Improving Performance Bounds in Feed-Forward Networks by Paying Multiplexing Only Once

Jens B. Schmitt, Frank A. Zdarsky, and Ivan Martinovic
disco | Distributed Computer Systems Lab
University of Kaiserslautern, Germany

Abstract

Bounding performance characteristics in communication networks is an important and interesting issue. In this study we assume uncertainty about the way different flows in a network are multiplexed, we even drop the common FIFO assumption. Under so-called arbitrary multiplexing we derive new bounds for the tractable, yet non-trivial case of feed-forward networks. This is accomplished for pragmatic, but general traffic and server models using network calculus. In particular, we derive an end-to-end service curve for a flow of interest under arbitrary multiplexing, establishing what we call the *pay multiplexing only once principle*. Finally, we present some numerical results to compare our bounds against the best known bounds for networks of arbitrary multiplexing nodes.

1 Introduction

Bounding performance characteristics in communication networks is a fundamental issue and has important applications in network design and control. Network calculus, which is a set of relatively new developments that provide deep insights into flow problems encountered in networks of queues [9], provides a deterministic framework for worst-case analysis of delay and backlog bounds. The basic network calculus results pioneered by [5], [6] in the early 1990s, however, implicitly assume some form of per-flow treatment inside the network in that they only apply to tandems of nodes. Large-scale packet-switched networks as the Internet operate on large aggregates of traffic and are far-off from supporting any per-flow state operations, as for example, sophisticated per-flow scheduling. Nevertheless, as [9] pointed out in 2001: "The state of the art for aggregate multiplexing is surprisingly poor." Since then considerable efforts have been made to address issues related to bounding performance characteristics in networks of aggregate multiplexing: [4] gives a delay bound for general FIFO networks; [15] extends it to nodes that are allowed to time-stamp packets; [7] treats the case of feed-forward FIFO networks; [10] investigates the optimal delay bound under FIFO multiplexing; [14] gives additive delay bounds for tree topologies.

All of the above work assumes FIFO nodes. However in practice, as nicely argued in [8], many devices cannot be accurately described by a FIFO model because packets arriving at the output queue from different input ports may experience different delays when traversing a node. Therefore, in this work we drop the FIFO assumption and make essentially no assumptions on the way aggregates are multiplexed at servers, i.e., we assume arbitrary multiplexing:

Definition 1: (Arbitrary Multiplexing) If a node multiplexes several flows and the arbitration discipline between the flows accessing the service of the node is assumed to be unknown we call it an arbitrary multiplexing node.

Another issue that arises during the investigation of aggregate scheduling is stability, i.e., whether a finite delay bound exists [1]. For general networks of arbitrary topology it is still very much an open research problem under which circumstances a bound on the delay exists at all [9]. At the other end of the spectrum of topologies, we have tandem networks for which network calculus has become famous to deliver tight bounds for any utilization ≤ 1 (mainly building upon the celebrated concatenation theorem [9]). We concentrate on the middle ground between these two extremes: *feed-forward networks*.

Definition 2: (Feed-Forward Network) A network is feed-forward if it is possible to find a numbering of its links such that for any flow through the network the numbering of its traversed links is an increasing sequence.

It is well-known and easy to show that feed-forward networks are stable for any utilization ≤ 1 [9]. While many networks are obviously not feed-forward, many important instances are. Among these are: switched networks that typically use spanning trees for routing; wireless sensor networks that can often be modelled as a single-sink or multiple-sink topology that is feed-forward, in fact wireless sensor networks have even been proposed as an application field of network calculus [11]; MPLS multipoint-to-point label-switched paths which also have been tackled using network calculus in [14]. Furthermore, there are very effective techniques to make a general network feed-forward. One of the advanced techniques (in comparison to a simple spanning tree) is the so-called turn-prohibition algorithm [13].

2 Network Calculus Background

Network calculus is a min-plus system theory for deterministic queuing systems which builds on the calculus for network delay in [5], [6]. A detailed treatment of min-plus algebra and of network calculus can be found in [2] and [3], [9], respectively.

As network calculus is built around the notion of cumulative functions for input and output flows of data, the set of real-valued, non-negative, and wide-sense increasing functions passing through the origin plays a major role:

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

In particular, the input function $F(t)$ and the output function $F^*(t)$, which cumulatively count the number of bits that are input to respectively output from a system $\mathcal{S}$, are $\in \mathcal{F}$. Throughout the paper, we assume in- and output functions to be continuous in time and space. This is not a major restriction as there are transformations from discrete to continuous models which do not affect the accuracy of the results significantly [9].

Definition 3: (Min-plus Convolution and Deconvolution) The min-plus convolution respectively deconvolution of two functions f and g are defined to be

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

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

It can be shown that the triple $(\mathcal{F}, \wedge, \otimes)$, where $\wedge$ denotes the minimum operator (which ought to be taken pointwise for functions), constitutes a dioid [9]. Some rules most important to our investigation are summarized in the following theorem:

Theorem 1: (Properties of $\otimes$) Let $f, g, h \in \mathcal{F}$

1. Commutativity of $\otimes$: $f \otimes g = g \otimes f$
2. Associativity of $\otimes$: $(f \otimes g) \otimes h = f \otimes (g \otimes h)$
3. Convolution of concave function: if f and g are concave functions and $f(0) = g(0) = 0$ then $f \otimes g = f \wedge g$.
4. Convolution of piecewise linear convex functions: if f and g are piecewise linear convex functions, $f \otimes g$ is obtained by putting end-to-end the different linear pieces of f and g, sorted by increasing slopes.

The proof of these properties can be found in [9]. Let us turn now to the performance characteristics of flows that can be bounded by network calculus means:

Definition 4: (Backlog and Delay) Assume a flow with input function F that traverses a system $\mathcal{S}$ resulting in the output function F^*. The backlog of the flow at time t is defined as

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

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

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

Now the arrival and server processes specified by input and output functions are bounded based on the central network calculus concepts of arrival and service curves:

Definition 5: (Arrival Curve) Given a flow with input function R a function $\alpha \in \mathcal{F}$ is an arrival curve for R if and only if

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

$$\Leftrightarrow R \le R \otimes \alpha \Leftrightarrow \alpha \ge R \oslash R$$

Note that an arrival curve which is not sub-additive can be improved by its sub-additive closure [9]. As a further remark, remember that any concave function is sub-additive.

Definition 6: (Minimum and Maximum Service Curve) If the service provided by a system $\mathcal{S}$ for a given input function F results in an output function F^* we say that $\mathcal{S}$ offers a minimum service curve β respectively a maximum service curve $\bar{\beta}$ if and only if

$$F^* \geq F \otimes \beta \quad \text{respectively} \quad F^* \leq F \otimes \bar{\beta}$$

The minimum service curve is more commonly used (therefore the minimum is often dropped from its name), since the maximum service curve is a weaker concept. Nevertheless, the maximum service curve will prove itself valuable for our purposes in improving the output bound of interfering flows. There is also a *strict* (minimum) service curve definition which is less general than the minimum service curve but sometimes is required to enable an analysis.

Definition 7: (Strict Minimum Service Curve) Let $\beta \in \mathcal{F}$. We say that system $\mathcal{S}$ offers a *strict* minimum service curve β to a flow if, during any backlogged period of duration u, i.e., for any t for which $\forall s, 0 \leq s < u : b(t-s) > 0$, the output of the flow is at least equal to $\beta(u)$.

Note that any strict service curve is also a service curve. Using those concepts it is possible to derive basic performance bounds on backlog, delay and output:

Theorem 2: (Performance Bounds) Consider a system $\mathcal{S}$ that offers a minimum service curve β and a maximum service curve $\bar{\beta}$. Assume a flow F traversing the system has an arrival curve α. Then we obtain the following performance bounds:
Backlog: $\forall t : b(t) \leq (\alpha \oslash \beta)(0) =: v(\alpha, \beta)$
Delay: $\forall t : d(t) \leq \inf \{t \geq 0 : (\alpha \oslash \beta)(-t) \leq 0\} =: h(\alpha, \beta)$
Output (arrival curve α^* for F^*): $\alpha^* \leq \left(\alpha \otimes \bar{\beta}\right) \oslash \beta$

One of the strongest results of network calculus (albeit being a simple consequence of the associativity of $\otimes$) is the concatenation theorem that enables us to investigate tandems of systems as a single system:

Theorem 3: (Concatenation Theorem for Tandem Systems) Consider a flow that traverses a tandem of systems $\mathcal{S}_1$ and $\mathcal{S}_2$. Assume that $\mathcal{S}_i$ offers a minimum service curve β_i and a maximum service curve $\bar{\beta}_i$, $i = 1, 2$ to the flow. Then the concatenation of the two systems offers a minimum service curve $\beta_1 \otimes \beta_2$ and a maximum service $\bar{\beta}_1 \otimes \bar{\beta}_2$ to the flow.

So far we have only covered the tandem network case, the next result factors in the existence of other interfering flows. In particular, it states the minimum service curve available to a flow at a single node under cross-traffic from other flows at that node.

Theorem 4: (Arbitrary Multiplexing Nodal Service Curves) Consider a node arbitrarily multiplexing two flows 1 and 2. Assume that the node guarantees a *strict* minimum service curve β and a maximum service $\bar{\beta}$ to the aggregate of the two flows. Assume that flow 2 has α_2 as an arrival curve. Then

$$\beta_1 = [\beta - \alpha_2]^+$$

is a service curve for flow 1 if $\beta_1 \in \mathcal{F}$. $\bar{\beta}$ remains the maximum service curve also for flow 1 alone. Here, the $[.]^+$operator is defined as $[x]^+ = x \vee 0$, where $\vee$ denotes the maximum operator.

Note that we require the minimum service curve to be strict. In [9] an example is given showing that the theorem otherwise would not hold. It is *this theorem* that *we will generalize* to a feed-forward network case in Section 4 in such a way that the costs of multiplexing with interfering flows are only paid once, which forms the main contribution of this paper.

3 Network Calculus with Piecewise Linear Curves

To be able to compute performance bounds in networks of arbitrary multiplexing nodes, we need to apply basic network calculus operations like min-plus convolution and deconvolution on the arrival and service curves we encounter in a given scenario. This cannot be done on general functions with arbitrary characteristics but must, of course, eventually be done with specific instances of arrival and service curves. However, a class of functions which is both tractable as well as general enough to express all common candidates is that of piecewise linear functions. In the definition of piecewise linear arrival and service curves the following catalog of functions from $\mathcal{F}$ is helpful:

Definition 8: Auxiliary functions from $\mathcal{F}$

Burst delay functions: $\delta_T(t) = \begin{cases} +\infty & t > T \\ 0 & t \leq T \end{cases}$

Affine function (token bucket): $\gamma_{r,b}(t) = \begin{cases} rt + b & t > 0 \\ 0 & t \leq 0 \end{cases}$

Rate latency function: $\beta_{R,T}(t) = \begin{cases} R(t-T) & t > T \\ 0 & t \leq T \end{cases}$

From these functions general piecewise linear function can be constructed using the $\bigvee$ and $\bigwedge$ as we will encounter in the following. Note that, as mentioned in [9], it applies that $\forall f \in \mathcal{F} : (f \otimes \delta_T)(t) = f(t-T)$. Hence, a convolution with the burst delay function δ_T results in a shift along the x-axis according to the value of T.

3.1 Choice of Arrival and Service Curves

Arrival Curve. Let us assume a piecewise linear concave arrival curve:

$$\alpha = \bigwedge_{i=1}^{n} \gamma_{r_i,b_i}$$

On the one hand it is possible to already use such a function as a traffic source description in order to be able to closely approximate a source's worst-case behaviour. On the other hand, looking at the network analysis we cannot, as we will see when we factor in the maximum service curve, avoid to model arrival curves *inside* the network as general piecewise linear concave functions. This is due to the fact that they result from the addition of multiple flows induced by the multiplexing of the latter. To assume concavity is no major restriction because, as discussed in [9], non-concave functions (unless they are not sub-additive to be accurate) can be improved by pure knowledge of themselves, thus they cannot be tight.

Minimum Service Curve. Let us assume a piecewise linear convex minimum service curve:

$$\beta = \bigvee_{j=1}^{m} \beta_{R_j,T_j}$$

A piecewise linear convex minimum service curve results from deriving the service curve for a flow of interest at a node that arbitrarily multiplexes this flow with other flows which, as an aggregate, have a piecewise linear concave arrival curve. Again, it might also be useful to model a node's service curve using several linear segments. To assume convexity is not a major restriction, as it might for instance apply if a node also has other duties. Though, in contrast to the arrival curve which is not sensible for non-concave functions (non-sub-additive to be exact), there are potentially sensible non-convex service curves, although this is rather uncommon.

Maximum Service Curve. Let us assume a piecewise linear *almost concave* maximum service curve:

$$\bar{\beta} = \left(\bigwedge_{k=1}^{l} \gamma_{\tilde{r}_k, \tilde{b}_k} \right) \otimes \delta_L$$

By *almost concave* we mean that the curve is only concave for values of $t > L$ and is 0 for values $t \leq L$. If $L > 0$ this models a node that has a certain minimum latency. The piecewise concavity models again the fact that a node may also have other duties. The concavity in contrast to the convexity for the minimum service curve is due to the fact that a best-case instead of a worst-case perspective has to be taken.

3.2 Network Calculus Operations

We now have everything set to examine the network calculus operations we require as basic building blocks for a network analysis. First of all computing output bounds of the flows in the network is an important operation as it allows to separate flows of interest from interfering flows by using Theorem 4 respectively the results we present in Theorem 5. The general rule to compute the output bound is:

$$\alpha^* = \left(\left(\alpha \otimes \bar{\beta} \right) \oslash \beta \right)$$

Hence, starting from the innermost operation, the convolution of the arrival curve and the maximum service must be determined first:

$$\alpha \otimes \bar{\beta} = \alpha \otimes \left(\left(\bigwedge_{k=1}^{l} \gamma_{\tilde{r}_k, \tilde{b}_k} \right) \otimes \delta_L \right) = \left(\alpha \otimes \bigwedge_{k=1}^{l} \gamma_{\tilde{r}_k, \tilde{b}_k} \right) \otimes \delta_L$$

$$= \left(\alpha \wedge \left(\bar{\beta} \otimes \delta_{-L} \right) \right) \otimes \delta_L =: \sigma$$

Here we used first the associativity of $\otimes$, then rule 3 from Theorem 1. While σ might look complex, it is easy to compute: first shift the maximum service curve to the left by its latency then take the minimum with the concave arrival curve and shift the result to the right by the latency of the maximum service curve. Next, the deconvolution of the resulting almost concave function σ with the minimum service curve is calculated as:

$$\sigma \oslash \beta = (\sigma \otimes \delta_{-X}) - \beta(X) =: \zeta$$

where $X = \sup_{t \geq 0} \left\{ \frac{d\sigma}{dt}(t) \geq \frac{d\beta}{dt}(t) \right\}$. X must be at one of the inflexion points of σ and β. Note that ζ is concave since always $X \geq L$. So, in terms of the initial curves we receive as an overall result for the output bound:

$$\alpha^* = \left(\left(\left(\left(\alpha \wedge \left(\bar{\beta} \otimes \delta_{-L}\right)\right) \otimes \delta_L\right) \otimes \delta_X\right) - \beta\left(X\right)\right)$$

Equipped with this a single node analysis can be accomplished. However, another basic operation consists of using the concatenation theorem to collapse systems in sequence into one large system by calculating their min-plus convolution. So the convolution of piecewise linear minimum and maximum service curves must be computed. For the minimum service curves we can draw upon rule 4 from Theorem 1 as they are piecewise linear convex functions $\in \mathcal{F}$. For maximum service curves the next lemma gives us the computation of the min-plus convolution of two almost concave function:

Lemma 1: (Min-Plus Convolution of Almost Concave Functions) Consider two almost concave piecewise linear functions $\bar{\beta}_1$ and $\bar{\beta}_2$ with latencies L_1 and L_2, then their min-plus convolution can be computed as follows

$$\bar{\beta}_1 \otimes \bar{\beta}_2 = \left(\bar{\beta}_1 \otimes \delta_{-L_1}\right) \wedge \left(\bar{\beta}_2 \otimes \delta_{-L_2}\right) \otimes \delta_{L_1+L_2}$$

Proof: Again we make use of min-plus algebra

$$\begin{aligned}\bar{\beta}_1 \otimes \bar{\beta}_2 &= \left(\left(\bar{\beta}_1 \otimes \delta_{-L_1}\right) \otimes \delta_{L_1}\right) \otimes \left(\left(\bar{\beta}_2 \otimes \delta_{-L_2}\right) \otimes \delta_{L_2}\right) \\ &= \left(\left(\bar{\beta}_1 \otimes \delta_{-L_1}\right) \otimes \left(\bar{\beta}_2 \otimes \delta_{-L_2}\right)\right) \otimes \left(\delta_{L_1} \otimes \delta_{L_2}\right) \\ &= \left(\left(\bar{\beta}_1 \otimes \delta_{-L_1}\right) \wedge \left(\bar{\beta}_2 \otimes \delta_{-L_2}\right)\right) \otimes \delta_{L_1+L_2}\end{aligned}$$

Here we used associativity and commutativity of $\otimes$ as well as again rule 3 from Theorem 1. □

4 The End-to-End Service Curve under Arbitrary Multiplexing

Before we can present some results on network analysis applying the basic operations that were just presented, we investigate different alternatives to derive the end-to-end service curve for a flow of interest through a network of arbitrary multiplexing nodes. One possibility is to derive the end-to-end service curve based on the concatenation theorem and the result for single node arbitrary multiplexing in Theorem 4. This evident method is mentioned in [9]. For example, if a scenario as depicted in Figure 1 is to be analysed for flow 1, a *straightforward* end-to-end service curve for flow 1 would be determined as follows (using the notation in Figure 1):

$$\beta_1^{SF} = [\beta_1 - \alpha_2 - \alpha_3]^+ \otimes [\beta_2 - \alpha_2^* - \alpha_3^*]^+ \otimes [\beta_3 - \alpha_3^{**}]^+$$

Yet, another way to analyse the system is to concatenate node 1 and 2, subtract flow 2 and thus receive the service curve for flow 1 and 3 together, concatenate this with node 3 and subtract flow 3, essentially making optimal use of the sub-path sharing between the interfering flows:

$$\beta_1^{PS} = \left[[(\beta_1 \otimes \beta_2) - \alpha_2]^+ \otimes \beta_3 - \alpha_3\right]^+$$

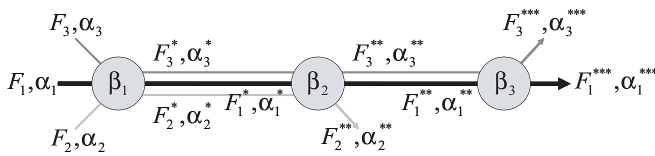

Figure 1: Nested interfering flows scenario.

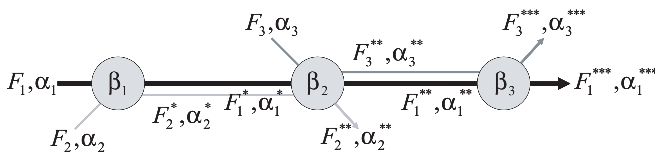

Figure 2: Overlapping interfering flows scenario.

If, for example, $\beta_i = \beta_{3,0}$, $i = 1,2,3$ and $\alpha_2 = \alpha_3 = \gamma_{1,1}$ we obtain: $\beta_1^{SF} = \beta_{1,12\frac{1}{2}}$ and $\beta_1^{PS} = \beta_{1,2}$. Hence, exploiting the sub-path sharing between the interfering flows, we obtain an end-to-end service curve with considerably lower latency. Put in other words, we have to pay for the arbitrary multiplexing of the flows only once, which is why we also call this phenomenon the pay multiplexing only once (PMOO) principle, in analogy to the well-known pay bursts only once principle [9]. Basically the same observation was made in [7] for *FIFO multiplexing* under the special case of token bucket arrival curves and rate-latency service curves. We derive the end-to-end service curve for *arbitrary multiplexing* exploiting the PMOO principle under *general piecewise linear concave arrival curves* and *general piecewise linear convex service curves*.

The difficulty in obtaining the end-to-end service under arbitrary multiplexing for a flow of interest lies in situations as depicted in Figure 2. Here, in contrast to the scenario of nested interfering flows as in Figure 1, we have a scenario of overlapping interfering flows, i.e., flows 2 and 3 which interfere with flow 1, our flow of interest, that share some servers with each other but each also traverses servers the other does not traverse. For such a scenario the end-to-end service curve cannot be derived as easily as demonstrated before but requires to look deeper into the input and output relationships of the flows. The following theorem states how to calculate the end-to-end service curve under arbitrary multiplexing exploiting the PMOO principle for the canonical example of overlapping interfering flows in Figure 2.

Theorem 5 (End-to-End Minimum Service Curve under Arbitrary Multiplexing – Pay Multiplexing Only Once Principle)
Consider a scenario as shown in Figure 2: a flow of interest F_1 interfered by two overlapping other flows F_2 and F_3. F_2 and F_3 have arrival curves α_2 and α_3. The three servers each offer a *strict* minimum service curve $\beta_i, i = 1,2,3$. The output flows of each of the servers are denoted as in Figure 2. If $\alpha_2 = \bigwedge_{i=1}^{n} \gamma_{r_i,b_i}$

and $\alpha_3 = \bigwedge_{j=1}^{m} \gamma_{\hat{r}_j,\hat{b}_j}$ are piecewise linear concave arrival curves then

$$\phi = \bigvee_{i=1}^{n} \bigvee_{j=1}^{m} \left[(\beta_1 - \tilde{\gamma}_{r_i,b_i}) \otimes (\beta_2 - \gamma_{r_i,0} - \gamma_{\hat{r}_j,0}) \otimes \left(\beta_3 - \tilde{\gamma}_{\hat{r}_j,\hat{b}_j} \right) \right]^+$$

constitutes an end-to-end service curve for the flow of interest, in particular $F_1^{***} \geq F_1 \otimes \phi$. Here we use a new notation: $\tilde{\gamma}_{r,b}(t) = \begin{cases} \gamma_{r,b}(t) & t \neq 0 \\ b & t = 0 \end{cases}$.

Proof: Since β_3 is a strict service curve we know that $\forall t \geq 0 : \exists u \geq 0 :$

$$\left(F_1^{***} + F_3^{***}\right)(t) \geq \left(F_1^{**} + F_3^{**}\right)(t-u) + \beta_3(u)$$

and $t-u$ is the start of the last backlogged period at node 3.

$$\implies F_1^{***}(t) - F_1^{**}(t-u) \geq \beta_3(u) - \left(F_3^{***}(t) - F_3^{**}(t-u)\right) \tag{1}$$

Along the same lines we can show for node 2 that $\forall (t-u) \geq 0 : \exists s \geq 0 :$
$\implies F_1^{**}(t-u) - F_1^{*}(t-u-s) \geq$

$$\beta_2(s) - \left(F_2^{**}(t-u) - F_2^{*}(t-u-s)\right) - \left(F_3^{**}(t-u) - F_3^{*}(t-u-s)\right) \tag{2}$$

and $t-u-s$ is the start of the last backlogged period at node 2 and the composed service at node 3 and 2.
And for node 1 that $\forall (t-u-s) \geq 0 : \exists r \geq 0 :$
$\implies F_1^{*}(t-u-s) - F_1^{*}(t-u-s-r) \geq$

$$\beta_1(r) - \left(F_2^{**}(t-u-s) - F_2^{*}(t-u-s-r)\right) \tag{3}$$

and $t-u-s-r$ is the start of the last backlogged period at node 1 and and the composed service at node 1,2, and 3.
If we add up (1), (2), and (3) we obtain $\forall t \geq 0 : \exists u, s, r \geq 0 :$

$$F_1^{***}(t) - F_1(t-u-s-r) \geq \beta_1(r) + \beta_2(s) + \beta_3(u)$$
$$-\left(F_2^{**}(t-u) - F_2(t-u-s-r)\right) - \left(F_3^{***}(t) - F_3(t-u-s)\right)$$

and $t-u-s-r$ is the start of the last backlogged period of the composed service at node 1, 2, and 3. Now using the causality of the systems under observation and the arrival curves for F_2 and F_3 we arrive at

$$F_1^{***}(t) - F_1(t-u-s-r) \geq \beta_1(r) + \beta_2(s) + \beta_3(u) - \alpha_2(s+r) - \alpha_3(s+u)$$

$$\geq \inf_{\substack{r', s', u' \geq 0 \\ r'+s'+u' = s+u+r}} \left\{\beta_1(r') + \beta_2(s') + \beta_3(u') - \alpha_2(s'+r') - \alpha_3(s'+u')\right\}$$

$$= \inf_{\substack{r', s', u' \geq 0 \\ r'+s'+u' = s+u+r}} \{\beta_1(r') + \beta_2(s') + \beta_3(u')$$

$$
- \bigwedge_{i=1}^{n} \gamma_{r_i,b_i}(s'+r') - \bigwedge_{j=1}^{m} \gamma_{\hat{r}_j,\hat{b}_j}(s'+u') \Bigg\}
$$

$$
= \inf_{\substack{r',s',u' \geq 0 \\ r'+s'+u'=s+u+r}} \Bigg\{ \bigvee_{i=1}^{n} \bigvee_{j=1}^{m} (\beta_1(r') + \beta_2(s') + \beta_3(u'))
$$

$$
-\gamma_{r_i,b_i}(s'+r') - \gamma_{\hat{r}_j,\hat{b}_j}(s'+u') \Big\}
$$

$$
\geq \bigvee_{i=1}^{n} \bigvee_{j=1}^{m} \inf_{\substack{r',s',u' \geq 0 \\ r'+s'+u'=s+u+r}} \{\beta_1(r') + \beta_2(s') + \beta_3(u')
$$

$$
-\gamma_{r_i,b_i}(s'+r') - \gamma_{\hat{r}_j,\hat{b}_j}(s'+u') \Big\} \tag{4}
$$

Next, we need to bring in the special shape of the arrival curves being additively separable. In particular, we have

$$
\begin{aligned}
\gamma_{r_i,b_i}(s'+r') &= \begin{cases} \gamma_{r_i,0}(s') + \tilde{\gamma}_{r_i,b_i}(r') & s'+r' > 0 \\ 0 & s'+r' = 0 \end{cases} \\
\gamma_{\hat{r}_j,\hat{b}_j}(s'+u') &= \begin{cases} \gamma_{\hat{r}_j,0}(s') + \tilde{\gamma}_{\hat{r}_j,\hat{b}_j}(u') & s'+u' > 0 \\ 0 & s'+u' = 0 \end{cases}
\end{aligned}
$$

Note that we have different *equivalent* choices for the additive separation of the token buckets. To return to (4) we need to distinguish four cases:
Case 1: $s'+r' > 0$ and $s'+u' > 0 \Rightarrow s+r > 0$ and $s+u > 0 \Rightarrow$ either $s > 0$ or $r,u > 0$. This corresponds to having a non-zero backlog at least at two of the three nodes or at node 2, thus

$$
F_1^{***}(t) - F_1(t-u-s-r)
$$

$$
\geq \bigvee_{i=1}^{n} \bigvee_{j=1}^{m} \inf_{\substack{r',s',u' \geq 0 \\ r'+s'+u'=s+u+r}} \{\beta_1(r') + \beta_2(s') + \beta_3(u')
$$

$$
-\gamma_{r_i,0}(s') - \tilde{\gamma}_{r_i,b_i}(r') - \gamma_{\hat{r}_j,0}(s') - \tilde{\gamma}_{\hat{r}_j,\hat{b}_j}(u') \Big\}
$$

$$
= \left(\bigvee_{i=1}^{n} \bigvee_{j=1}^{m} \phi_{i,j} \right) (u+s+r)
$$

with $\phi_{i,j} = (\beta_1 - \tilde{\gamma}_{r_i,b_i}) \otimes \left(\beta_2 - \gamma_{r_i,0} - \gamma_{\hat{r}_j,0}\right) \otimes \left(\beta_3 - \tilde{\gamma}_{\hat{r}_j,\hat{b}_j}\right)$ for $i = 1,\ldots,n$ and $j = 1,\ldots,m$.

Case 2: $s'+r' = 0$ and $s'+u' = 0 \Rightarrow s+r = 0$ and $s+u = 0 \Rightarrow s = r = u = 0$. This corresponds to having a non-zero backlog for the composed service at time

$t = t - u - s - r$, thus

$$F_1^{***}(t) - F_1(t-u-s-r) = 0 \geq \left(\bigvee_{i=1}^{n}\bigvee_{j=1}^{m}\phi_{i,j}\right)(0) = \bigvee_{i=1}^{n}\bigvee_{j=1}^{m} -\left(b_i + \hat{b}_j\right)$$

Case 3: $s' + r' > 0$ and $s' + u' = 0 \Rightarrow s + r > 0$ and $s + u = 0 \Rightarrow r > 0$ and $s = u = 0$. This corresponds to having a non-zero backlog only at node 1, thus

$$F_1^{***}(t) - F_1(t-u-s-r) \geq \bigvee_{i=1}^{n}\{\beta_1(r) + \tilde{\gamma}_{r_i,b_i}(r)\}$$

$$= \bigvee_{i=1}^{n}\{\beta_1(r) - r_i r - b_i\} \geq \left(\bigvee_{i=1}^{n}\bigvee_{j=1}^{m}\phi_{i,j}\right)(r) = \bigvee_{i=1}^{n}\bigvee_{j=1}^{m}\left\{\beta_1(r) - r_i r - b_i - \hat{b}_j\right\}$$

Case 4: $s' + r' = 0$ and $s' + u' > 0 \Rightarrow s + r = 0$ and $s + u > 0 \Rightarrow u > 0$ and $s = r = 0$. This corresponds to having a backlog only at node 3, thus (analogous to *Case 3*)

$$F_1^{***}(t) - F_1(t-u-s-r) \geq \bigvee_{j=1}^{m}\left\{\beta_3(u) + \tilde{\gamma}_{\hat{r}_j,\hat{b}_j}(u)\right\}$$

$$\geq \left(\bigvee_{i=1}^{n}\bigvee_{j=1}^{m}\phi_{i,j}\right)(u) = \bigvee_{i=1}^{n}\bigvee_{j=1}^{m}\left\{\beta_3(u) - \hat{r}_j u - \hat{b}_j - b_i\right\}$$

Taking all four cases together we arrive at $\forall t \geq 0 : \exists u, s, r \geq 0 :$

$$\left(\bigvee_{i=1}^{n}\bigvee_{j=1}^{m}\phi_{i,j}\right)(u+s+r) \leq F_1^{***}(t) - F_1(t-u-s-r) = F_1^{***}(t) - F_1^{***}(t-u-s-r)$$

The last equality is due to the fact that $t-u-s-r$ is the start of the last backlogged period for the service of the composed system. Since $F_1^{***} \in \mathcal{F}$ we obtain $\forall t \geq 0 : \exists u, s, r \geq 0 :$

$$F_1^{***}(t) - F_1(t-u-s-r) \geq \left[\left(\bigvee_{i=1}^{n}\bigvee_{j=1}^{m}\phi_{i,j}\right)(u+s+r)\right]^+$$

$$= \bigvee_{i=1}^{n}\bigvee_{j=1}^{m}[\phi_{i,j}(u+s+r)]^+ = \phi(u+s+r)$$

and $t-u-s-r$ is the start of the last backlogged period of the composed service system. This establishes ϕ as an end-to-end service curve for flow 1. □

As we are dealing with piecewise linear convex service curves we provide a corresponding application of Theorem 5 in the following corollary:

Corollary 1: (PMOO End-to-End Minimum Service Curve under Piecewise Linear Convex Nodal Service Curves) Under the assumptions of Theorem 5, suppose the three servers offer (strict) piecewise linear convex minimum service curves $\beta_1 = \bigvee_{i=1}^{n_i} \beta_{R_i,T_i}$, $\beta_2 = \bigvee_{j=1}^{n_j} \beta_{\hat{R}_j,\hat{T}_j}$, and $\beta_3 = \bigvee_{k=1}^{n_k} \beta_{\tilde{R}_k,\tilde{T}_k}$. With $\alpha_2 = \bigwedge_{l=1}^{n_l} \gamma_{r_l,b_l}$ and $\alpha_3 = \bigwedge_{m=1}^{n_m} \gamma_{\hat{r}_m,\hat{b}_m}$,

$$\phi = \bigvee_{i=1}^{n_i} \bigvee_{j=1}^{n_j} \bigvee_{k=1}^{n_k} \bigvee_{l=1}^{n_l} \bigvee_{m=1}^{n_m} \beta_{R_{i,j,k,l,m},T_{i,j,k,l,m}}$$

with

$$R_{i,j,k,l,m} = (R_i - r_l) \wedge \left(\hat{R}_j - r_l - \hat{r}_m\right) \wedge \left(\tilde{R}_k - \hat{r}_m\right)$$

and

$$T_{i,j,k,l,m} = T_i + \hat{T}_j + \tilde{T}_k + \frac{b_l + \hat{b}_m + r_l\left(T_i + \hat{T}_j\right) + \hat{r}_m\left(\hat{T}_j + \tilde{T}_k\right)}{(R_i - r_l) \wedge \left(\hat{R}_j - r_l - \hat{r}_m\right) \wedge \left(\tilde{R}_k - \hat{r}_m\right)}$$

constitutes a (piecewise linear convex) end-to-end service curve for the flow of interest, in particular $F_1^{***} \geq F_1 \otimes \phi$.

The corollary exhibits nicely the PMOO principle, since as can be observed in the latency terms of the rate-latency functions the burst terms of interfering flows are accounted for only once. The generalization to an arbitrary number of nodes is notationally complex but follows along the same line of argument as the three nodes with overlapping interference scenario, therefore it is left out here. It had of course to be taken into account for the implementation of the network calculus tool that is used for the mumerical investigations in Section 5. The derivation of the end-to-end maximum service curve is simple and given in the next lemma:

Lemma 2: (E2E Maximum Service Curve under Arbitrary Multiplexing) Consider a flow R that traverses a sequence of systems $\mathcal{S}_i$, $i = 1, \ldots, n$ where it is arbitrarily multiplexed with an arbitrary number of flows. Assume each of the systems offers a nodal maximum service $\bar{\beta}_i$, ß $= 1, \ldots, n$ then flow R is offered the following maximum service curve on its path

$$\bar{\beta} = \bigotimes_{i=1}^{n} \bar{\beta}_i$$

5 Numerical Experiments

To validate and evaluate the proposed methods for computing performance bounds in arbitrary multiplexing networks we used a toolbox called `DISCO Network Calculator` (described partially in [12]). This toolbox is more generally usable than just for the analysis of networks of arbitrary multiplexing nodes. For instance, we have also implemented FIFO service curves for general piecewise concave arrival curves and piecewise linear convex service curves. The

`DISCO Network Calculator`, which is written in Java^{TM}, is publicly available[1] and may hopefully be of use for other researchers interested in network calculus.

5.1 Experimental Design

In order to create a representative network we used the BRITE topology generator with the following recommended parameter settings: 2-level-top-down topology type with 10 ASes with 10 routers each, Waxman's model with $\alpha = 0.15, \beta = 0.2$, and random node placement with incremental growth type. The diameter of the generated topology is 11 hops. Hosts are created randomly at access routers: 30% random routers in each AS get a uniform random number of hosts between 1 and 5.

This topology is then transformed into a feed-forward network using the turn-prohibition algorithm. The turn-prohibition algorithm resulted in a change of 13% of the shortest path routes with an average change of 1.9 hops and a standard deviation of 1.2. So, altogether the routes were changed only by 0.25 hops on average. This can be considered a success for the turn-prohibition algorithm, because it was able to transform this general network with a diameter of 11 into a feed-forward network.

Now, for each source a flow is generated towards a randomly assigned sink. Each flow's arrival curve is drawn from a set $\mathbb{S}_\alpha$ of candidate arrival curves. Specifically, for this investigation we restricted the set members to token buckets. The minimum service curves were restricted to be rate-latency functions. These provide 1 of 3 rates: C, $2C$, or $3C$, where C is chosen such that the most loaded link can carry its load (plus a margin of 10%) and for the other links the rate is chosen closest to their carried load. The latency is chosen to be 10ms network-wide. If a maximum service curve is used its latency is set to zero and its rate equal to the rate of the corresponding nodal minimum service curve.

5.2 PMOO Analysis vs. Straightforward Analysis

The goals of this study was to find out how the straightforward analysis and the PMOO analysis compare to each other. We perform this comparison for two different traffic scenarios, one with low burstiness ($\mathbb{S}_\alpha = \left\{\gamma_{r,0.05[s]r}\right\}$, with $r = 10$ Mbit/s) and the other one with significantly higher burstiness ($\mathbb{S}_\alpha = \left\{\gamma_{r,0.05[s]r}, \gamma_{r,0.25[s]r}, \gamma_{r,0.5[s]r}\right\}$, with $r = 10$ Mbit/s). Furthermore, we want to isolate the effects by (1) the service curve computation using the PMOO principle, (2) the output bound computation using the PMOO principle, and (3) the maximum service curve. We use all available sinks for flow generation and compute 10 replications with different seeds. The averaged results can be found in Figure 3.

First we take a look at the effect of the PMOO service curve alone: It is considerably better than its straightforward counterpart for both low and high burstiness of the traffic. As expected, the burstier traffic incurs higher absolute delays, but the percentage of improvement is roughly the same for both types of flows: $\approx 66 \pm 6.5\%$ at a 95% level of confidence. When the PMOO output bound is used this gives a further significant improvement of $\approx 13\%$ for the PMOO analysis (for both traffic types). When the maximum service curve is

[1] http://disco.informatik.uni-kl.de/content/Downloads

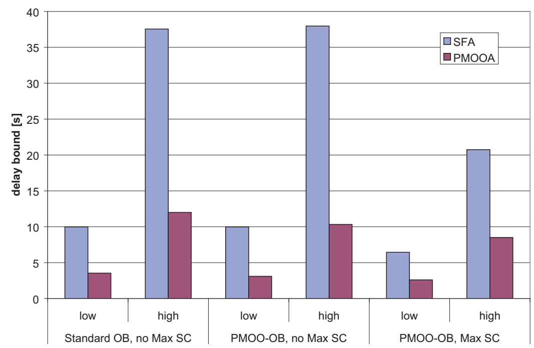

Figure 3: PMOO analysis (PMOOA) vs. straighforward analysis (SFA) in turn-prohibited networks.

used this reduces the bounds by another $\approx 16\%$. The maximum service curve also improves the straighforward analysis by $\approx 35\%$ (low) and $\approx 45\%$ (high) and thus keeps its promise to be valuable for a good delay bound analysis although it leads to more complex models.

6 Conclusion

In this paper, we have presented a comprehensive set of methods to address the issue of computing performance bounds in feed-forward networks under arbitrary multiplexing assumptions. Starting from the derivation of the basic operations for piecewise linear arrival and service curves required for the network analysis, we derived an interesting result on a novel way to compute the end-to-end service curve for a flow of interest under arbitrary multiplexing, again for the pretty general case of piecewise linear arrival and service curves. Equipped with these results we demonstrated that in realistic environments the new bounds can be significantly better, reducing the existing bounds by at least 50% in our scenarios.

References

[1] M. Andrews. Instability of fifo in session-oriented networks. In *Proc. SODA*, pages 440–447, March 2000.

[2] F. Baccelli, G. Cohen, G. J. Olsder, and J.-P. Quadrat. *Synchronization and Linearity: An Algebra for Discrete Event Systems*. Probability and Mathematical Statistics. John Wiley & Sons Ltd., West Sussex, Great Britain, 1992.

[3] C.-S. Chang. *Performance Guarantees in Communication Networks*. Telecommunication Networks and Computer Systems. Springer-Verlag, London, Great Britain, 2000.

[4] A. Charny and J.-Y. Le Boudec. Delay bounds in a network with aggregate scheduling. In *Proc. QofIS*, pages 1–13, September 2000.

[5] R. L. Cruz. A calculus for network delay, Part I: Network elements in isolation. *IEEE Transactions on Information Theory*, 37(1):114–131, January 1991.

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

[7] M. Fidler and V. Sander. A parameter based admission control for differentiated services networks. *Computer Networks*, 44(4):463–479, 2004.

[8] J.-Y. Le Boudec and A. Charny. Packet scale rate guarantee for non-fifo nodes. In *Proc. IEEE INFOCOM*, pages 23–26, June 2002.

[9] J.-Y. Le Boudec and P. Thiran. *Network Calculus A Theory of Deterministic Queuing Systems for the Internet.* Number 2050 in Lecture Notes in Computer Science. Springer-Verlag, Berlin, Germany, 2001.

[10] L. Lenzini, L. Martorini, E. Mingozzi, and G. Stea. Tight end-to-end per-flow delay bounds in fifo multiplexing sink-tree networks. *Perform. Eval.*, 63(9-10):956–987, 2006.

[11] J. Schmitt and U. Roedig. Sensor network calculus - a framework for worst case analysis. In *Proc. DCOSS*, pages 141–154, June 2005.

[12] J. Schmitt and F. Zdarsky. The DISCO Network Calculator - A Toolbox for Worst Case Analysis. In *Proceedings of the First International Conference on Performance Evaluation Methodologies and Tools (VALUETOOLS'06), Pisa, Italy.* ACM, November 2006.

[13] D. Starobinski, M. Karpovsky, and L. A. Zakrevski. Application of network calculus to general topologies using turn-prohibition. *IEEE/ACM Transactions on Networking*, 11(3):411–421, 2003.

[14] G. Urvoy-Keller, G. Hébuterne, and Y. Dallery. Traffic engineering in a multipoint-to-point network. *IEEE Journal on Selected Areas in Communications*, 20(4):834–849, May 2002.

[15] Z.-L. Zhang, Z. Duan, and Y. T. Hou. Fundamental trade-offs in aggregate packet scheduling. In *Proc. ICNP*, pages 129–137, November 2001.