Network Calculus:
Application to switched real-time networking

Jean-Philippe Georges, Thierry Divoux, Éric Rondeau
Research Centre for Automatic Control, Nancy-University, CNRS
Campus Sciences, B.P. 70239
Vandœuvre lès Nancy, France
jean-philippe.georges@cran.uhp-nancy.fr

ABSTRACT

In this paper, we show how Network Calculus can be used to determine whether a switched network may satisfy the time constraints of a real-time application. If switched architecture are interesting in the sense that they offer flexible design and may eliminate collisions in Ethernet-based network, they are not guaranteeing end-to-end performances (in particular in terms of delay), especially when cross-traffic are present. We illustrate Network Calculus usefulness by showing how the internal switching structure of an Ethernet switch simplify the analysis and which kind of traffic inter-dependencies are problematic.

Categories and Subject Descriptors

C.2 [**Computer Systems Organization**]: Computer Communication Networks—*Network Architecture and Design, Local and Wide-Area Networks*; C.4 [**Computer Systems Organization**]: Performance of Systems—*Modeling Techniques*

1. INTRODUCTION

Historically, industrial communications were mainly based on specific networks called fieldbuses such as Profibus or CAN. Those networks interconnect programmable controllers, CNC, robots, etc. to exchange technical data for monitoring, controlling, and synchronizing industrial processes. Their protocols ensure that the end-to-end delays of messages remain limited, compared with the time constraints of the applications. Thus, these networks are deterministic and it is possible for such networks to obtain directly time performances.

During the last decade, a trend to replace these dedicated networks by widely-used standardized solutions like Ethernet has been going on. The expected benefits are less costly network installations, because equipment is available off-the-shelf, and the avoidance of interoperability problems, because Ethernet technology is broadly used. Other advantages are that Ethernet is a well-known protocol, which is widely implemented, and its performance improves continuously with technological evolution (especially bandwidth). However, access to the Ethernet medium relies on the nondeterministic CSMA/CD algorithm, which applies a stochastic method for resolving collisions. In bus topology, Ethernet cannot guarantee that messages will be received in a bounded time. The idea is to ensure that the data flow is sufficiently quickly handled, compared with the time-cycle of the industrial applications. For [1, 8] the micro-segmentation with full-duplex switches combined with the appropriate real-time scheduling and fault-tolerance techniques also must enable the use of Ethernet in safety-critical applications with hard real-time constraints. Switched Ethernet networks are employed in a large scale of systems even in avionic aircrafts [4, 12] and spatial launchers [17]. Figure 1 shows a switched topology with full-duplex links and based on the micro-segmentation principle where collisions cannot appear but are shifted to potential congestion into switches.

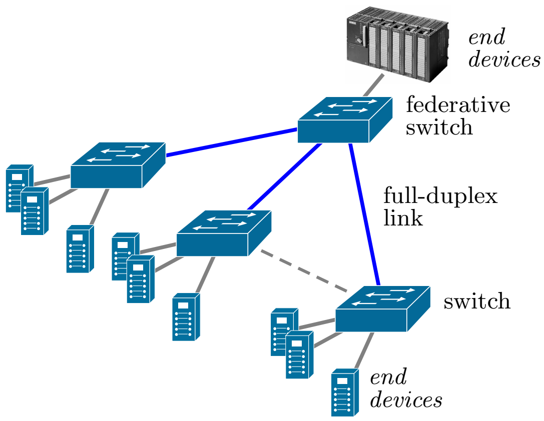


Figure 1: A hierarchical switched topology.

But all these considerations do not prove that all the packets are effectively received under a predefined bound. Figure 2 shows variations of end-to-end delays for a flow competing for access of an Ethernet switch with another flow. It highlights the interest for determining an upper-bound delivery time (defined in the IEC 61784 standard) which corresponds to the time required to send application data (message payload) from one node to another.

According to [14], Ethernet-based products can be summarily classified into three main categories: the native Ether-

VALUETOOLS 2011, May 16-20, Paris, France

DOI 10.4108/icst.valuetools.2011.245944

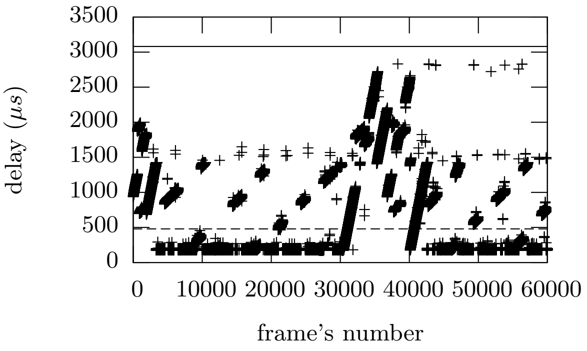


Figure 2: Measurement of end-to-end delays.

net standard (Ethernet/Industrial Protocol, Modbus/TCP), Ethernet solutions using the priorities defined in IEEE 802.1 D/Q, and Ethernet-based solutions that incorporate new scheduling features in ASIC/FPGA (EtherCAT, Profinet IRT). The last approach provides *a priori* known communication modes such that it simplifies transmission time estimation. For such protocols, when Ethernet is used as a dedicated network, [14] gives direct analysis of delays.

The time performance evaluation remains an issue when only native switched Ethernet architecture are considered. Rather than using specific electronics, it means that stations, sensors, actuators, etc. are not synchronized and might try accessing the medium in a pure CSMA/CD manner (non TDMA, non polling and non clock synchronization strategy). For example, the Ethernet/IP runs without introducing other concepts than those defined in IEEE 802.1D/Q standards. As a consequence, several flows might simultaneously sending frames on the network and create varying congestion situations as shown in Figure 2. Moreover, this issue will be more sensitive when the switched Ethernet networks is shared with other non time constrained applications. Next sections will detail how the Network Calculus has been applied for such networks and how the switching networking knowledge could be taken into account.

2. MOTIVATIONS

2.1 Network Calculus framework

The network calculus theory is based on the $(\min, +)$ algebra and models flows and services in a network with non-decreasing functions taking their values in the $(\min, +)$ semiring. Defined as a min-plus system theory for deterministic queuing systems, the network calculus has been used for DiffServ architectures on Internet, industrial Ethernet networks [9], avionics networks, wireless sensor networks [20] or networks on chip [21]. Main developments and results about network calculus can be found in [5, 6, 15]. In addition to the elementary calculus, several filtering operations are also translated in the $(\min, +)$ setting. The min-plus convolution respectively deconvolution of two functions f and g are hence defined to be:

$$(f \otimes g)(t) = \inf_{0 \le s \le t} \{f(t-s) + g(s)\}$$

$$(f \oslash g)(t) = \sup_{u \ge 0} \{f(t+u) - g(u)\}$$

Network modeling in network calculus is achieved by non-decreasing functions, characterising an amount of data at a time t. The set of such functions is given by:

$$\mathcal{F} = \left\{ f : \mathbb{R}^+ \to \mathbb{R}^+, \forall t \ge s : f(t) \ge f(s), f(0) = 0 \right\}$$

The first use of such function is for the input $R(t)$ and the output function $R^*(t)$, which cumulatively count the number of bits that are input to respectively output from a system $\mathcal{S}$. This corresponds to the real movements of data. Throughout the paper, we assume these functions to be continuous in time and space. This is not a major restriction as there are transformations from discrete to continuous models [15].

The second use is for the constraints that the real movements of data satisfy. Given a flow with input function R, a function $\alpha \in \mathcal{F}$ is an arrival curve for R if:

$$\forall t, s \ge 0, s \le t : R(t) - R(t-s) \le \alpha(s) \Leftrightarrow R \le R \otimes \alpha$$

The third use is for the service offered by a system to a flow defined by an input function R which results in an output function R^*. Then, the system is said to provide a minimum service curve $\beta \in \mathcal{F}$ if:

$$R^* \ge R \otimes \beta$$

In addition, β will be called a strict service curve for the system $\mathcal{S}$ if during any backlogged period of duration u, at least $\beta(u)$ data is served.

Let us turn now to the performance characteristics of flows that can be bounded by network calculus. Assume a flow with input function R that traverses a system $\mathcal{S}$ resulting in the output function R^*. The backlog of the flow at time t is defined as:

$$b(t) = R(t) - R^*(t)$$

Assuming first-in-first-out delivery, the delay for an input at time t is defined as:

$$d(t) = \inf\{\tau \ge 0 : R(t) \le R^*(t+\tau)\}$$

One may now consider the arrival and service curves definitions. It gives the bounds defined below.

THEOREM 1 ([5, 15]). *Consider a system $\mathcal{S}$ that offers a service curve β and that stores input data in a FIFO-ordered queue. Assume a flow R traversing the system that has an arrival curve α. Then we obtain the following performance bounds for the backlog b, delay d and output arrival curve α^* for R^*:*

$$b(t) \le \sup\{t \ge 0 \mid \alpha(t) - \beta(t)\} = (\alpha \oslash \beta)(0)$$
$$d(t) \le \inf\{d \ge 0 \mid \forall t \ge 0,\ \alpha(t) \le \beta(t+d)\}$$
$$\alpha^* \le \alpha \oslash \beta$$

Finally, the consideration of the service offered to a flow along a given path may be studied by considering the two following results.

LEMMA 1 (TANDEM SYSTEMS [5, 15]). *Consider a flow crossing two nodes in tandem with respective service curves*

β_1 and β_2. Then the concatenation of the two nodes offers a minimum service curve $\beta_1 \otimes \beta_2$.

LEMMA 2 (RESIDUAL SERVICE [5, 15]). *Consider a node offering a strict service curve β and two flows entering that server, with respective arrival curves α_1 and α_2. Then a service curve for flow 1 is $\beta_1 = (\beta - \alpha_2)^+$.*

Next sections show how this theory has been applied until now in the framework of the performance evaluation of Ethernet based real-time communications.

2.2 Network Calculus interests

Network calculus is an interesting tool for the design of industrial/embedded networking where applications require hard quality of service guarantees, in particular bounded end-to-end delays. This theory has been hence used for switched Ethernet networks, in particular when the network is shared by several applications. Initial Network Calculus developments [5, 6, 15] (which were more considered for resources reservation in Internet) were used in a short time for Ethernet networks. In [13], the Network Calculus theory is employed in order to bound delays by simply assuming that the service provided by switches is following rate-latency functions. It enables to compare efficiency of line vs star topologies. In other studies, for instance [11], a functional analysis of switching principles extends the switch modeling in a more detailed manner rather the simple indication of a $\beta(t) = R(t - T)^+$ switch service curve. Results have been next used in order to optimize the topology by using the end-to-end delays formula as the objective function [10]. Switched Ethernet topologies in a real-time context are not only appropriate to the factory area, but covers also embedded systems like the EADS A380 aircraft. In [12] the modeling of AFDX networks (switched Ethernet with admission control techniques) was achieved thankfully the Network Calculus theory. Extensions of these preliminary works were dedicated to tackle strict priority scheduling strategies in order to reduce maximum delays for time constraints applications.

These first results allowed to validate the method to determine time performances bounds in spite of best effort protocols. However, it was only limited to simple networks. In fact, first results mainly address the tandem systems shown in Figure 3(a) thankfully Lemma 1. The pay bursts only once principle enables here to tight the bounds rather than computing each time the output arrival curve (as defined in Theorem 1). For rate-latency services curves, [15] shows that the composition of rate-latency curves (the most employed) correspond to another rate-latency one. It gives also delays bound expression for affine arrival curve.

Next step consisted in considering the aggregation issue shown in Figure 3(b) for acyclic networks. If the service curve β is for the whole service offered by a system (i.e. a switch) and not dedicated to a given flow, it means that bounds could be only computed for the aggregated traffic $(\alpha_1 + \alpha_2)$. Due to inter-flow dependencies, multiplexing of a flow depends on other flow and *vice versa*. To obtain bounds for a given flow (for instance α_1), the scheduling policy of the

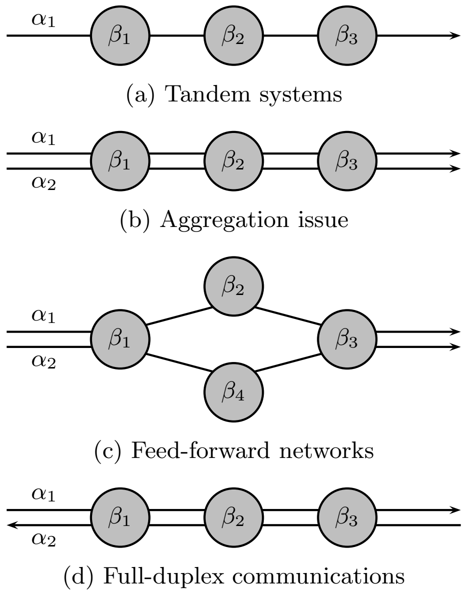
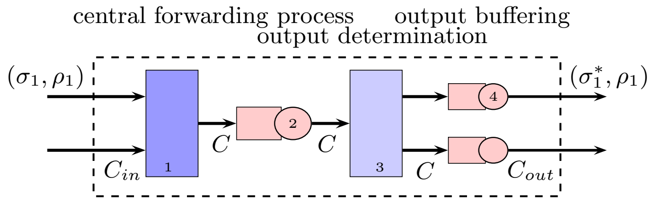


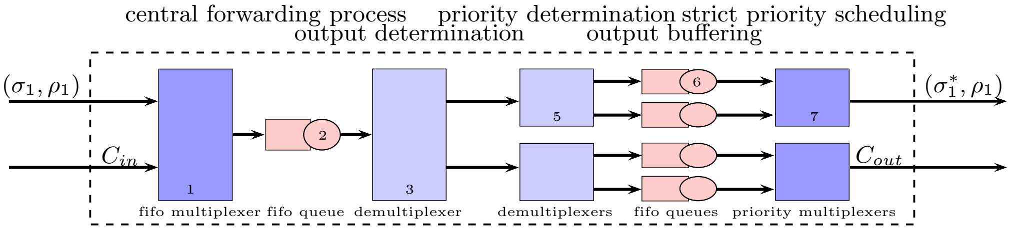
Figure 3: Flows mapping on switched topology.

system should be taken into account in order to derivate an appropriate service curve β' from the initial curve β. For this topic, Lemma 2 gives an interesting result known as blind multiplexing or residual service. It is equivalent to consider that in the worst case, a given flow will be served in an equivalent manner as a low priority traffic for a strict priority scheduler. [2, 16, 18] detail next how this lemma can be used in conjunction with the pay bursts only once principle in order to catch the pay multiplexing only once principle. For affine function and rate-latency functions, it is shown that the path service curve offered to the traffic $\alpha 1$ is given by $(\beta_1 \otimes \beta_2 \otimes \beta_3 - \alpha_2)^+$. A complete study is given in [2] about the conditions required to use this result and also how to compute more tight bounds. Associated network topology are related to the feed-forward property (see Figure 3(c)) defined by [19] as a network in which nodes can be labeled in such a way that the path of every flow is composed of an increasing sequence of node labels. Different methods have been proposed as the least upper delay bound in [16]. Here the end-to-end service curve in a tandem is computed by iteratively removing interfering flows. Recent developments about this topic can be found in [3]. It gives an algorithm which computes the maximum end-to-end delay for a given flow for any feed-forward networks under blind multiplexing with concave arrival curves and convex service curves.

These new theoretical developments have been taken into consideration in the context of performance evaluation based on Network Calculus for switched Ethernet architectures. For instance, [4] identify in the special case of AFDX networks, advanced knowledges about the aggregation (call grouping) of several flows. It enables to obtain more precise arrival curve functions and hence more tight delays. This paper shows the relation between theoretical results and applied results in local networking where additional *a priori* knowledge information might be extracted.



(a) Basic model.



(b) Prioritised switching model.

Figure 4: The prioritisation mechanism modifies the output buffering at output ports.

However communications in a realistic network might be more complex than the acyclic network illustrated in Figures 3(b). Figure 3(d) show more complicated studies. This one is typical in an industrial context where a calculator might send a control update to an actuator/sensor while the sensor is sending the feedback. In such case, previous results can not be applied such that it might be not possible to identify the service guaranteed by a path for a given flow. As a consequence, it is not possible to use the pay bursts only once principle for instance, and only local upper-bounds delays (i.e. the delay for crossing one switch) will be computed according output arrival functions given in Theorem 1. It means here that studies deal with total flow analysis.

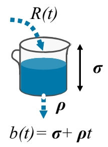
Since computed upper-bounded end-to-end delays are in an industrial context the key point to decide if a switched Ethernet topology might be employed or should be optimized, or even to decide to degrade the global applications performances, the tightness of these bounds is an important topic. Furthermore, it may be noticed here that switched Ethernet networks corresponds to a particular case study: paths are *a prori* known (defined by spanning tree algorithms), routing issue like the one illustrated in Figure 3(c) are not possible due to the routing along the spanning tree and switches have multiple input/output ports which means that two level of aggregation should be discussed: between flows entering by different ports and between flows entering by the same input port. Moreover, flows entering by the same input port are not necessarily exiting a switch by the same output port.

In following sections, this paper introduces which kind of additional information could be integrated in the application of the Network Calculus theory in switched Ethernet networks. It shows also where blind multiplexing issues are remaining.

3. SERVICE OFFERED BY SWITCHES

3.1 Arrival curve

Since the objective is to guarantee some deterministic performances of the network, the incoming traffic has to be bounded. For this, the leaky bucket controller concept [15] is mainly used to model the arrival constraints representing each traffic [11, 12, 13].



$\alpha(t) = \sigma + \rho t$

$\forall x, y;\ y \geq x,\ x \geq 0$, then

$R \sim \alpha \Leftrightarrow \int_x^y R(t)\,dt \leq \sigma + \rho(y - x)$

The burstiness constraint [6] imposes the traffic generation to be bounded by an affine function $\alpha(t)$, in which a variable burst value σ is associated to a constant rate ρ. $R(t)$ represents here the instantaneous rate of the stream, C the capacity of the links on the network, σ is the maximum amount of traffic that can arrive in a burst, i.e. the maximum length of the frames and ρ is an upper bound on the long-term average rate of the traffic flow : the data amount sent each time cycle. To take into account the capacity of the links, the previous affine function is completed with a stability constraint $\alpha(x) \leq Cx$. It means that the arrival of data cannot be greater than the capacity C of the link. So, we have:

$$\alpha(t) = \min\{Ct, \sigma + \rho t\} \tag{1}$$

In industrial communications in which controls are often time-triggered, it might be noticed here that affine function are interesting for modeling periodic arrivals.

3.2 Service curves

In switched networks, a key topic is relative to the identification of a service curve for each switch. The challenge is here to determine the service curve parameters and to not directly assume that a rate-latency curve is enough. Whereas the use of rate-latency curve consists in a black box strategy [13], other models that have been proposed try to isolate the elementary functions of the switching principle as proposed in [4, 7, 11].

Figure 4(a) is the result of previous studies (see [9]) in which several models are built and compared according to the IEEE 802.1D standard. It is constituted of a sequence of three basic elements: one multiplexer, one queue and one demultiplexer. In order to take into account the internal speed of a switch, the three following capacities should be used: C, the throughput inside the switch, C_{in}, the throughput of arrival of data on input ports and C_{out}, the output capacity. Since the study is about Ethernet networks, the scheduling policy has to be non preemptive. Moreover, it is considered that the components are work-conserving systems, i.e. they cannot have vacations and their forwarding policy is the best-effort. In this paper, the FIFO forwarding policy is chosen in order to respect the fairness of Ethernet: a FIFO memory to model the switching process and FIFO queues to model the transmission function of the output ports.

This model has been improved in [9]. The model is extended with the priority management. The implementation of priorities inside a switch mainly consists in adding for each switch output port as many buffers as priorities. These buffers are modeled by FIFO queues (component 6 on the figure 4(b)). The classification operation of frames belonging to a same switch output port is achieved by a demultiplexer (component 5). It enables to select the relevant queue from the frames priority. Finally the goal of the last multiplexers (component 7; one per switch output port) is to apply a scheduling policy between their different queues. The output capacity of the component 7 corresponds to the network bandwidth. Figure 4(b) represents a two-ports switch model which is able to manage the frames forwarding with two levels of priorities. In order to take into account the internal speed of a switch, the three capacities C_{in}, C and C_{out} are used.

Based on this model, the minimal service of a switch depends hence on the service curve defined for each elementary components. While elements like demultiplexer correspond to electronic latencies (and hence burst delay curves) and may be neglected, others like multiplexers are related to the scheduling strategies. The service curve offered by a switch might be directly obtained by considering the switch as a tandem systems and hence Lemma 1. For Figure 4(a), it gives then $\beta = \beta_1 \otimes \beta_2 \otimes \beta_3 \otimes \beta_4$.

Consider firstly the switch model defined in Figure 4. The first element is a FIFO multiplexer. The service by this multiplexer is related to the internal switch capacity C and to the switching mode latency (store & forward, fragment-free, etc.), such that a rate-latency service curves might be defined. For instance, for store & forward, it gives $\beta(t) = C\left(Rt - L/C_{in}\right)^{+}$. It means here that results like blind multiplexing may be applied in order to determine a separate flow analysis. It is important to note here that two aggregation levels have to be considered: between input ports and between flows sharing the same input port. In switched networks with full-segmentation, it means also that the task scheduling in end devices (local multiplexing) should be also considered. Nevertheless, it is important to note here that the switching fabric capacity is larger than the sum of the input port capacities $C \gg C_{in}$ (backplane property). It means that only the switching mode has a real impact and the service consists mainly in a burst delay. The second element, represents electronic latencies, and stands too for a burst delay. Demultiplexers identify the associated output port and/or the associated queueing policy. It corresponds to a burst delay service (but it is generally neglected).

Finally, it is important to note here how the output buffering in switches is managed. Modeling should be able to catch the capacity for a switch to simultaneously forward frames on different output ports. The output buffering strategy (which is selected to avoid the head of line blocking) reflects this point. (Output) multiplexing in switch is mainly related to flows trying to access the same output port. Without classification of service mechanisms (i.e. FIFO multiplexing), blind multiplexing strategies have to deal not with the whole traffic entering in the switch, but only with the amount of traffic exiting the switch by the same port (which limits the aggregation issue). For FIFO multiplexers, the service is hence defined from the output port capacity. With classification of service, it corresponds to a locally FIFO scheme. Globally, the service offered will be relative to a strict priority or a fair queue scheduler and flows belonging to the same class (or priority) and forwarded to the same output port will be served according to a FIFO strategy. The main interest of strict priority or fair queueing policies is to not depend on cross-traffic parameters, but only on static priorities and weights. It enables to compute specific curves for *super* flows gathering all traffic with the same class at the same output port as shown in the following section. Strategies like blind multiplexing will hence only be used for the FIFO scheme.

4. OUTPUT SCHEDULING

The evolution of Ethernet to segmented architectures and the definition of the *Virtual Local Area Networks* (VLAN) have led to the birth of a new standards set (802.1D/p, 802.1Q) in which new encapsulation fields are added to the native frame format. One of these fields defines a priority level (8 levels are supported). These levels are related to 8 types of applications (voice, video, network management, best effort, etc.). The number of classes of service may be different to the number of priority levels, and also different for each port. Two scheduling policies are supported: the strict priority (SP) and the weighted round robin (WRR).

In the following, it is assumed that the size of a frame is upper-bounded by L. The notion of flow corresponds here to the set of traffic sharing the same priority and exiting the switch by the same output port (i.e. the traffic entering by a common input port of an output multiplexer on Figure 4(b). We assume also that the capacity of the inputs and outputs are relatively fixed at C_{in} and C b/s. Each flow $i \in \mathbb{N}^*$ is (σ_i, ρ_i) upper-constrained with $\sum_{j=1}^{n} \rho_j < C$ where $n \in \mathbb{N}^*$ is the number of priority/class and a weight ϕ_i is given to each flow. Now, we analyze in particular the delays for each flow according to whether the policy of the node is SP or WRR.

4.1 Strict priority

In the strict priority policy, none guarantee is offered to one flow. The selection order will simply depend on the priority (weight) order. Also, we have to distinguish the service curve offered to each flow. The strict priority policy guarantees to the packets of the flow with the highest priority (here, the flow 1) to be selected first. But, since the forwarding of a packet on the network cannot be preempted, packets of the flow 1 might have to wait the complete forwarding of a packet with a lower priority. Also, the service curve associated to the flow with the highest priority is given by :

$$\beta_1(t) = R(t-T)^+ \qquad T = L/C,\ R = C \tag{2}$$

Obviously, packets of the flow 2 are served before packets of the flow 3, but they will have to wait that none packets of the flow 1 is waiting before to be served. So there is a first latency corresponding to the processing time of the initial bursty period of the flow 1. Moreover, like above, it will not be possible to preempt the forwarding of a packet of the flow 3 in order to serve a packet of the flow 2 that has just arrived. Finally, since the node will serve first packets of the flow 1, the forwarding rate will be limited to $C - \rho_1$. A packet of the flow 3 will be served only if none packet of the flows 1 and 2 are waiting. That is to say that the service has a first latency period defined by the processing time of the initial bursty period of the union of the two others flows. Moreover, the service rate will be limited in this case to $C - \rho_1 - \rho_2$. The service curve of a flow $i > 1$ will be hence defined by for a flow :

$$\begin{aligned} \beta_i(t) =& R(t-T)^+ \\ T =& \frac{\sum_{j=1}^{j=i-1} \sigma_j}{C - \sum_{j=1}^{j=i-1} \rho_j} + L/C,\ R = C - \sum_{j=1}^{j=i-1} \rho_j \end{aligned} \tag{3}$$

Equations show that with the strict priority, the service offered to one flow depends on the other flows and the service offered to the flow with the lowest priority may tend towards zero.

4.2 Fair queueing and weighted round robin

The weight associated to a given port i is defined by ϕ_i (with $\phi_i \in \mathbb{N}^*$). In other words, it means that no more ϕ_i flits might be successively served per round robin cycle for a port i. The maximum number of frames served during a round robin cycle is defined by $\Phi_n = \sum_{j=1}^{n} \phi_j$.

The Weighted Fair Queueing is also known as the Packetized Generalized Processor Sharing (PGPS). It is based on the conceptual algorithm called the Generalized Processor Sharing (GPS). A GPS server is characterized by n positive real numbers $\phi_1, \phi_2 \ldots \phi_n$. It operates at a fixed rate C and is work conserving. Each flow i has a guaranteed service rate c such as:

$$c = \frac{\phi_i}{\Phi_n} C$$

The GPS policy is interesting since it uses fairness, it is flexible (the number ϕ_i enables to modify the service offered to a given flow, and consequently to other flows) and finally, it is analyzable and bounded. For example, for the case where $n = 2$, supposing that R_2 is (σ_2, ρ_2) upper-constrained, [5] shows that the service curve (number of bits served at time t) for the first flow β_1 can be defined by :

$$\beta_1(t) = \max\left\{(C - \rho_1)t - \sigma, \frac{\phi_1}{\phi_1 + \phi_2} Ct\right\} \tag{4}$$

At all, in opposition to the strict priority, the service offered to one flow only depends on the weights of the flows and on its properties: it respects the fairness queueing.

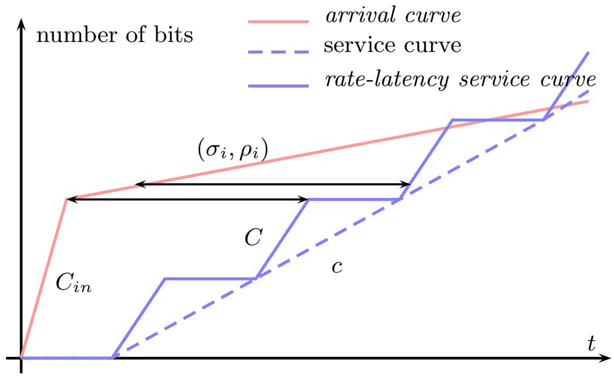


Figure 5: Service curve *weighted round robin.*

The definition of the minimal service curve offered to the flow will use the properties given in (4), but has also to consider that the worst case. When a frame just misses its slot, it will have to wait its next slot (at the next round). The frame will have to wait up to $\frac{\sum_{j \neq i} \phi_j}{C}$. A rate-latency service curve can be hence formulated as in (5) and shown in Figure 5.

$$\beta_i(t) = R(t-T)^+ \qquad R = C\frac{\phi_i}{\Phi_n},\ T = \frac{\sum_{j \neq i} \phi_j}{C} \tag{5}$$

It means that we retrieve here a rate-latency function as for SP. An upper-bound end-to-end delay can be hence computed as given in Theorem 1. We note $\tau_i = \frac{\sigma_i}{C_{in} - \rho_i}$.

$$\begin{aligned} d_i(t) \leq& \max_{t \geq 0}\left\{\inf_{\Delta \geq 0}\left\{\min\{C_{in}t, \sigma_i + \rho_i t\} = R(t + \Delta - T)^+\right\}\right\} \\ \leq& T \vee \max_{0<t<\tau_i}\left\{\inf\left\{\Delta \geq 0 : C_{in}t = R(t+\Delta-T)^+\right\}\right\} \\ &\vee \max_{t \geq \tau_i}\left\{\inf\left\{\Delta \geq 0 : \sigma_i + \rho_i t = R(t+\Delta-T)^+\right\}\right\} \end{aligned}$$

Since for all $t > 0$, $\min(C_{in}t, \sigma_i + \rho_i t) > 0$, $R \leq C_{in}$ and the stability conditions implies $\sum_j \rho_j \leq R$

$$\begin{aligned} d_i(t) \leq& T \vee \frac{(C_{in} - R)\tau_i + RT}{R} \vee \frac{(\rho_i - R)\tau_i + \sigma_i + RT}{R} \\ \leq& (T - \tau_i) + \frac{\sigma_i + \rho_i \tau_i}{R} \end{aligned} \tag{6}$$

Using the equation (6), it is now possible to compare delays provided by strict priority and weighted round robin. Considering $n = 3$. First, consider the flow 1 (with the highest priority). It is interesting to note here that if $\phi_2 \to 0$ and $\phi_3 \to 0$, the weighted round robin will be very closed to the strict policy. But if we consider now the flow 3 (with the lowest priority), if $\rho_1 + \rho_2 \to C$, the forwarding rate offered to the packet with a lower priority will tend to zero in the strict priority scheduling. This problem does not appear in the weighted round robin policy, since the forwarding rate will only depends on the weights values.

Figure 5 hightlights that rate-latency service curves introduce conservativeness in round robin principle since the forwarding rate of a packet does not correspond to the output port capacity. A local improvement of the delays bounds deals with the definition of the second service curve shown in dashed in Figure 5. The main idea is hence to take into account that it is the access to the switching fabric which is shared between the input ports, not directly the forwarding capacity. In fact, this service can be seen as the composition of the service offered by each cycle. Indeed, for a cycle n, the curve corresponds to a constant plus a rate latency $nK + \beta_{R,nT}(t)$ as shown by Figure 6.

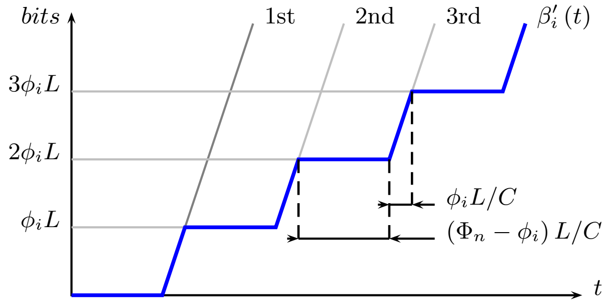


Figure 6: Composition of service for each round-robin cycle.

We propose then to replace (5) by:

$$\beta_i'(t) = (1\text{st}) \wedge (2\text{nd}) \wedge (3\text{rd}) \wedge \ldots = \inf_{k \in \mathbb{N}} \{kK + \beta_{R,kT}(t)\}$$
$$= \inf_{k\in\mathbb{N}} \left\{ k\phi_i L + C\,(t - \Phi_n\,(k+1))^+ \right\} \quad (7)$$

Different properties might be given for equation (7). To simplify, flows are constrained by an affine arrival curve $\alpha_i(t) = \sigma_i + \rho_i t$. Firstly, assume here that the service curve β_i' is a strict service curve offered by a server to the aggregate of two flows incoming by the same input port. Then if we have:

$$\beta_1(t) := \left(\inf_{k\in\mathbb{N}} \left\{ K + R\,(t-T)^+ \right\} - \alpha_2(t) \right)^+$$

Since the distributivity of $+$ with respect to $\wedge$, it gives:

$$\beta_1(t) = \left(\inf_{k\in\mathbb{N}} \left\{ K + R\,(t-T)^+ - \alpha_2(t) \right\} \right)^+$$
$$= \inf_{k\in\mathbb{N}} \left(K + R\,(t-T)^+ - \alpha_2(t) \right)^+$$

Then it is well known that the residual service between a rate-latency service curve and an affine arrival curve leads to a new rate latency service curve [15, 19]. The result is similar when a rate latency plus a constant service curve $(K + R\,(t-T)^+)$ is considered, such that :

$$\beta_1(t) = \inf_{k\in\mathbb{N}} \left\{ K' + R'\left(t - T'\right)^+ \right\}$$

with $R' = R - \rho_2$ (it is assumed that $R > \rho_2$), $T' = \frac{\sigma_2 + T\rho_2}{R - \rho_2}$ and $K' = (K - \sigma_2)^+$. It can be finally noticed that the curve β_1 is wide-sense increasing.

Definition 1. Consider a switch serving two flows, 1 and 2 entering into the switch by the same input port i with some unknown arbitration between the two flows. Assume that the switch guarantees a strict service curve β_i to the aggregate of the two flows. Assume that flow 2 is α_2-smooth and that $R > \rho_2$. The service curve offered to the flow 1 is defined by:

$$\beta_1(t) = \inf_{k\in\mathbb{N}} \left\{ K' + R'\left(t - T'\right)^+ \right\}$$

with $R' = R - \rho_2$, $T' = \frac{\sigma_2 + T\rho_2}{R - \rho_2}$, $K' = (K - \sigma_2)^+$ and K, R and T defined as in equation (7).

Secondly, according to the Lemma 1, we have:

$$\beta_1 \otimes \beta_2(t) = \inf_{0 \le s \le t} \{\beta_1(t-s) + \beta_2(s)\}$$
$$= \inf_{k_1,k_2\in\mathbb{N}} \left\{ \left(K_1 + R_1\,(t - T_1)^+\right) \otimes \left(K_2 + R_2\,(t-T_2)^+\right) \right\}$$

By considering K_1 and K_2 as two constant regarding t, it gives:

$$= \inf_{k_1,k_2\in\mathbb{N}} \left\{ K_1 + K_2 + R_1\,(t-T_1)^+ \otimes R_2\,(t-T_2)^+ \right\}$$

and by integrating the rate-latency convolution results, it finally gives:

$$= \inf_{k_1,k_2\in\mathbb{N}} \left\{ K_1 + K_2 + \min(R_1, R_2)\,(t - T_1 - T_2)^+ \right\}$$

Definition 2. Assume a flow traverses switches 1 and 2 in sequence. Assume that each switch offers a service curve of β_i, $i = 1, 2$ to the flow. Then the concatenation of the two switches offers a service curve:

$$\beta_1 \otimes \beta_2(t) = \inf_{k_1,k_2\in\mathbb{N}} \{K + R\,(t-T)\}^+$$

with $K = K_1 + K_2$, $R = \min(R_1, R_2)$ and $T = T_1 + T_2$.

4.3 Conclusion

In this section, service curves have been presented for the output multiplexing into the switch model in Figure 4(b). When no classification of service mechanism is used, the service offered by the output buffer is directly linked to a FIFO multiplexer with a rate related to the output port capacity. To obtain service curve for a particular flow, blind multiplexing should be considered.

5. END-TO-END UPPER-BOUNDS COMPUTATION

The objective of this section is to emphasize how Network Calculus results are then applied to compute end-to-end delays in switched Ethernet architectures. Figure 7 introduced the inter-depencies of the flow on such topology for models given in Figure 4.

In switched network, an important topic is to not loose the ability for a switch to forward flows with input/output ports, simultaneously in parallel. This impossibility might occurs if the switch is simply modeled by a global service curve β on which blind multiplexing is applied. The kind of full-duplex communications shown in Figure 3(d) is typical for such situation. Considering Figure 7, it may be noticed that even if flows entering in a switch are multiplexed in a common shared memory, it is done with a rate larger than

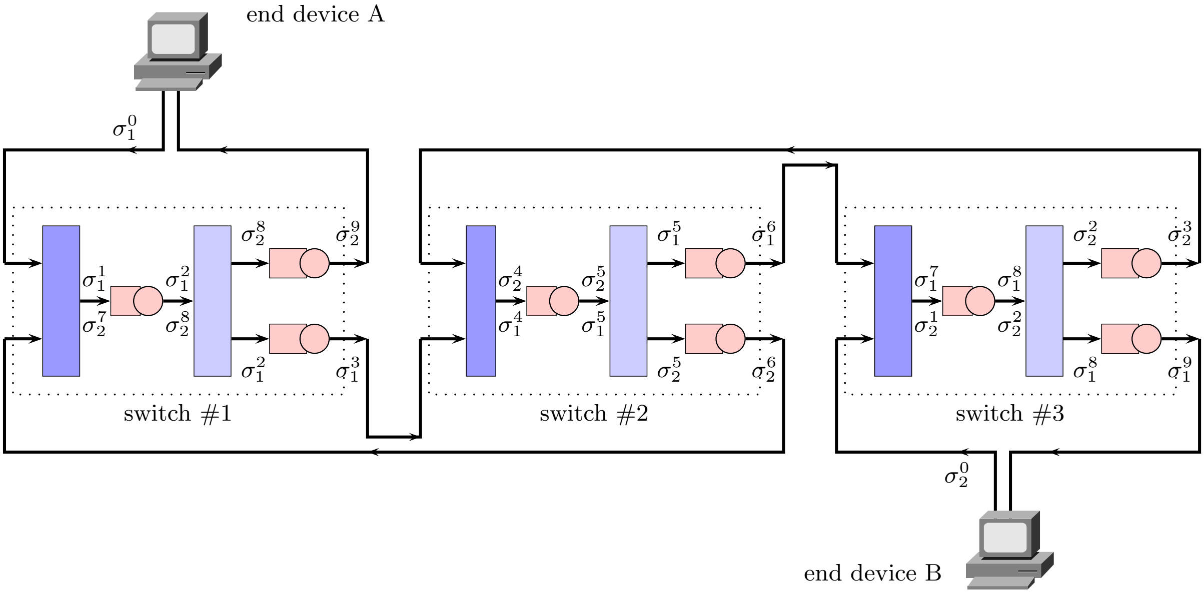


Figure 7: Synoptic to a full duplex communication between two node interconnected by a three switches linear topology.

the total input capacities. Hence, the additional delay added by this FIFO multiplexing remains limited compared to the output forwarding.

In such network, aggregation issues are limited between flows exiting a switch by the same output port. In addition, this issue is restricted when classification of service mechanisms are used for flows exiting a switch by the same output port and with the same class/priority level. Here current developments in the Network Calculus theory framework present a real interest.

The service offered by a switch to a given flow will hence be formulated for each output port. Considering Figure 7, if we neglect latencies and constant delays, it means that each switch may be represented by two distinct service curves related to the output forwarding. Hence, no aggregation issues are required as shown in Figure 8.

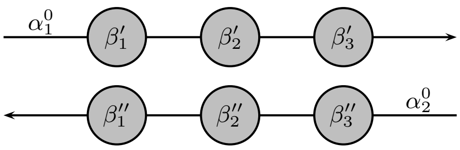


Figure 8: Flows mapping equivalent to synoptic of Figure 7.

6. CONCLUSION

This paper highlights how the Network Calculus has been applied for switched real-time communications when applications require time performance guarantees even if the network does not provide explicitly such guarantees. Computed end-to-end delays bounds are interesting to decide if the network will fulfilled the applications needs, to optimize the network topology (on which port on which switch an end devices should be interconnected) such that delays are minimized, to optimize classification of service parameters. This paper shows also how the last developments of this theory are improving the tightness of the end-to-end delay bounds. In the same time, it highlights that some issue are not applying to this kind of networks.

This paper talks about switched network, mainly switched Ethernet networks. One may considers other switched topologies like networks on chip. It is important to note here that aggregation issue will be different when frames queueing will be achieved at the input (and not at output as in this work).

7. REFERENCES

[1] M. Alves, E. Tovar, and F. Vasques. Ethernet goes real-time: a survey on research and technological developments. Technical Report HURRAY-TR-2K01, Groupe de recherche IPP-HURRAY, Polytechnic Institute of Porto (ISEP-IPP), Jan. 2000.

[2] A. Bouillard, B. Gaujal, S. Lagrange, and E. Thierry. Optimal routing for end-to-end guarantees using network calculus. *Performance Evaluation*, 65(11-12):883–906, Nov. 2008.

[3] A. Bouillard, L. Jouhet, and É. Thierry. Tight performance bounds in the worst-case analysis of feed-forward networks. In *Proceedings of the 29th conference on Information communications, INFOCOM'10*, pages 1–9, Mar. 2010.

[4] M. Boyer and C. Fraboul. Tightening end to end delay upper bound for afdx network calculus with rate latency fifo servers using network calculus. In *IEEE International Workshop on Factory Communication Systems (WFCS'08)*, pages 11–20, May 2008.

[5] C.-S. Chang. *Performance Guarantees in*

Communication Networks. Telecommunication Networks and Computer Systems. Springer Verlag, 2000.

[6] R. Cruz. A calculus for network delay, part I: Network elements in isolation. *IEEE Transactions on Information Theory*, 37(1):114–141, Jan. 1991.

[7] R. Cruz. A calculus for network delay, part II : Network analysis. *IEEE Transactions on Information Theory*, 37(1):132–141, Jan. 1991.

[8] J.-D. Decotignie. Ethernet-based real-time and industrial communications. *Proceedings of the IEEE*, 93(6):1102–1117, June 2005.

[9] J.-P. Georges, T. Divoux, and E. Rondeau. Strict priority versus weighted fair queueing in switched ethernet networks for time-critical applications. In *Workshop on Parallel and Distributed Real-Time Systems (WPDRTS'05 - IPDPS'05)*, pages 141–149, Apr. 2005.

[10] J.-P. Georges, N. Krommenacker, T. Divoux, and E. Rondeau. A design process of switched ethernet architectures according to real-time application constraints. *EAAI, Engineering Applications of Artificial Intelligence*, 19(3):335–344, Apr. 2006.

[11] J.-P. Georges, E. Rondeau, and T. Divoux. Evaluation of switched ethernet in an industrial context by using the network calculus. In *4th IEEE International Workshop on Factory Communication Systems (WFCS'02)*, pages 19–26, Aug. 2002.

[12] J. Grieu. *Analyse et évaluation de techniques de commutation Ethernet pour l'interconnexion des systèmes avioniques*. PhD thesis, Institut National Polytechnique de Toulouse, Ecole doctorale informatique et télécommunications, Sept. 2004.

[13] J. Jasperneite, P. Neumann, M. Theis, and K. Watson. Deterministic real-time communication with switched ethernet. In *4th IEEE International Workshop on Factory Communication Systems (WFCS'02)*, pages 11–18, Aug. 2002.

[14] J. Jasperneite, M. Schumacher, and K. Weber. Limits of increasing the performance of industrial ethernet protocols. In *12th IEEE Conference on Emerging Technologies and Factory Automation (ETFA'07)*, pages 17–24, Sept. 2007.

[15] J.-Y. Le Boudec and P. Thiran. *Network calculus, a theory of deterministic queueing systems for the Internet*. Lecture Notes in Computer Science. Springer Verlag, 2001.

[16] L. Lenzini, E. Mingozzi, and G. Stea. A methodology for computing end-to-end delay bounds in fifo-multiplexing tandems. *Performance Evaluation*, 65(11-12):922–943, Nov. 2008.

[17] J. Robert, J.-P. Georges, T. Divoux, P. Miramont, and B. Rmili. Ethernet networks for real-time systems: application to launchers. In ESA, editor, *International Space System Engineering Conference, EUROSPACE, DASIA'11*, May 2011. To appear in.

[18] J. Schmitt, F. Zdarsky, and M. Fidler. Delay bounds under arbitrary multiplexing: When network calculus leaves you in the lurch... In *27th IEEE Conference on Computer Communications (INFOCOM'08)*, pages 1669–1677, Apr. 2008.

[19] J. Schmitt, F. Zdarsky, and I. Martinovic. Performance bounds in feed-forward networks under blind multiplexing. Technical Report 349/06, Distributed Computer Systems Lab, University of Kaiserslautern, Germany, 2006.

[20] J. Schmitt, F. Zdarsky, and L. Thiele. A comprehensive worst-case calculus for wireless sensor networks with in-network processing. In *28th IEEE International Real-Time Systems Symposium (RTSS'07)*, pages 193–202, Dec. 2007.

[21] L. Thiele, S. Chakraborty, M. Gries, A. Maxiaguine, and J. Greutert. Embedded software in network processors - models and algorithms. In *1st International Workshop on Embedded Software (EMSOFT'01)*, pages 416–434, 2001.