BENCHMARKING SPACEWIRE NETWORKS

Session: Networks & Protocols

Long Paper

Asaf Baron, Isask'har Walter, Ran Ginosar and Isaac Keslassy
VLSI Systems Research Center, Elec. Eng. Dept., Technion, Haifa 32000, Israel

Ofer Lapid
Israel Ministry of Defense

E-mail: {ab@tx, zigi@tx, ran@ee, isaac@ee}.technion.ac.il, ofer.lapid@gmail.com

ABSTRACT

Measuring and comparing performance, cost, and other attributes of SpaceWire networks is a significant challenge. A new benchmark for SpaceWire-based satellites is presented. The benchmark contains a potential architecture of a multi-mission satellite, as well as specifications of the traffic flow. The proposed benchmark is demonstrated on an OPNET-based network simulation for various network configurations. The simulated SpaceWire network supports priority in multiple alternative manners: (1) a simple SpaceWire network that ignores priority, (2) a network that supports packet priority according to the SpaceWire specifications, (3) a non-standard support for N-Char interleaving on multiple virtual channels, and (4) support for both packet priority and N-Char interleaving.

1. INTRODUCTION

SpaceWire [1][2][3] has been designed to facilitate high-performance onboard data handling systems (DHS), to help reduce system integration time and costs, to enhance the reliability and efficiency of space systems, to promote compatibility between DHS and subsystems, to encourage re-use of DHS across several different missions and to facilitate spacecraft housekeeping and maintanance.

Payload processing involves several functions: controlling instruments, calibrating them, collecting data from instruments, storing the instrument data, processing and compressing the data and sending the data to ground via a down-link.

SpaceWire supports different architectures by using routers and point-to-point links that can be configured in different manners, customized and tuned to the requirements of specific missions. Links and routers may also be used redundantly, to enhance reliability, fault tolerance and availability.

The SpaceWire specification is still evolving. SpaceWire parts and networks are being proposed, and new amendments are being discussed [4][5][6]. However, no benchmark has been published in order to enable the comparative testing of standard variants or proposed networks and components.

In this paper we present a spacecraft model that can be used as a benchmark for SpaceWire implementations, networks, parts and protocols. The benchmark comprises typical spacecraft components such as sensors, processing units, mass-memory units and down-link telemetry systems. It defines the number of active components of each type and the data communication bandwidth that they require. The benchmark is

demonstrated by performance simulations of several alternative SpaceWire networks that could be employed for the benchmark spacecraft.

Benchmarks are necessary for performance evaluation, for comparison of network features and topologies, and for the development of simulation methods, common terminology and accepted figures of merit. So far, on-board spacecraft communications have typically been implemented without benchmarks and extensive simulation since they were based on ad-hoc connections of buses and dedicated serial links and on empirical design based on time division multiplexing.

In Section 2 we present the proposed benchmark. Section 3 describes the simulator used in this paper, and section 4 presents the simulation results, showing that the benchmark helps in identifying problems in various network configurations. We note that in this paper we have no intention of finding the best configuration for the network, but merely present the demands for it and demonstrate, by considering several alternative configurations, the usefulness of the benchmark and simulations.

2. THE BENCHMARK

Table 2, at the end of this paper, specifies the spacecraft components that are included in the benchmark. A real spacecraft may carry significantly more redundant inactive units for fault tolerance. Table 3 specifies the traffic requirements presented by the benchmark. Each row corresponds to one type of traffic. High bandwidth sensor data (Payload) are destined for the downlink through the storage. Other packet types (unmarked in the table) are either telemetry data (TM), comprising measurements sent to the DHS for processing, or tele-command data (TC), consisting of commands from the DHS to other units. In some cases the TM and TC traffic are symmetric, and in other cases they are not. Each traffic flow comprises BW bandwidth, in addition to addressing, parity and control overhead. Some flows happen in bursts, according to specified cycle times (as further detailed in Section 3). End-to-end latency expectation, if any, is also specified. Flows that constitute control loops include messages from the DHS to sample data by various units and messages back to the DHS with such sampled data. Finally, flows are tagged by high, medium or low priority (H, M, L respectively).

3. THE SIMULATOR

An OPNET-based wormhole network-on-chip simulator [7][8][9] was adapted for SpaceWire network simulations. The simulated networks comprise end nodes, as specified in Section 2, and routers [10][11]. End nodes are characterized by two parameters: average number of N-Chars per packet and average time between packets. These parameters are either constant or exponentially distributed.

The average packet size for cyclic traffic with cycle time c and bandwidth requirement BW is set at $BW \times c$, generating one packet per cycle. Note that in the benchmark of Section 2 such packets are limited to 100 N-Chars long, and are typically high priority. Payload packets are low priority, 60,000 N-Chars each. Other packets are either 100 N-Chars long (when BW>100 Bytes/sec) or $|BW|$-long, sending one packet per second. Ten overhead N-Chars are added to payload packets, to provide for control data such as for RMAP or RDDP, and one overhead N-Char is added to all other packets, to accommodate the logical address.

The average time between packets of cyclic control-loop packets is the cycle time c. For other packets, the inter-arrival time is exponentially distributed with parameter packet-size/BW.

All routers have eight ports, and each port may or may not be equipped with several Virtual Channels (VCs); different ports may have a different number of VCs. Routers perform priority arbitration, and round-robin arbitration among same-priority packets, first on different input ports and then on different VCs within each port.

The routers support Group Adaptive Routing (GAR), which means that more than one link can be connected in parallel between two routers, and messages may be routed between the routers over any one of multiple alternative links. Note that GAR behaves differently from a single connection with a bandwidth equal to the total bandwidth of all parallel GAR connections. Consider, for example, a low-priority packet transmitted from router A to router B. If there is a single connection between A and B, a high-priority packet will wait for the low-priority

packet to complete. In contrast, if there are two connections of half-the-speed each between A and B, the transmission of the low-priority packet takes longer, but a high-priority packet need not wait and may be transmitted in parallel over the second connection.

The simulator enables the implementation of N-Char Interleaving (NCI, presently disallowed by SpaceWire) [8]. NCI may outperform GAR. For instance, in the last example, if there are two connections, then the high and low priority packets can be transmitted in parallel but at a low rate. In contrast, if there is only one connection that is twice faster, then the high-priority packet can take over the line, transmit faster and arrive sooner, while the low-priority packet is blocked. Thus, resource sharing is improved. On the other hand, N-Char interleaving requires adding overhead bits to each N-Char.

The simulation does not implement physical routing. Rather, the routers support only logical routing, using the path with the minimal number of hops (routers). When there are multiple alternative minimal paths, traffic is divided equally among them, enabling arrival out of order. Routing tables are created automatically upon starting the simulation, using a BFS algorithm.

Flow control tokens are implemented in the simulator according to the SpaceWire specification. Time Codes and Errors are not implemented. Link data rates are assigned automatically: the bandwidth needed for each link is computed and then multiplied by a predetermined factor. Other simulation parameters include:

- Number of priority levels;
- Number of VCs [12] for each unit and each priority level;
- Buffer size for each unit and each VC (all VCs of the same unit and same priority level have the same buffer size);
- Network topology;
- Packet-Level Priority (PLP) [5][13], disabled or enabled in the routers; end units always support it and issue new packets in priority order;
- NCI disabled or enabled; in the latter case, service-level bits are added to each N-Char.

Only a small subset of the many possible configurations has been simulated using the benchmark, as presented in the next section.

4. SIMULATION RESULTS

This paper does not present an optimal network configuration. Rather, it demonstrates the simulation study of several alternative configurations using the proposed benchmark. Five different configurations are considered, all sharing the same linear topology and same network resources, as shown in Figure 1. In all routers, a 128-byte buffer is allocated to each input port, and when more than one virtual channel (VC) is used, then the same buffer is divided among the VCs [12]. Table 1 summarizes the configurations.

Table 1: Configurations parameters

Config.	N-Char Interleaving (NCI)	Packet Level Priority (PLP)	Virtual Channels (VCs)
1	No	No	1
2	No	Yes	1
3	Yes	No	4
4	Yes	Yes	4
5	Yes	Yes	8

Without NCI, each N-Char is 10-bit long. For NCI, 2 and 3 overhead bits are added to each N-Char to identify the VC when using 4 and 8 VCs, respectively.

Most traffic comprises low priority payload packets from the sensor to the storage and subsequently to the downlink units. We scaled this traffic by several factors to simulate different loads and present the average End-To-End (ETE) delay for each priority level as a function of that load. ETE delay is measured as the time from packet creation until the arrival of the tail N-Char at the destination.

Different capacities are assigned to different links, but they are kept constant per each link over all simulations. Capacity is determined by considering all data and overhead traffic requirements over each link and then selecting the slowest link rate from the set of {2, 10, 50, 100, 400} Mbit/sec which still accommodates the required traffic. Each simulation measures 10 seconds of real-time operation.

Figure 2 through Figure 4 show the ETE delay of all packets for configurations 1-5. Low-priority payload traffic from the sensor to the storage unit traverses links having capacity of 400 Mbit/sec. The ETE packet delay charts of low-priority payload packets in configurations 1 and 2 (Figure 2) reveal a relatively low upper bound, which is explained as follows. The downlink units can receive data at 60 Mbit/sec and are therefore connected by links with capacity of 100 Mbit/sec. The storage unit can send packets only to one of the downlink units at a time. Therefore, due to wormhole routing, the maximum rate in which the storage can send traffic to the downlink units is 100 Mbit/sec, even though the link emanating from the storage unit is designed with 400 Mbit/sec capacity. Most traffic generated by the storage unit is destined to downlink units and therefore its effective working rate is only 100 Mbit/sec when it can actually generate 300 Mbit/sec. Thus, for payload traffic above 1/3 of the total benchmark requirement, the queue in the storage is overloaded and ETE delay of packets to the downlink units increases dramatically. The results are unaffected by PLP (configuration 2) because payload traffic is low priority.

Still in Figure 2, configuration 3 exploits priority-based NCI and succeeds in eliminating the upper bound. Four VCs are established among the storage and downlink units. The storage can send packets to the downlink units at the maximum rate of 400 Mbit/sec. However, since an overhead of 2 bits per N-Char is required to identify the service level of the N-Char, the effective rate is now 360 Mbit/sec (the previously noted 300 Mbit/sec with 2 bits added to each 10-bit N-Char). As shown in Figure 2, the full generated bandwidth is managed successfully by the network.

Configuration 4 also employs four VCs among the storage and downlink units. However, PLP is also enabled, so the same number of VCs must be distributed among the various priority levels. We allocate 1-1-2 VCs to the H-M-L levels of priority. The two VCs at the lowest level (with the highest traffic load) are insufficient. The result is similar to that of configurations 1 and 2, except that the storage can send traffic at 200 Mbit/sec. Thus, an upper bound of 200/360 Mbit/sec is encountered.

Configuration 5 attempts to mend the problem of configuration 4 by using eight VCs, assigned 2-2-4 to priorities H-M-L, respectively. However, due to the added overhead bit, the storage needs to send data at the rate of 390 Mbit/sec. Furthermore, the storage does not only send data to the downlink units, but also flow control tokens (FCT) to the sensor. In a steady state, for each eight arriving N-Chars, one FCT must be sent. Since the storage receives traffic at 390 Mbit/sec, it also sends about 26 Mbit/sec of FCTs. This requirement creates a total flow of 416 Mbit/sec out of the storage. However, since the maximum link rate is only 400 Mbit/sec, the storage is limited to serving only 0.96 of the required benchmark traffic. This limitation is shown as a 0.96 upper bound in Figure 2.

Regarding medium and high priority packets (Figure 3 and Figure 4, respectively), for configurations 1 and 2 we observe that the ETE delay increases about linearly with the load over the first 1/3 of the benchmark traffic. At this point the network is overloaded by the low-priority packets and therefore any increase in the load beyond this point will have little effect on high and medium priority packets.

When using NCI and PLP (configurations 4 and 5) the load of the low-priority packets has almost no effect on the ETE delay of high-priority traffic. However, when we use NCI but no PLP (configuration 3), the load of low-priority packets affects the ETE delay of high-priority packets since in this case there is no guarantee that a high-priority packet will find an available VC.

The average ETE delay of high-priority packets (Figure 4) is longer than that of medium-priority ones (Figure 3) because of the larger average number of hops that high-priority packets must traverse in the example network topology.

5. CONCLUSIONS

We have proposed a benchmark for studying SpaceWire networks. The benchmark specifies spacecraft units, as well as requirements for inter-unit communication traffic. To demonstrate the usefulness of the benchmark, we considered a network topology and five different network configurations. The network has been simulated using OPNET, and end-to-end delay characteristics are presented. The simulated configurations include packet-level priorities and N-Char interleaving.

The simulations show that both packet-level priorities and N-Char interleaving may improve network performance. It appears that allowing atomic transfer units larger than N-Characters may improve network efficiency.

6. REFERENCES

[1] spacewire.esa.int

[2] Parkes, S. M., “SpaceWire: Links, Nodes, Routers and Networks”, European Cooperation for Space Standardization, standard number ECSS-E50-12A, January 2003.

[3] Parkes S. M., McClements C., “SpaceWire Networks”, DASIA 2002,

[4] S Mills, S Parkes, TCP/IP Over SpaceWire, Proc. DASIA 2003.

[5] G Rakow, R Schnurr, S Parkes, SpaceWire protocol ID: what does it means to you? IEEE Aerospace Conference, 2006.

[6] S. Parkes *et al*., CCSDS Time-Critical Onboard Networking Service, SpaceOps 2006.

[7] OPNET Modeler, www.opnet.com

[8] E. Bolotin, I. Cidon, R. Ginosar and, A. Kolodny, "QNoC: QoS Architecture and Design Process for Network on Chip", Journal of Systems Architecture, Volume 50, February 2004

[9] P. Fourtier, Simulation Of A Spacewire Network, DASIA 2007, 2A

[10] S.M. Parkes, C. McClements, G. Kempf, S. Fischer and A. Leon, “SpaceWire Router,” *Int. SpaceWire Sem.,* ESTEC, 2003.

[11] W.J. Dally and C. Seitz, "The Torus Routing Chip", Distributed Computing, vol. 1, no. 3, 1986.

[12] W. Dally, "Virtual Channels Flow Control", Proc. ISCA, May 1990.

[13] SM Parkes, P Armbruster, SpaceWire: a spacecraft onboard network for real-time communications, Real Time Conf., 2005.

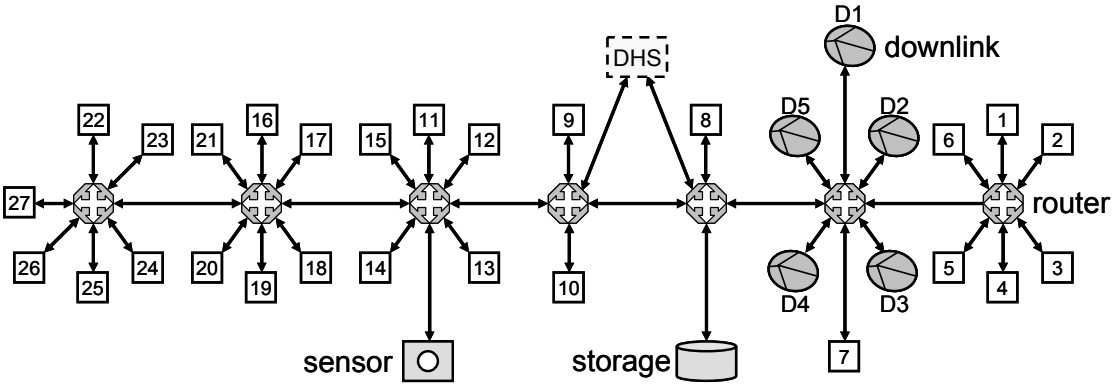

Figure 1: Example SpaceWire Network

Table 2: Benchmark Spacecraft Components Interconnected by SpaceWire

System	Name	Description	Num. Units	Num. Active Units
DHS	DHS	CPU	2	1
DHS	RU	Reconfiguration Unit	1	1
PLD	SENSOR	Camera, Particle Detector, etc.	3	1
PLD	STORAGE	Mass Memory for Storing Sensor Data	3	1
PLD	DOWNLINK	High-speed Communication Link	6	5
PLD	DAP	Downlink Antenna Positioning	2	1
PLD	DAS	Downlink Antenna Selection	2	1
TT&C	TTCGCS	Interface to Ground Control Station	2	1
TT&C	TTCAP	TT&C Antenna Positioning	2	1
TT&C	TTCAS	TT&C Antenna Selection	2	1
	THERMAL	Thermal Control	50	50
Elect	PCU	Electric Power Conditioning Unit	2	1
Elect	PDU	Electric Power Distribution Unit	2	1
Elect	BAPTA	Solar Array Bearing, Power Transfer Assembly	2	2
AOCS	STR	Star Tracker	3	3
AOCS	RW	Reaction Wheel	4	4
AOCS	MGM	Magnetic Flux Meter (Magnetometer)	2	2
AOCS	CSS	Coarse Sun Sensor	4	4
AOCS	FSS	Fine Sun Sensor	2	2
AOCS	GPS	GPS	2	1
AOCS	HPS	Hydrazine Propulsion System	2	1
AOCS	HETS	Hall Effect Thruster System	2	1

System acronyms:

DHS	Data Handling System
PLD	Payload
TT&C	Tracking Telemetry & Command
AOCS	Attitude and Orbit Control System

Table 3: Benchmark Traffic Specifications

From	To	Type	BW B/Sec	Cycle [Sec]	Latency	Control Loop	Priority
SENSOR	DHS		1,000	0.1			M
DHS	SENSOR		1,000		Low		H
STORAGE	DHS		1,000				M
DHS	STORAGE		1,000				M
SENSOR	STORAGE	Payload	100M				L
DHS	DOWNLINK		100				M
DOWNLINK	DHS		100				M
STORAGE	DOWNLINK	Payload	20M				L
DAP	DHS		500	0.1		Yes	H
DHS	DAP		200	0.1		Yes	H
DAS	DHS		1				M
DHS	DAS		1				M
DHS	TTCGCS		10K				M
TTCGCS	DHS		5K				M
TTCAP	DHS		500	0.1	Low	Yes	H
DHS	TTCAP		200	0.1		Yes	H
TTCAS	DHS		10				M
DHS	TTCAS		1				M
THERMAL	DHS		2.5	60			M
DHS	THERMAL		0.25	120			M
PCU	DHS		100				M
DHS	PCU		10				M
PDU	DHS		500				M
DHS	PDU		50				M
BAPTA	DHS		50	1		Yes	H
DHS	BAPTA		10	1		Yes	H
STR	DHS		2,000	0.1	Low	Yes	H
DHS	STR		1,000	0.1		Yes	H
RW	DHS		500	0.1	Low	Yes	H
DSH	RW		500	0.1	Low	Yes	H
MGM	DHS		50	1		Yes	H
DHS	MGM		1	1		Yes	H
CSS	DHS		20	1		Yes	H
DHS	CSS		1	1		Yes	H
FSS	DHS		20	1		Yes	H
DHS	FSS		1	1		Yes	H
GOS	DHS		100	1			M
DHS	GPS		17	30			M
HPS	DHS		100	0.1			M
DHS	HPS		100		Low		H
HETS	DHS		1000	0.1			M
DHS	HETS		100				M
RU	DHS		100	0.1	Low		H
DHS	RU		10	0.1			M

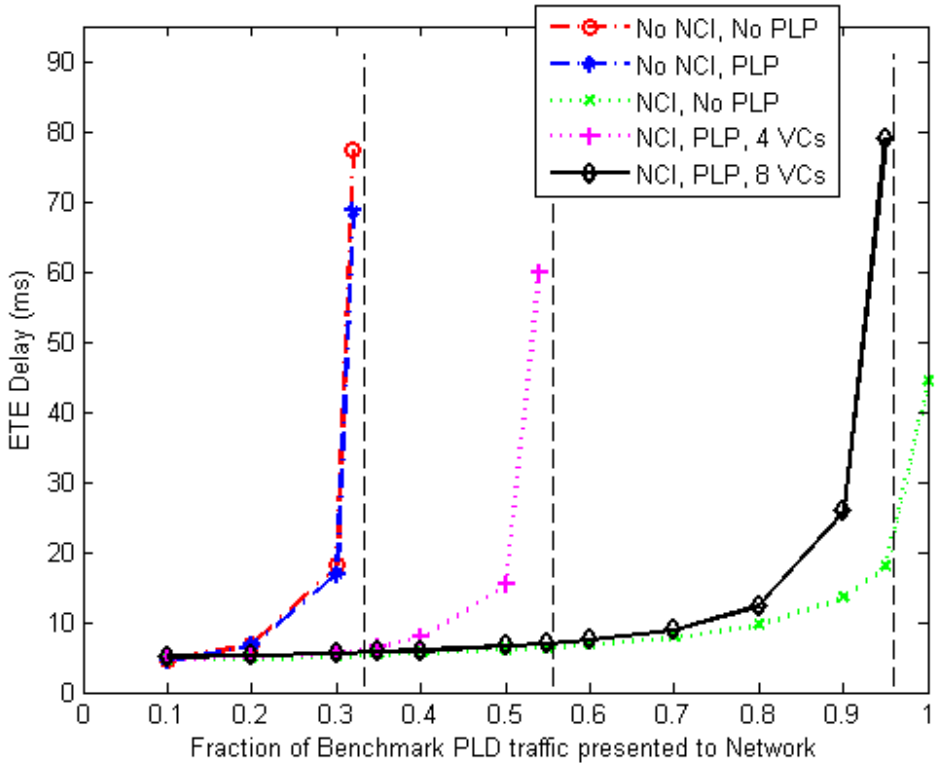

Figure 2: Average ETE delay of low-priority packets

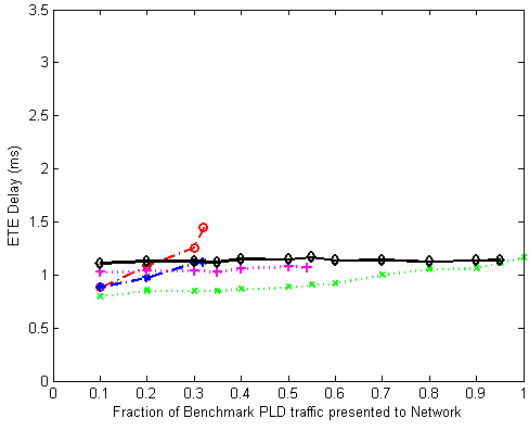

Figure 3: Average ETE delay of medium-priority packets

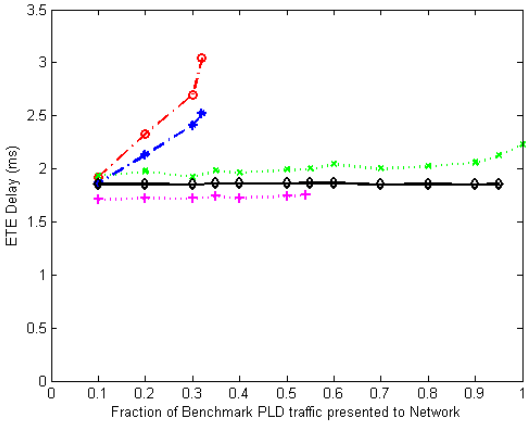

Figure 4: Average ETE delay of high-priority packets