministic guarantees are made on a worst-case analysis, whereas probabilistic guarantees are often connected to average behavior and hence yield a higher utilization of the system resources. If the offered service is deterministic, it is said to be predictable and suitable for hard real-time systems, since normally both a guaranteed minimum throughput and a bounded end-to-end delay is offered.

According to the arrival pattern, traffic can be classified as being *periodic, aperiodic, or sporadic*. Periodic traffic is typically found in control applications where the periodicity is strictly governed by control principles or other stability criteria. In contrast, aperiodic traffic does not have any restrictions on their periodicity. Sporadic traffic has a minimum inter-arrival time between invocations. Aperiodic and sporadic traffic are typically triggered by events such

Constraints		Description
Demanded guaranteed real-time services	Guaranteed deterministic service	The time constraints must be hundred percent guaranteed
	Guaranteed probabilistic service	Only guarantee to meet a specified QoS with a certain probability
QoS constraints	Bounded delay	The end-to-end delay of the message transmission must be bounded
	Throughput	The amount of transmitted data in a specified amount of time.
	Bounded jitter	The maximum tolerable derivation from the periodicity constraint
Traffic models	Periodic	Data is released regularly at fixed rates
	Aperiodic	Data is released irregularly at some unknown and possibly unbounded rate.
	Sporadic	Data is released irregularly with some known bounded rate.
	Burstiness constrained	The maximum burstiness is bounded

Table 1. Real-Time application properties.

as pressing a button. Another arrival pattern, typically used for Internet traffic, has strict constraints on its burstiness. Obvious, a periodic traffic arrival pattern is more predictable than the other types of arrival pattern. Moreover, most of the hard real-time traffic in the embedded domain is periodic traffic.

Application areas for dependability include business-critical applications and embedded systems, for example, financial transactions and fly-by-wire. Some general application properties of interest for dependable embedded applications are given in Table 2. Dependability is not treated very much in this report but is an important field.

3. Network Properties

In this section, we summarize some general network properties, with a focus on network components properties.

The properties chosen to be studied for real-time perspectives in this report are listed in Table 3.

Packet-switching and *circuit-switching* are two major variants of switching technologies. A typical example of service based on circuit-switching technology is normal telephone service. When using circuit-switching, the dedicated resource,

Constraints		Description
Safety		A property of a system that it will not endanger human life or the environment
Fault		Cause of an error, e.g., an open circuit, a SW bug or an external disturbance
Error		Part of the system state which is liable to lead to failure, e.g., a wrong value in a program variable
Failure		Delivered service does not comply with the specification, e.g., a cruise control locks at full speed
Fault types	Random faults/ Systematic faults	Random faults are usually physical faults, and Systematic faults are design faults in HW or SW
	SW faults/ HW faults	Software faults are caused by specification mistakes and implementation mistakes. Hardware faults are caused by specification/implementation mistakes, external disturbances or component defects.

Table 2. Dependable application properties.

for example, a physical channel is allocated and then the communication with guaranteed bandwidth can start. Disadvantages of circuit switching are long setup times and low bandwidth utilization when the channel is idle for a long time since the bandwidth normally cannot be reused. In contrast, packet switching refers to networks in which the data is divided into packets before being sent. Each packet is then transmitted individually and can even follow different routes to its destination. Most modern Local Area Network (LAN) and Wide Area Network (WAN) networks, including IP networks, X.25 networks and switched Ethernet networks are based on packet-switching technologies. Compared with circuit-switching networks, in a packet-switched network, resources can be used more efficiently but handling real-time traffic is more difficult.

In packet-switched networks, a switch can be designed to forward frames in one of two ways: *store-and-forward* or *cut-through*. Store-and-forward switches receive each packet into a memory buffer and examine it for errors before transmission. Cut-through switches instead begin forwarding the frame as soon as the switch has read the destination address. As a result, cut-through switches exhibit shorter latency (i.e., forwarding delays) than store-and forward switches as long as a packet can be forwarded directly (if the output port is busy, the packet is stored as in store-and-forward). Store-and-forward switches have the advantages: 1) no error propagation by performing error detection, 2) being able to handle speed conversions and support heterogeneous networks.

Properties			Description
Switching technology	Packet-switching	Store-and forward	A switch will wait to forward a frame until it has received the entire frame.
		Cut-through switching	A switch starts forwarding the frame as soon as the switch has read the destination address.
	Circuit-switching		The dedicated resources (physical links) are allocated for transmission between two parties.
Routing	Static/fixed routing		The routing decision is rarely changed
	Dynamic/adaptive routing		The best next-hop is adaptively chosen based on, e.g., congestion statistics.
Buffering strategy	Input-queued (IQ)		A switch stores packets in the input ports
	Output-queued (OQ)		A switch sends packets directly to output port queues
	Virtual Output Queued (VOQ)		Each queue stores those packets which have arrived at a given input port and are destined to a given output port
Time overhead	Extra delays introduced by software		e.g., preparation of data packets, protocol stacks, or data copy, etc.
	Extra delays introduced by hardware		e.g., memory access
Data overhead	Extra data		e.g. protocol overhead and packetisation

Table 3. Network properties relates to real-time perspective.

The routing decision, or path selection, in a packet-switched interconnection network can be made in one of two ways: static routing or dynamic routing. Static routing means that the routing decision is rarely changed. In contrast, dynamic routing is more sophisticated and involves adaptive routing, meaning the best next hop is adaptively chosen based on, e.g., congestion statistics. Although adaptive routing can utilize resources better in the network than static routing, it increase the implementation cost and complicates real-time analysis.

In packet-switched networks, since one output port can be the destination for multiple packets, some packets must be queued in the switch to be sent later (so-called output blocking). Therefore, the switch performance depends not only on the switch fabric speed, but also on the queuing and arbitration that decides which packets to send and which to buffer. Therefore, according to the buffering

strategy, switches can be classified as: *Input Queued (IQ) switches*, *Virtual Output Queued (VOQ) switches* and *Output Queued-(OQ) switches* (other switch architectures exist but are not treated here). An IQ switch, queuing packets at its input port, requires low complexity and few circuits. However, the FCFS nature and the IQ strategy result in a so called Head-of-Line (HOL) blocking phenomenon. That is, when a packet of a certain buffer at the input cannot be switched to an output port because of contention, the rest of the packets in that buffer are blocked by that HOL packet, even if there is no contention at the destination output ports for those packets. To overcome the HOL-blocking problem, many IQ switches are controlled by sophisticated scheduling algorithms at centralized schedulers, which restricts the design of the switch architecture. OQ and VOQ can be used to remove the HOL blocking. VOQ is an efficient yet simple buffering strategy where each queue stores those packets which have arrived at a given input port and destined to a given output port. Thus, packets entering input ports can successfully reach output ports only after competing for output ports with other input ports. OQ strategy means that one queue is maintained per output port and packets in each output queue to the output link are scheduled in FCFS. Although OQ switches require high performance on the switch circuits, they have the best average performance [Karol et al. 1987].

The performance of the switching network depends on many factors, including technologies, software architecture and traffic characteristics. Concerning the performance, the overhead introduced by the network must be considered. There are two different types of overheads, *time overheads* and *data overheads*. Time overheads are the extra delays introduced by software (data copy and packets preparation) and/or hardware (memory access). Data overheads are introduced by the protocol overheads and packetisation.

Moreover, power consumption is an important aspect and switch fabric contributes significantly in the total power consumption. In a switch fabric circuit, the power is dissipated on three components: 1) the internal node switches, located on the intermediate nodes between input and output ports, 2) the internal buffers, and 3) the interconnect wires that connect node switches. The power consumptions on these three components changes differently under different traffic loads and configuration. The switch fabric power consumption has been analyzed [T. T. Ye et al. 2002]. The important observations includes: 1) interconnect contention induces significant power consumption on internal buffers and the power consumption on buffers will increase sharply as throughput increases; 2) For switch fabrics with a small number of ports, internal node switches dominate the power consumption, for switch fabrics with a large number of ports, interconnect wires will gradually dominate the power consumption.

In summary, to have deterministic behavior, keep the cost low and performance high, while maintaining flexibility such as having support for different bit-rate links, packet-switched networks with store-and-forward switches, fixed routing and OQ/VOQ switches are good candidates.

Some properties from a dependability perspective are given in Table 4 for completeness, but are mostly out of scope of this report.

Constraints		Description
Hardware redundancy		Redundant networks or links.
Time redundancy		Retransmission
Information redundancy		Error correcting
Data overhead	Extra data	e.g. protocol overhead and packetisation

Table 4. Network properties relates to dependability perspective.

4. Evaluation of Available Network Components

In this section, we will summarize the properties of available network components and give an evaluation study on several examples.

4.1 RapidIO switches

A recent networking standard for high-performance embedded systems is RapidIO [RapidIO] [Fuller 2005]. The specific RapidIO switch we have studied [Tundra manual 2006], uses a combined input-output buffering technology, that is, virtual output queues at each input port and small output buffers for rate matching and control symbol embedding. Therefore, RapidIO has HOL blocking avoidance. To be specific, special arbitrary algorithms at both the ingress and egress sides of the internal switch fabric and crossbar switching are implemented to avoid HOL blocking.

RapidIO switches support both store-and-forward and cut-through, which can be configured individually for each port. For multicast or broadcast, RapidIO switches allocate a broadcast buffer. The operation mode for such traffic is store-and-forward. For dependability purpose, Tundra-specific RapidIO switch implementations support dead link timer expiration to detect possible faults.

RapidIO has built-in mechanisms for acknowledgement and retransmission of packets. It is even possible for retransmitted packets to get priority over other packets. However, it is an open question how to perform a real-time analysis of both the retransmission scheme in whole and the prioritization mechanisms. A real-time analysis framework need to be developed, with which the probability of meeting delay bounds can be calculated. Moreover, it must be investigated how retransmissions will affect the delay of following packets and how the queue populations will be affected (to not get dropped packets).

The performance measurement is reported in [Tundra manual 2006]. Tundra's implementation provides functions to monitor the queue depth to detect bottleneck traffic in the RapidIO interface. The information is stored in special registers that can be accessed remotely.

4.2 Gigabit Ethernet switches

Ethernet switches have been generally categorized into the following three market classes: the node/workgroup switch, the segment switch and the

backbone switch. Since most of switches today provide 10 Mbits/s to 100 Mbits/s switching, they will be in store-and-forward mode.

When priority is supported, eight priority levels can be used. However, the standard says that two priority queues are enough to implement, while eight priority queues are needed for a one-to-one mapping from the priority level given by the priority field in a frame. As an example of an Ethernet switch chip, Intel Media Switch IXE2424 has four queues per port, which can, e.g., be reconfigured to be mapped on certain priority levels [Intel Media Switch 2001]. The Epoch MultiLayer Switch Chipset from Music Semiconductors is another example, which has eight priority queues per output port [Music Epoch 1999]. Another question regarding priority queuing is whether it is strict priority queuing or it is combined with some kind of fairness queuing. The real-time scheduling analysis presented in [Fan 2007] [Fan and Jonsson 2005] assumes strict priority queuing but it should be possible to adapt the analysis if knowing the guaranteed minimum throughput and the worst-case extra delay before the highest priority queue can be considered to be served with a rate that guarantees the minimum throughput. When using the real-time analysis, guaranteed real-time traffic is assumed to always using the highest priority level. The eight priority queues per output port in the Epoch MultiLayer Switch Chipset can be used either for strict priority queuing or for eight traffic classes scheduled by Weighted Round Robin (WRR) [Music Epoch 1999]. The WRR ensures some fairness by giving each traffic class a least guaranteed fraction of the capacity. The analysis presented in [Fan 2007] [Fan and Jonsson 2005] also gives us the opportunity to calculate the minimum needed buffer space (queue size) for the real-time traffic class. To be able to guarantee that no buffer overflow will occur, we must know the queue sizes in the switch chip. If no dedicated memory is given to each queue, using some kind of shared-memory architecture, we need to ensure that there still is a minimum guaranteed amount of memory for the queue that handles the real-time traffic.

It is important that the switch chip supports wire speed switching, meaning that the switch has no bottlenecks but can forward traffic to all ports independent of the traffic pattern. Most switch chips support wire speed and clearly states that in the data sheets. Moreover, it is important that it can be guaranteed that no head-of-line (HOL) blocking occurs and this is sometimes clearly stated as for the Epoch MultiLayer Switch Chipset from Music Semiconductors [Music Epoch 1999]. For other switch chips it might be necessary to study the queuing architecture to find out whether it is free of HOL blocking.

Some open questions are whether the standardized flow-control is something that must be considered in the real-time analysis. Another study would be to investigate if the VLAN functionality often supported by Ethernet switch chips can help to better support real-time traffic.

5. Design Space

In this section, we will now address design space issues that are affected by the application and network properties discussed earlier. Table 5 illustrates our view of the design space.

Constraints		Description
Traffic regulator	Source node	Shape the traffic according to certain constraints.
	Switch	
Traffic arrival model	Periodic model	The minimum inter-arrival time of the data traffic is specified
	(r, b)-model used in Network Calculus	An upper-bound to the long term average rate of traffic flow and the maximum burstiness of traffic are specified
Queuing strategies	Priority-based queuing	Sort the queued packets according to priorities, for example, Earliest Deadline First.
	FCFS queuing	First Come First Serve, used in standard Ethernet swtiches
Service disciplines	Work-conserving	transmission will occur as long as there are packets eligible for transmission
	Non-work-conserving	transmission may not occur even if there are packets eligible for transmission
Real-time analysis	Communication scheduling	Modeling the traffic with adapted task model and then using scheduling technique to check schedulability
	Network Calculus	to estimate the worst-case delay for traffic regulated by (r, b)-model and sorted with FCFS-order

Table 5. Parameters in the design space.

In a packet-switched network, each packet traverses a number of *hops* from its source towards the final destination. Traversing a hop comprises passing through a switching controller. More specifically, for each hop that is traversed, a packet is transferred from the incoming link, through the switching fabric and to the output queues of the outgoing link if the OQ switching is used. After each hop, the packet is stored in a switch (or router). The choice of queuing architecture, traffic handling etc is essential for the QoS characteristics.

Real-time communication often relies on some kind of traffic regulators at the source node, to ensure the traffic source obey to its predefined traffic characteristics. Moreover, because of jitter, the variation of delay can be accumulated for each hop and therefore make a worst-case analysis very pessimistic. A lot of the real-time research concern about implementing some policing mechanism to regulate the injection rate in the switches. It should be

noted that adding traffic regulators in switches significantly increases the cost and implementation complexity.

To assess real-time performance, it is important to have the knowledge of the arrival traffic model. One widely used model for real-time traffic is the periodic model, that is, the minimum inter-arrival time of the traffic is specified. Another traffic arrival model is described with the aid of so-called arrival curves obeying (r, b)-model, which quantify constraints on average rate and the maximum burstiness of the traffic flow. In the embedded networking domain, the majority of hard real-time traffic is periodic.

Regarding approaches to handle real-time communication, one is to schedule the traffic according to, e.g., relative deadline or priority. However, such sorting functions also result in added cost and modification since many standard packet-switched network components only support FCFS.

Queuing service disciplines can be divided into *work-conserving* service disciplines and *non-work-conserving* service disciplines. Using a work-conserving service discipline, transmission will occur as long as there are packets eligible for transmission. Note that a work-conserving service discipline maintains a good utilization, and FCFS-queuing supported by standard components belongs to the work-conserving category. With a non-work-conserving service disciplines instead, transmission may not necessarily occur even if there are packets eligible for transmission. This can be good with regard to jitter and hence the size of jitter eliminating buffers. An example of well-known work-conserving service disciplines is Delay-EDD [Ferrari and Verma 1990], whereas Jitter-EDD is an example of a non-work-conserving service disciplines [Verma et al.1991].

Guaranteed deterministic services offer a hundred percent guarantee of the stated QoS level by having an *admission control* mechanism to verify that the specified requirements can be met. An admission controller is run to check that each switch on the path can guarantee the specified QoS. The idea behind admission control is *real-time analysis*. There are two widely used analytical schemes for such purpose, *communication scheduling* and *Network Calculus (NC)*. Communication scheduling, similarly to task scheduling, is to look on the transmission medium as a limited shared resource, to model the traffic with adapted task model and then to check the schedulability on all links in the routing path and to calculate the worst-case end-to-end communication delays. Although there is the need for using network components with FCFS-queuing, schedulability analysis for FCFS-queued periodic real-time traffic has not being deeply investigated. The objective of NC is to estimate the worst-case delay for traffic regulated by (r, b)-model and sorted with FCFS-order. There are several limitations of NC. One is that it only deals with FCFS-queuing. Another weakness is that the burstiness constraints indicate the requirement of using traffic shapers in the source nodes and the switches, which, of course, will increase the implementation complexity and cost. The third weakness is that NC cannot be used directly for periodic traffic, unless the periodic model being transformed into (r, b)-model. Moreover, pessimism will be introduced by such model transformation.

6. Conclusion

In this paper, we have discussed architectural features of network components from a real-time perspective. We have identified several things to have in mind when evaluating a specific implementation.

Even if a solid real-time scheduling analysis framework has been developed and reported in earlier publications, there are still open questions. One thing to investigate is how treat flow-control in the real-time analysis, while another is how to possibly use the VLAN functionality often supported by Ethernet switch chips to better support real-time traffic. For complex queuing architectures or architectures where special mechanisms are implemented to avoid HOL blocking, further investigation is needed if the data sheets do not give detailed information on how the performance is affected. Another open question is how to synchronize the updating of routing tables and treat this in the real-time analysis.

Also needed are real-time analysis methods for retransmission schemes and multicasting. Retransmission schemes can be used both end-to-end and on a per-hop basis and it must be investigated how retransmissions affect the real-time analysis and how to form protocols for which retransmissions do not hazard the stated guarantees for ordinary transmissions. A real-time analysis framework need to be developed, with which the probability of meeting delay bounds can be calculated. Moreover, it must be investigated how retransmissions will affect the queue populations to not get dropped packets.

References

[Fan 2007] X. Fan, "Real-time services in packet-switched networks for embedded applications," *Doctoral thesis, Department of Computer Engineering, Chalmers University of Technology, Göteborg, Sweden*, June 2007.

[Fan and Jonsson 2005] X. Fan and M. Jonsson, "Guaranteed real-time services over standard switched Ethernet," *Proc. of the 30th Annual IEEE Conference on Local Computer Networks (LCN'2005),* Nov. 15th-17th, 2005, Sydney, Australia.

[Ferrari and Verma 1990] D. Ferrari and D. Verma, "A scheme for real-time channel establishment in wide area network," *IEEE Journal on Selected Areas in Communications,* vol. 8, no. 3, Apr. 1990, pp. 368-379.

[Fuller 2005] S. Fuller, *RapidIO – The Embedded System Interconnect*. John Wiley & Sons Ltd., 2005, ISBN 0-470-09291-2.

[Intel Media Switch 2001] Intel Media Switch IXE2424 10/100+Gigabit L2/3/4 Advanced Device, *Data Sheet, Rev. 1.2, Intel*, Nov. 2001.

[Karol et al. 1987] M. J. Karol, M. G. Hluchyj, and S. P. Morgan, "Input Versus output queuing on a space-division packet switch," *IEEE Transaction on communicatioons*, vol. 35, no. 12, pp. 1347-1356, Dec. 1987.

[Music Epoch 1999] Epoch MultiLayer Switch Chipset, *Preliminary Data Sheet, Rev. 1.4, Music Semiconductors*, 29 Mar. 1999.

[RapidIO] RapidIO Trade Association: "RapidIO architecture specification, Release 1.3," http://www.rapidio.org, last checked 2007-05-01.

[T. T. Ye et al. 2002]. T. T. Ye, L. Benini and G. D. Micheli, “Analysis of Power Consumption on Switch Fabrics in Network Routers”´, Proc. Of DAC, June 10-14, 2002, New Orleans, Louisiana, USA.

[Tundra manual 2006] *Tsi578 Serial RapidIO Switch User Manual*. Preliminary Version, Tundra, Dec. 2006.

[Verma et al. 1991] D. C. Verma, H. Zhang, and D. Ferrari, "Delay jitter control for real-time communication in a packet switching network," *Proc. of 6th IEEE International Conference on Communications for Distributed Applications and Systemss,* pp. 35-43, Apr. 1991.