Using Network Calculus to compute end-to-end delays in SpaceWire networks

Thomas Ferrandiz
Univ. de Toulouse, ISAE
Email: thomas.ferrandiz@isae.fr

Fabrice Frances
Univ. de Toulouse, ISAE
Email: fabrice.frances@isae.fr

Christian Fraboul
Univ. de Toulouse, IRIT/ENSEEIHT-INPT
Email: christian.fraboul@enseeiht.fr

***Abstract*—The SpaceWire network standard is promoted by the ESA and is scheduled to be used as the sole on-board network for future satellites. This network uses a wormhole routing mechanism that can lead to packet blocking in routers and consequently to variable end-to-end delays. As the network will be shared by real-time and non real-time traffic, network designers require a tool to check that temporal constraints are verified for all the critical messages.**

Network Calculus can be used for evaluating worst-case end-to-end delays. However, we first have to model SpaceWire components through the definition of service curves. In this paper, we propose a new Network Calculus element that we call the Wormhole Section. This element allows us to better model a wormhole network than the usual multiplexer and demultiplexer elements used in the context of usual Store-and-Forward networks.

I. Introduction

SpaceWire [1], [2] is an on-board network for satellites designed by the European Space Agency and the University of Dundee. It uses high-speed serial, point-to-point links and simple routers to interconnect satellite equipment using arbitrary topologies. In the future, the ESA plans to use SpaceWire as the sole on-board network in their satellites. The idea is to use the same network for both the payload and command/control traffic. This will simplify the network architecture and reduce the costs of the satellites.

At the moment, SpaceWire provides enough bandwidth (up to 200 Mbps) to carry both types of traffic at the same time. However, in order to reduce memory size (radiation-hardened memory is very expensive), SpaceWire uses wormhole routing to transmit the data packets across the network. The downside of this technique is that it can lead to blocked packets and thus huge variations in the end-to-end delays of those packets. Furthermore, SpaceWire does not provide built-in real-time mechanisms that guarantee the timely delivery of urgent packets.

Thus, network designers need a tool to check that temporal constraints are verified for urgent packets before SpaceWire can be used to carry command/control traffic. Using simulations is not possible because covering every scenario would be very long and costly. A better solution is to design an analytical model to compute an upper-bound on the worst-case end-to-end delay of a flow.

We proposed a first model in [3] that allowed us to compute such an upper-bound for a SpaceWire network. This model works well in most cases but has some limitations. As it did not require a specific model of input traffic, it was pessimistic when the network was not fully loaded.

As a consequence, we chose to create a new model based on Network Calculus theory [4]. It is a theory designed to study deterministic queueing systems which we have already successfully used in [5] to study the AFDX network. One strong point of Network Calculus is that it allows us to model the input traffic precisely by using traffic envelopes.

However, Network Calculus has been mostly used to study Internet components and not wormhole routers. As a consequence, the usual multiplexer and demultiplexer elements are not adequate to model a wormhole network. Thus, in Section II, we propose a new network element we call a "Wormhole Section". We can then divide the path of a flow into a series of Wormhole Sections to deduce an end-to-end service curve.

Furthermore, existing models are either not suited to a wormhole network or very pessimistic. In Section III, we give a new model to compute the delay caused by a flow when it leaves a Wormhole Section.

Finally, we conclude and discuss future work in Section IV.

II. The wormhole section element

A complete overview of Network Calculus would be beyond the scope of this paper. Readers not familiar with this theory can refer to [6] for a short introduction.

Here, we just recall this fundamental theorem.

Theorem 1: Concatenation of two systems
Assume that a flow traverses two systems S_1 and S_2 in sequence. Assume that S_1 and S_2 offer the service

curves β_1 and β_2 respectively. Then the concatenation of the two systems offers the service curve $\beta_1 \otimes \beta_2$ to the flow. $\beta_1 \otimes \beta_2(t) = \inf_{0 \leq s \leq t}\{\beta_1(t-s) + \beta_2(s)\}$ is the Min-Plus convolution of β_1 and β_2.

This allows us to combine the network elements successively traversed by a flow to obtain an end-to-end service curve offered to the flow by the network as a whole.

A. Assumptions

We consider a network composed of SpaceWire routers and terminals. Each terminal has only one SpaceWire interface but can be the source and/or destination of any number of flows. Each flow f is modeled by a source, a destination, a path through the network and an arrival curve α_f. Usually, the arrival curve is a staircase function which gives a more precise model of the input traffic than an affine function.

Since SpaceWire routers use static routing, we consider only a static network. We also assume that the network is stable, i.e. that no link is required to transfer more data than its capacity. In Network Calculus terms, this can be written as follows (see [7]). For each link j, we note I_j the number of flows traversing that link and α_i^j the arrival curve of flow f_i at link j. The stability condition is now:

$$\forall j, \forall i \in \{1, \ldots, I_j\}, \lim_{t \to +\infty} [\beta^j - \sum_{i=1}^{I_j} \alpha_i^j](t) = +\infty \quad (1)$$

B. A new network element

To use Network Calculus theory, we first need to determine a service curve for each element traversed by a flow. We can then compute an End-To-End (ETE) service curve using Theorem 1. However, in a wormhole network, the service offered by a router is not independent from the service offered by downstream routers. Because of the flow control, a router can output data at an average rate far inferior to its nominal capacity.

For this reason, we do not propose a classic multiplexer/demultiplexer model of each router. Instead, we adopt a macroscopic view of the network which tries to optimize the ETE service curve of each flow while accounting for the interdependency between routers.

When a flow encounters interferences with other flows, the impact of each conflict is twofold. First, there is a delay when the flow is multiplexed with the interfering flow. Then, when the interfering flow is demultiplexed, the flow control mechanism will propagate its own delays backward to the studied flow.

To properly model this, we propose a new network element that we call a "Wormhole Section". The basic idea is to divide the path followed by a flow into a serie of sections. Each section is composed of a set of successive output ports shared by the same flows. Thus, a Section corresponds to the arrival or the departure of an interfering flow from the path of the studied flow.

Each wormhole section offers a service curve that depends on the arrival curves of the interfering flows. Once we have computed the service curve of every section in the path of a flow, we can deduce the end-to-end service curve by using Theorem 1.

Of course, once an interfering flow has left the path of the studied flow, it will go through other wormhole sections before reaching its destination. The delays caused in those sections are those that will be carried over to the studied flow.

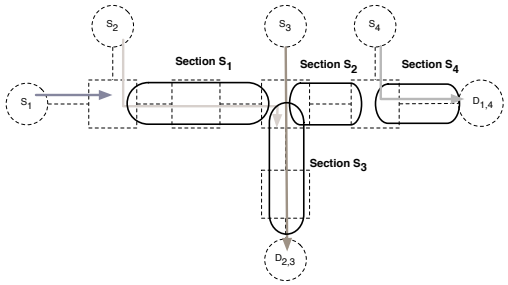

Figure 1. A network divided into wormhole sections

As an example, consider the network in Figure 1. Each flow f_i goes from its source S_i to its destination D_i. When a destination is shared by two flows i and j it is denoted $D_{i,j}$.

Let us take a closer look at f_1. f_1 has to go through three wormhole sections (S_1, S_2, S_4) to reach $D_{1,4}$ with sections S_1 and S_4 shared with other flows. S_1 and S_4 are thus the two sources of direct delays for f_1. In addition, since after leaving S_1, f_2 traverses section S_3 which it shares with f_3, S_3 will be another source of delay for f_1 but only indirectly.

As can be seen in this example, the wormhole section network element makes it easier to analyse the conflicts in a wormhole network. In the next section, we present a detailed model of this new network element.

C. Section sharing

We will present the model when there is only one interfering flow. We note f_1 the studied flow. f_2 is the interfering flow. f_i has $\alpha_i^{S_{in}}$ as an arrival curve at the entrance of section S. Let us assume that section S comprises M routers.

For $j = 1, \ldots, M$, β^j is the service curve offered by the output port of router R^j to the two flows. Because of

the flow control mechanism, the amount of data which is transmitted by the port may actually be inferior to β^j. Thus, β^j should be seen as an intermediate parameter used to determine the service guaranteed to a given flow by this section of the network. Once we have analysed the conflicts in each output port traversed by a flow, we can use Theorem 1 to compute a service curve for the section, then for the complete path. This end-to-end curve is now valid because it takes into account the influence of all the ports used by a flow, including the indirect delays. As a consequence, it represents the real end-to-end output of the network.

Since all the routers in the section are shared between the two flows and no other interference occurs, we can view these routers as a single router with service curve $\beta^S = \bigotimes_{j=1}^{M} \beta^j$.

SpaceWire routers use a simplified Round-Robin access scheme that guarantees that each input port gets a fair acces to each output port. However, each packet can use the output port for an unlimited duration. The usual round-robin model attributes some weight to each flow and shares the bandwidth in proportion to those weights. Since the SpaceWire standard does not define such weights, we have no way of using this model and have to rely on the more pessimistic "blind multiplexing" model [4], Theorem 6.2.1.

Thus, the service curve offered by the section to f_1 is

$$\beta_1^S = (\bigotimes_{j=1}^{M} \beta^j - \alpha_2^{S_{in}})_{\uparrow}. \tag{2}$$

$(\beta)_{\uparrow}$ is the *positive and non-decreasing upper closure* of β defined as $(\beta)_{\uparrow} = \max(\sup_{0 \leq s \leq t} \beta(s), 0)$.

β_1^S is a worst-case service curve that implies that all the interfering flows have a higher priority than f_1 and can instantly preempt it. Of course, in reality the packets are not interrupted but this model ensures that we have a worst-case service curve for any possible scheduling of the packets.

III. Section demultiplexing

A. Limits of existing models

In a classic Store-And-Forward network model, the packets are instantaneously demultiplexed. Furthermore, once a packet has left the router, the delays it can endure are not propagated backward on the network. Thus, the demultiplexer has no impact on the delay (see [7] for example).

However, this is not true for a wormhole network. In fact, after two flows f_1 and f_2 have been demultiplexed, f_2 can still have an impact on f_1. This is because the flow control mechanism will carry over the delays caused to f_2 on the end of its path backward to f_1.

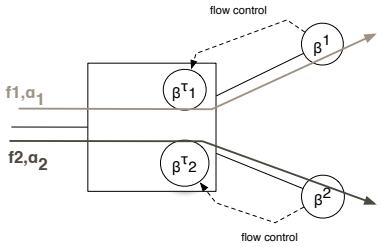

Figure 2. Demultiplexing of two flows in a wormhole network

A possible way to model this phenomenon is given in [8]. In this paper, the authors consider the situation described on Figure 2. In this network, after they have been demultiplexed both flows f_1 and f_2 have to go through a flow controller. Flow controller τ_i models the impact of the flow control on the downstream output link on flow f_i with a service curve β^{τ_i}. In turn, β^{τ_i} depends on the service curve of the downstream router.

The authors consider that, as far as the aggregate flow $f_{\{1,2\}}$ is concerned, τ_1 and τ_2 are parallel traffic regulators. As a consequence, the service offered to the aggregate flow $f_{\{1,2\}}$ is $\beta_{\{1,2\}} = \min(\beta^{\tau_1}, \beta^{\tau_2})$. This aggregate service is then shared between the two flows to derive the service offered to each individual flow.

This service is valid but is is very pessimistic. Let us consider the example in Figure 2. The arrival curves are affine: $\alpha_i(t) = r_i.t + b_i$ and the service curves are linear: $\beta_i(t) = C_i.t$ We use the following values:

	f_1	f_2
r_i (Mbps)	50	10
b_i (bits)	1000	200
C_i (Mbps)	100	20

On the one hand, the service offered to $f_{\{1,2\}}$ is $\beta_{\{1,2\}}(t) = \min(C_1, C_2).t = C_2.t$. On the other hand, the arrival curve of $f_{\{1,2\}}$ is $\alpha_{\{1,2\}}(t) = \alpha_1(t) + \alpha_2(t) = r_{1,2}.t + b_{1,2}$ with $r_{1,2} = r_1 + r_2 = 60$ Mbps and $b_{1,2} = b_1 + b_2 = 1200$ bits.

The horizontal distance $h(\alpha_{\{1,2\}}, \beta_{\{1,2\}})$ between the two curves is clearly infinite and we have to conclude that the network cannot carry both flows. This is very pessimistic because it is easy to see that this network can handle both flows.

Thus we need a new, more precise network model of wormhole demultiplexer.

B. A new model of demultiplexing

Since both flows go in different directions in the network, we cannot determine an exact service for the

aggregated flow. Each flow receives its own service but can still cause delays to the other flow thanks to the flow control mechanism.

The delay actually caused to an individual packet of f_1 is hard to determine because it depends on which exact conflicts occur farther on the path of f_2. However, we can determine an upper-bound on this delay.

In fact, the maximum delay caused by f_2, which we will denote d_2, is at most the maximum delivery delay from the demultiplexer to the destination of f_2. Thus,

$$d_2 = h(\alpha_2^{S_{out}}, \beta_2^{dest}) \tag{3}$$

where $\alpha_2^{S_{out}}$ is the arrival curve of f_2 at the end of S and β_2^{dest} the service curve offered to f_2 between the end of S and its destination.

Here, since we have assumed a blind multiplexing with f_2 as the higher priority flow,

$$\alpha_2^{S_{out}} = \alpha_2^{S_{in}} \oslash \bigotimes_{j=1}^{M} \beta^j \tag{4}$$

where $(\alpha \oslash \beta)(t) = \sup_{s \geq 0}\{\alpha(t+s) - \beta(s)\}$ is the Min-Plus deconvolution of α and β.

With this model, in the example in Figure 2 the delay caused by f_2 to f_1 is only $h(\alpha_2, \beta_2) = \frac{200}{10^7}s = 10\mu s$. As can be seen, this model is far less pessimistic than the model from [8].

C. Complete service curve offered by a Wormhole Section

We can combine the results from Subsection II-C and III-B to obtain the complete service curve offered by section S. When two flows share Section S, the service curve offered to f_1 is

$$\beta_1^S = (\bigotimes_{j=1}^{M} \beta^j - \alpha_2^{S_{in}})_{\uparrow} \otimes \delta_{h(\alpha_2^{S_{out}}, \beta_2^{dest})} \tag{5}$$

with the delay function $\delta_T(t) = 0$ if $t \leq T$ and $\delta_T(t) = +\infty$ if $t \geq T$ for any $T \geq 0$.

This formula can be easily generalized when there are more than one interfering flow.

D. Computing an end-to-end service curve

Once we can compute a service curve for each Wormhole Section, we have to combine them to determine an end-to-end service curve. When interfering flows arrive and depart in various routers along the path of a flow, we cannot directly use the model presented here.

We would have to use a Section for each router crossed by the flow which would give us a suboptimal result. Rather, we try to optimize the ETE service curve by combining a set of flows sharing several consecutive routers as an aggregate flow. We can then determine Wormhole Sections for this aggregate flow and deduce the service curves offered to it. Those service curves will then be shared between the individual flows composing the aggregate flow.

To automate this, we plan to use the three interference patterns defined in [8] to determine the order in which the residual services are computed.

IV. Conclusion

In this paper, we have defined a new Network Calculus element that can be used to model a SpaceWire network more accurately than the usual multiplexer and demultiplexer elements. We call this new element a Wormhole Section since it represents a part of the path followed by a flow. The Wormhole Section should allow us to obtain tighter bounds by considering successive routers as only one element.

Furthermore, we also described a new way to compute the delay caused by a flow leaving a wormhole section. We showed on a simple example that our method is a lot less pessimistic than existing demultiplexer models for a wormhole network.

We plan to implement a software tool based on the Wormhole Section element and the interference patterns defined in [8] to study some industrial configurations provided by Thales Alenia Space.

Acknowledgment

This work was supported by a PhD grant from the CNES and Thales Alenia Space.

References

[1] S. M. Parkes and P. Armbruster, "Spacewire: A spacecraft onboard network for real-time communications," *IEEE-NPSS Real Time Conference*, no. 14, Feb 2005.

[2] ECSS, "Spacewire – links, nodes, routers and networks," Aug 2008.

[3] T. Ferrandiz, F. Frances, and C. Fraboul, "A method of computation for worst-case delay analysis on spacewire networks," *SIES '09*, 2009.

[4] J. LeBoudec and P. Thiran, "Network calculus a theory of deterministic queuing systems for the internet," *Springer Verlag*, no. LNCS 2050, Apr 2004.

[5] F. Frances and C. Fraboul, "Using network calculus to optimize the afdx network," *ERTS 2006*, Jan 2006.

[6] J. L. Boudec and P. Thiran, "A short tutorial on network calculus. i. fundamental bounds in communication networks," *ISCAS 2000*.

[7] R. L. Cruz, "A calculus for network delay, part i: Network elements in isolation," *IEEE Transactions on Information Theory*, vol. 37, no. 1, Jan 1991.

[8] Y. Qian, Z. Lu, and W. Dou, "Analysis of worst-case delay bounds for best-effort communication in wormhole networks on chip," *Proceedings of the 2009 3rd ACM/IEEE International Symposium on Networks-on-Chip-Volume 00*.