Different Scenarios of Concatenation at Aggregate Scheduling of Multiple Nodes

Ulrich Klehmet Kai-Steffen Hielscher
Computer Networks and Communication Systems
Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany
Email: {klehmet, hielscher}@informatik.uni-erlangen.de

***Abstract*—Network Calculus (NC) offers powerful tools for performance evaluation in queueing systems. It has been proven as an important mathematical methodology for worst-case analysis of communication networks. One of its main application fields is the determination of QoS guarantees in packet switched communication systems. One issue of nowadays' research is the applicability of NC concerning the performance evaluation of aggregate multiplexing flows either at one node or at multiple nodes. Then, we have to differ whether the FIFO property at merging single flows can be assumed or not as in case of so-called *blind multiplexing*. In this paper, we are dealing with problems of computing the service curve for the single individual flow at demultiplexing in connection with aggregate scheduling of both – a singular service system (node) or of multiple nodes, at least two. These service curves are relevant for worst-case delay computation. In particular we define important application scenarios and compare their resulting single flow service curves. These are of practical benefit in many applications and can not be found in literature.**

***Index Terms*—Network Calculus; FIFO Multiplexing; Blind Multiplexing; Concatenation of nodes; Pay Multiplexing Only Once**

I. INTRODUCTION

In the framework of NC, the modelling elements *arrival curve* and *service curve* play an important role. They are the basis for the computation of maximal deterministic boundary values like backlog bounds and delay bounds found in [1], [2].

Definition 1 (Arrival curve): Given a system S with input flow $x(t)$. Let $\alpha(t)$ be a non-negative, non-decreasing function. $x(t)$ is constrained by or has arrival curve $\alpha(t)$ iff
$x(t) - x(s) \leq \alpha(t - s)$ for all $t \geq s \geq 0$.
Another speech is: flow F is α-smooth.

Example 1: A commonly used arrival curve is the token bucket constraint:

$\alpha_{r,b}(t) = b + rt$ for $t > 0$ and zero otherwise.

As one can see in Fig. 1 this arrival curve forms an upper limit for traffic flows $x(t)$ with (average) rate r and instantaneous burst b . That means $x(t)-x(s) \leq \alpha_{r,b}(t-s) = b+r\cdot(t-s)$. For $\Delta t := t - s$ and $\Delta t \to 0$ it holds

$$\lim_{t\to s}\{x(t) - x(s)\} \leq \lim_{\Delta t\to 0}\{r \cdot \Delta t + b\} = b$$

An important definition of NC is the following one:

Definition 2 (Min-plus convolution): Let $f(t)$ and $g(t)$ be non-negative, non-decreasing functions that are 0 for $t \leq 0$. A third function, called min-plus convolution is defined by

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

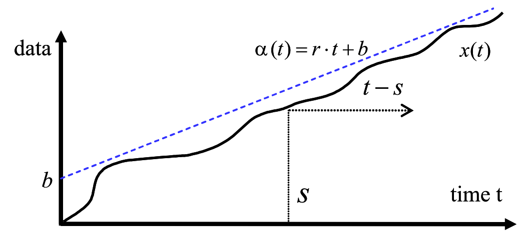

Fig. 1. Token Bucket Arrival Curve

Applying Definition 2 we can characterize the arrival curve $\alpha(t)$ with respect to $x(t)$ as:

$$x(t) \leq (x \otimes \alpha)(t)$$

The concept of arrival curves describes an upper bound to an input stream of a system processing some type of data. Concerning the output of this system we are interested in some service guarantees, i.e. is there a guaranteed minimum of output $y(t)$ – the amount of data leaving system S? The modeling element *service curve* deals with this problem.

Definition 3 (Service curve): Given a system S with input flow $x(t)$ and output flow $y(t)$. The system offers a (minimum) service curve $\beta(t)$ to the flow iff $\beta(t)$ is a non-negative, non-decreasing function with $\beta(0) = 0$ and $y(t)$ is lower bounded by the convolution of $x(t)$ and $\beta(t)$:

$$y(t) \geq (x \otimes \beta)(t).$$

Fig. 2 demonstrates $(x \otimes \beta)(t)$ as an example for the lower bound of the output $y(t)$ and any given input $x(t)$.

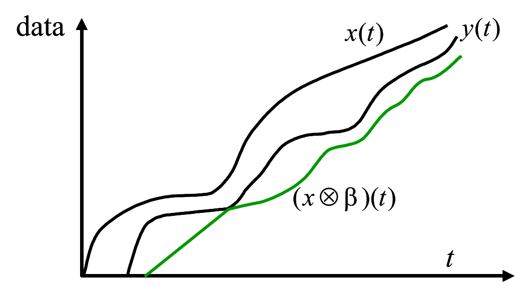

Fig. 2. Convolution as a Lower Output Bound

Example 2: One commonly used service curve is the rate-latency function: $\beta(t) = \beta_{R,T}(t) =$
$R \cdot [t - T]^+ := R \cdot \max\{0; t - T\}$. The rate-latency function reflects a service element which offers a minimum service of rate R after a worst-case latency of T. Having in mind a worst case performance analysis, it is possible to abstract away

from complex (queuing) systems with different scheduling strategies.
In Fig. 4, the (green) graph $\beta_{R,T}(t)$ reflects a rate-latency service curve with rate R and latency T.

Theorem 1 (Backlog bound and output bound): Consider a system S with input flow $x(t)$ and output flow $y(t)$. Be $x(t)$ α-smooth and S offers a service curve $\beta(t)$. The backlog v at time t, $v(t) = x(t) - y(t)$, is bounded by the supremum of the vertical deviation of arrival curve and service curve:

$$x(t) - y(t) \leq \sup_{s \geq 0}\{\alpha(s) - \beta(s)\}$$

and output $y(t)$ is constrained by the arrival curve

$$\alpha^*(t) = \alpha \oslash \beta = \sup_{s \geq 0}\{\alpha(t+s) - \beta(s)\}.$$

The complete backlog $v(t) = x(t) - y(t)$ at time t within a system is sometimes denoted as *buffer(t)*.
If the node or system serves the incoming data of a flow in FIFO order (First In First Out), the following bound is computable:

Theorem 2 (Delay bound): Assume a flow constrained by arrival curve $\alpha(t)$ passing a system with service curve $\beta(t)$. The maximal virtual delay d is given as the supremum of all possible virtual delays of data, i.e. is defined as the supremum of the horizontal deviation between arrival curve and service curve:

$$d \leq \sup_{s \geq 0}\{\inf\{\tau : \alpha(s) \leq \beta(s+\tau)\}\}.$$

Fig. 3 depicts both theorems.

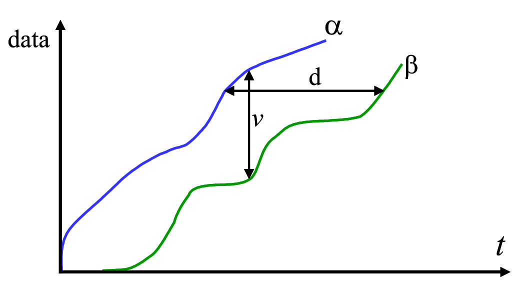

Fig. 3. Backlog and delay bound

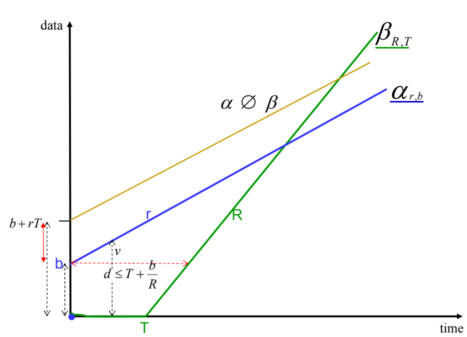

Fig. 4. Example for the bounds

Example 3: Suppose there is a system with input according to a token bucket, thus $x(t) - x(s) \leq \alpha_{r,b}(t-s)$ and rate-latency output:

$$y(t) \geq \inf_{s \leq t}\{x(s) + \beta_{R,T}(t-s)\}$$

Based on the above theorems we get the delay bound $d \leq b/R + T$, the output bound $\alpha^*(t) = r(t+T) + b$, and the backlog is bounded by $v = b + rT$. Fig. 4 shows the results.
Remark: Always in case of token-bucket like input and rate-latency output the worst-case delay d_{max} is computable by

$$d_{max} = \frac{burst}{servicerate} + latency.$$

II. AGGREGATE SCHEDULING

Until now, only per (single) flow-based scheduling have been considered. But in real systems, *aggregate scheduling* arises in many cases. Always, if there are more than one separat input flows entering some kind of data processing/transferring system and then dealt as a whole stream of data – we speak of aggregate scheduling. Important examples are aggregate based networks such as Differentiated Service domains (DS) of the Internet [3]. In order to address such class-based networks, we have to look for rules of multiplexing and aggregate scheduling. Assume that m flows enter a system (network) or system node and are scheduled by aggregation. According to [4] the aggregate input flow and arrival curve are given as follows.

Theorem 3 (Multiplexing): An aggregation, or multiplexing of m flows can be expressed by addition of the input functions respective arrival curves. W.l.o.g. be $m = 2$, then the aggregated input flow is $x(t) = x_1(t) + x_2(t)$ and $\alpha(t) = \alpha_1(t) + \alpha_2(t)$, where x_1, x_2 and α_1, α_2 are the corresponding single input flows and arrival curves.

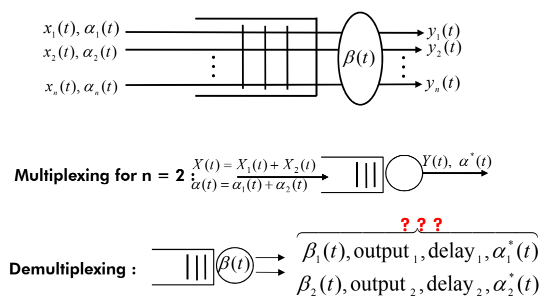

Fig. 5. Multiplexing of input x_i, output y_i with arrival & service curve α_i, $\beta = \beta aggr$

Now, as is shown in Fig. 5 important questions arise: Is it possible to apply the same analysis e.g. of buffer bounds and maximal delay of Theorems 1 and 2 to the single flows x_i? Does there exists a service curve β_i for the individual flow x_i, sometimes denoted as left-over service curve? What is the maximal delay, say of flow x_1, after servicing the aggregate and subsequently demultiplexing? The answers are based on the type of multiplexing in each case, i.e. in which manner the aggregate scheduling is done: **FIFO**, priority-scheduling, or multiplexing by complete unknown arbitration between the flows, which is the definition of **Blind** scheduling [5]. Together with the particular scheduling type one has to take into consideration the service curve of the aggregate flow. From a practical point of view we will discuss here the two important scheduling disciplines: FIFO and Blind. Regarding aggregate flow servers the next both theorems given by [1] are important.

Theorem 4 (FIFO Service curves): Consider a node serving the flows x_1 and x_2 in FIFO order. Assume first that the node guarantees a service curve β to the aggregate of the flows and secondly, flow x_2 is $\alpha_2 - smooth$. Define the family of functions $\beta^1_\theta(t) := [\beta(t) - \alpha_2(t-\theta)]^+ \ if \ t > \theta$ otherwise $\beta^1_\theta(t) := 0$. Then for any $\theta \geq 0$ it holds $y_1 \geq x_1 \otimes \beta^1_\theta$, where y_1 is the output of flow x_1. If β^1_θ is a non-negative, non-decreasing function, flow x_1 has the service curve β^1_θ.

Note $[x]^+ = x$ if $x \geq 0$ otherwise 0.

If no knowledge is given about the choice of service between the flows, i.e. in case of *blind multiplexing* one has to differ between *strict* or *non-strict* aggregate service curves [1].

Theorem 5 (Blind Multiplexing): Consider a node serving the flows x_1 and x_2, with some unknown arbitration between the two flows. Assume the node guarantees a strict service curve β to the aggregate of the two flows and that flow x_2 is $\alpha_2 - smooth$. Define $\beta_1(t) := [\beta(t) - \alpha_2(t)]^+$. If β_1 is wide-sense increasing, then it is a service curve for flow x_1.

But what does it mean, a service curve is *strict* ?

Definition 4 (Strict service curve): A system S offers a strict service curve β to a flow if during any backlogged period $[s,t]$ of duration $u = t - s$ the output y of the flow is at least equal to $\beta(u)$, i.e. $y(t) - y(s) \geq \beta(t-s)$, or equivalently $y(z) \geq \beta(z) \ \forall z \in [s,t]$.
Of course, any strict service curve is a service curve in terms of definition 3, but not vice versa - see for instance [1] or [6].

Example 4: The constant rate server in Fig. 6 with input flow x and output y has a strict service curve $\beta(t) = ct$. Let s be the start of a busy period, that means $y(s) = x(s)$, then $y(t) - y(s) = c(t-s)$, and so $y(t) - x(s) \geq \beta(t-s)$.

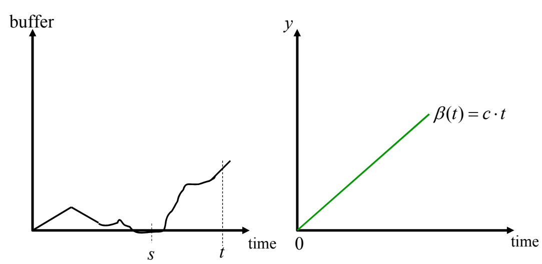

Fig. 6. Constant rate server

Our main objective in this paper is the consideration of *typical application scenarios* concerning multiplexed flows in FIFO or blind schedule situations: Based on Theorems 1 and 2 we want to apply the same analysis, e.g., for getting buffer bounds and maximal delay-values of for instance the single flows x_i after being demultiplexed. For that, what is the 'best' service curve β_i for the individual flow x_i, respectively?

Concerning most practical applications we focus in particular on input flows with token bucket like arrival constraints $\alpha_{r,b}$ and rate-latency service curves $\beta_{R,T}$.

A. Determination of the best service curve at FIFO scheduling:

First, let us come back to Theorem 4 for FIFO schedule in case of two flows x_1 and x_2. The main statement is that for any θ with $0 \leq \theta < t$ the expression $\beta^1_\theta(t) := [\beta(t) - \alpha_2(t-\theta)]^+$ is a service curve for flow x_1. Because that is valid for each $t = t_0$ – we may ask for which especial θ we get the 'best' service curve, i.e. the least pessimistic – or in other words the greatest β^1_θ, (so guaranteeing the least worst case delay etc.) Of course, since α_2 is a wide-sense increasing function – formula $\beta^1_\theta(t) := [\beta(t) - \alpha_2(t-\theta)]^+$ in general will get the largest value if θ is converging to t from left: $\theta \to t$ with $\theta < t$, for that we use the notation $\theta \to \ t_-$.
As we said before concerning practical applications, the arrival and service curves are often a token bucket-type $\alpha_{r,b}(t) = b + rt$ and rate-latency function $\beta_{R,T}(t) = R \cdot [t-T]^+$, respectively. Therefore, in order to demonstrate the search for a 'best service' we will take these both types of curves. That is to say get the supremum of
$\beta_{1,\theta} = [\beta_{R,T}(t) - \alpha_2(t-\theta)]^+$ with $\alpha_2(t) = r_2 t + b_2 \Rightarrow \sup_{0 \leq \theta < t}\{R \cdot (t-T)^+ - [r_2 \cdot (t-\theta) + b_2]\} = \sup_{0 \leq \theta < t}\{Rt - RT - r_2 t + r_2\theta - b_2\}$ which outcomes to $\theta = \theta_{opt} := T + \frac{b_2}{R}$. Thus, the 'best' rate-latency service curve is $\beta_{1,\theta} = \beta(t) - \alpha_2(t - (T + \frac{b_2}{R}))$ with $\beta(t) \ \ (= \beta_{aggr})$ as service curve to the aggregate.

If we now compare the service curves $\beta_{1,\theta}$ of both the FIFO ($\theta = \theta_{opt} = T + \frac{b_2}{R}$) and blind Multiplexing ($\theta = 0$) – of the same multiplexed server – one could expect in case of FIFO the service curve of single flow x_1 is larger, and consequently the better one w.r.t. the worst-case delay d_{max}. Let's denote this as $\beta_{1,FIFO}(t) > \beta_{1,Blind}(t)$. Our following computation will conform to this.

Blind Multiplexing:

$\beta_{1,\theta=0}(t) = \beta_{1,Blind}(t) = \beta(t) - \alpha_2(t-0) = R(t-T)^+ - (r_2 t + b_2) = \cdots = (R - r_2)[t - \frac{RT+b_2}{R-r_2}]^+$
The result is (again) a rate-latency service curve:
$\beta_{1,Blind}(t) = \beta_{R',T'}(t)$ with rate $R' = R - r_2$ and latency $T' = \frac{RT+b_2}{R-r_2}$.

FIFO Multiplexing:

$\beta_{1,\theta}(t) = \beta_{1,FIFO}(t) = \beta(t) - \alpha_2(t - \theta_{opt}) = R(t-T)^+ - (r_2 \cdot (t - (T + \frac{b_2}{R})) + b_2) = \cdots = (R - r_2)(t - [T + \frac{b_2}{R}]^+)$
The result, again a rate-latency service curve:
$\beta_{1,FIFO}(t) = \beta_{R',T'}(t)$ with rate $R' = R - r_2$ and latency $T' = T + \frac{b_2}{R}$.
It is easy to see:

$\beta_{1,FIFO} = (R - r_2)(t - [T + \frac{b_2}{R}])^+ > (R - r_2)[t - \frac{RT+b_2}{R-r_2}]^+ = \beta_{1,Blind}$.

 ISBN: 978-1-61208-256-1

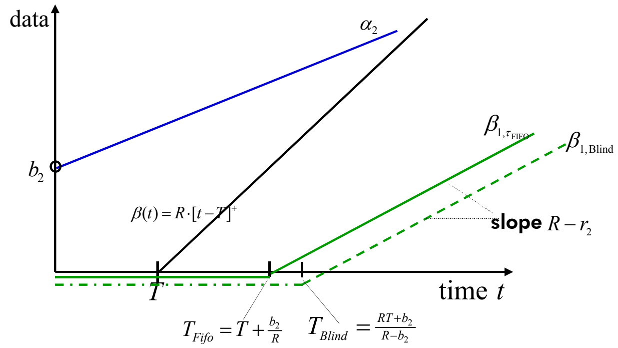

Fig. 7. Service curve FIFO vs. Blind of single flow x_1

In summary is to state:
For both we get the same service rate $R' = R - r_2$, however – as we expected – the latency increases from FIFO to Blind multiplexing. And because a service curve by definition 3 defines a lower output limit $\beta_{1,FIFO}$ specifies a greater lower limit to a single flow x_1 than $\beta_{1,Blind}$. Fig. 7 shows this issue.

Given now a service system multiplexing two flows x_1 and x_2. Theorems 4 or 5 provide a service curve β_i e.g. β_1 for the single flow x_1.
If x_1 is α_1-smooth and by Theorems 1 and 2 – the maximum backlog bound for the demultiplexed single x_1 is given by

$$x_1(t) - y_1(t) \leq \sup_{s \geq 0}\{\alpha_1(s) - \beta_1(s)\}$$

and the important worst case end-to-end-delay parameter of x_1 by

$$d \leq \sup_{t \geq 0}\{\inf\{\tau : \alpha_1(t) \leq \beta_1(t+\tau)\}\}$$

at which expression $d_\tau(t) = \inf\{\tau \geq 0 : \alpha_1(t) \leq \beta_1(t+\tau)\}$, the so-called *virtual delay*, is needed: If an input x at time t has arrived it is assured that not later than $d_\tau(t)$ it has left the service facility. This is guaranteed for FIFO scheduling but not for blind Multiplexing. However, we may presume FIFO per single flow x_i within the aggregate and thus apply all bounding theorems without any restrictions.

III. Different aggregate scheduling scenarios

So far, we have considered elementary service nodes (network elements). We now want to discuss the concatenation of aggregate network elements. First of all let's give the important theorem given in[1]:

Theorem 6 (Concatenation of nodes): Assume a flow traverses systems S_1 and S_2 in sequence and β_i is a service curve of S_i, i = 1, 2. Then the concatenation of the two systems offers a service curve of $\beta = \beta_1 \otimes \beta_2$ to the flow, like in Fig. 8.
Using this service curve $\beta = \beta_1 \otimes \beta_2$ we mention the important property [1] **Pay Burst Only Once (POO)**: Applying delay bound Theorem 2 one gets tighter end-to-end delay bounds if the delay computation is based on the concatenated end-to-end service curve β: $D_\otimes \leq D_1 + D_2$ with: $D_1 \leq \frac{b}{R_1} + T_1$, $D_2 \leq \frac{b+rT_1}{R_2} + T_2$ and $D_\otimes \leq \frac{b}{\min(R_1,R_2)} + (T_1 + T_2)$, again token-bucket and rate-latency curves supposed. (The burst b affects the sum $(D_1 + D_2)$ twice whereas $D_\otimes$ only once.)

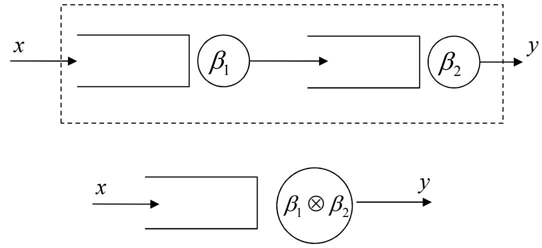

Fig. 8. Service curve of concatenated nodes

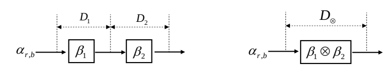

Fig. 9. Pay Burst Only Once-Principle

A. Concatenation of aggregated nodes

Now we will regard the concatenation of aggregate nodes, exemplarily for an input of two flows x_1, x_2 and a concatenated two-node system as shown in Fig. 10.

What is the end-to-end service curve of let's say flow x_1 ? By Theorem 6 and the (aggregation-) Theorems 4 (or 5) with $\beta^1_\tau(t) = [\beta(t) - \alpha_2(t-\tau)]^+$ we get:
$\beta^{tot}_1(t) = (\beta^I_{1,\tau} \otimes \beta^{II}_{1,\vartheta})(t)$
$= [\beta^I(t) - \alpha^I_2(t-\tau)]^+ 1_{t>\tau} \;\otimes\; [\beta^{II}(t) - \alpha^{II}_2(t-\vartheta)]^+ 1_{t>\vartheta}$,
where β^I, β^{II} are service curves of the aggregated flows of node I or node II, and α^I_2 and α^{II}_2 the arrival curves of the individual flow x_2 at the corresponding nodes.
(The term $1_{t>\theta}$ is zero for $t \leq \theta$. In the following, formulas, for the sake of clarity we will omit this term frequently). As

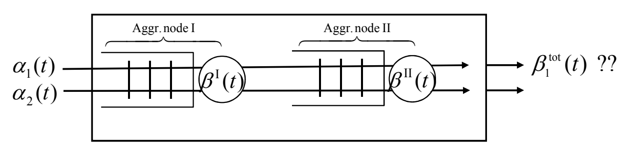

Fig. 10. Concatenation of aggregate nodes

given in Fig.10 flow x_1 is aggregated with x_2 only at node I, i.e. multiplexing happens only once. These thoughts lead to the PMOO-principle (Pay Multiplexing Only Once) [5]: First do the concatenation $\otimes$ of both nodes w.r.t. service curve β and afterwards apply Theorem 4 (or 5 in case of Blind):

$$\beta^{tot}_{1,PMOO}(t) = [(\beta^I_1 \otimes \beta^{II}_1)(t) - \alpha^I_2(t-\kappa)]^+.$$

Question: Is $\beta^{tot}_{1,PMOO}$ better than β^{tot}_1 or in other words $\beta^{tot}_{1,PMOO}(t) \geq \beta^{tot}_1(t)$?

Again, suppose: rate-latency service curves $\beta^I(t) = R^I \cdot [t - T^I]^+$, $\beta^{II}(t) = R^{II} \cdot [t - T^{II}]^+$ and token bucket arrival curves $\alpha^I_2(t) = r_2 \cdot t + b^I_2$, $\alpha^{II}_2(t) = r_2 \cdot t + b^{II}_2$.

At this point, we have to differ between FIFO and blind multiplexing, that means in formula $\beta^1_\tau(t) = [\beta(t) - \alpha_2(t-\tau)]^+$ we define $\tau = \tau_{opt} = T + \frac{b_2}{R}$

 ISBN: 978-1-61208-256-1

or $\tau = 0$, respectively.

*1) **Case FIFO**:* Let be $\tau = T^I + \frac{b_2^I}{R^I}$ of node I, $\vartheta = T^{II} + \frac{b_2^{II}}{R^{II}}$ of node II It follows:

- $\beta_1^{tot}(t) = \min(R^I - r_2, R^{II} - r_2) \cdot [t - T^I - T^{II} - \frac{b_2^I}{R^I} - \frac{b_2^{II}}{R^{II}}]^+$

And according to **PMOO** with $\tau = T^I + \frac{b_2^I}{R^I}$, $\vartheta = T^{II} + \frac{b_2^{II}}{R^{II}}$, $\kappa = (T^I + T^{II}) + \frac{b_2^I}{\min(R^I, R^{II})}$ we get

- $\beta_{1,PMOO}^{tot}(t) = [\min(R^I, R^{II}) - r_2] \cdot [t - T^I - T^{II} - \frac{b_2^I}{\min(R^I, R^{II})}]^+$

The results are two service curves β_1^{tot} or $\beta_{1,PMOO}^{tot}$ of flow x_1 again of type rate-latency.

Since $b_2^I \leq b_2^{II}$ it follows: $\beta_{1,PMOO}^{tot}(t) \geq \beta_1^{tot}(t)$.

Computing the worst-case delay D by $D = \frac{burst}{servicerate} + latency$, for $D = D_1$ or $D = D_{1,PMOO}$ and using β_1^{tot}, respectively $\beta_{1,PMOO}^{tot}$ we get:

- $D_1(t) = \frac{b_1}{\min(R^I - r_2, R^{II} - r_2)} + [T^I + T^{II} + \frac{b_2^I}{R^I} + \frac{b_2^{II}}{R^{II}}]$
- $D_{1,PMOO}(t) = \frac{b_1}{\min(R^I - r_2, R^{II} - r_2)} + [T^I + T^{II} + \frac{b_2^I}{\min(R^I, R^{II})}]$,

thus $D_{1,PMOO}(t) < D_1(t)$.

Result: $\beta_{1,PMOO}^{tot}$ is better than β_1^{tot}, since it produces a shorter worst case delay D.

*2) **Case Blind**:* $\tau = 0$, $\vartheta = 0$, $\kappa = 0$ (after Theorem 5) It follows for the end-to-end service curve $\beta_1^{tot}(t)$ of x_1:

- $\beta_1^{tot}(t) = [\min(R^I - r_2, R^{II} - r_2)] \cdot [t - (\frac{R^I T^I + b_2^I}{R^I - r_2} + \frac{R^{II} T^{II} + b_2^{II}}{R^{II} - r_2})]^+$

And applying the PMOO-principle here again, we get:

- $\beta_{1,PMOO}^{tot}(t) = [\min(R^I - r_2, R^{II} - r_2)] \cdot [t - \frac{\min(R^I, R^{II}) \cdot (T^I + T^{II}) + b_2^I}{\min(R^I, R^{II}) - r_2}]^+$

Unfortunately, now it is not always true: $\beta_{1,PMOO}^{tot} \geq \beta_1^{tot}$: $\beta_{1,PMOO}^{tot}$ per se does not causes less delay than β_1^{tot}. We get

$$\beta_{1,PMOO}^{tot} \geq \beta_1^{tot} \Leftrightarrow \begin{cases} (*) & b_2^{II} \geq \frac{r_2 T^{II}(R^{II} - R^I)}{R^I - r_2} \\ (**) & b_2^I \geq \frac{r_2 T^I (R^I - R^{II})}{R^{II} - r_2} \end{cases}$$

if (*) $\min(R^I, R^{II}) = R^I$ or (**) $\min(R^I, R^{II}) = R^{II}$. That means: $D_{1,PMOO}(t) < D_1(t)$ for condition (*) or (**).

B. More general concatenation settings

For practical application and comparisons we complete these scenarios and introduce the following definitions.

Definitions – FIFO:

$\beta_1^{FIFO} := [\beta^I(t) - \alpha_2^I(t - \tau)]^+ \otimes [\beta^{II}(t) - \alpha_2^{II}(t - \vartheta)]^+$

$\beta_{1,PMOO}^{FIFO} := [(\beta^I \otimes \beta^{II})(t) - \alpha_2^I(t - \kappa)]^+$

$\tilde{\beta}_{1,PMOO}^{FIFO} := [\beta^I(t) - \alpha_2^I(t - \tau)]^+ \otimes \beta^{II}(t)$ or

$\tilde{\beta}_{1,PMOO}^{FIFO} := \beta^I(t) \otimes [\beta^{II}(t) - \alpha_2^{II}(t - \tau)]^+$ with $\tau = T^I + \frac{b_2^I}{R^I}$, $\vartheta = T^{II} + \frac{b_2^{II}}{R^{II}}$ and $\kappa = (T^I + T^{II}) + \frac{b_2^I}{\min(R^I, R^{II})}$.

Definitions – Blind:

$\beta_1^{Blind} := [\beta^I(t) - \alpha_2^I(t - 0)]^+ \otimes [\beta^{II}(t) - \alpha_2^{II}(t - 0)]^+$

$\beta_{1,PMOO}^{Blind} := [(\beta^I \otimes \beta^{II})(t) - \alpha_2^I(t - 0)]^+$

$\tilde{\beta}_{1,PMOO}^{Blind} := [\beta^I(t) - \alpha_2^I(t - 0)]^+ \otimes \beta^{II}(t)$ or

$\tilde{\beta}_{1,PMOO}^{Blind} := \beta^I(t) \otimes [\beta^{II}(t) - \alpha_2^{II}(t - 0)]^+$ here $\tau = \vartheta = \kappa = 0$.

But what does it mean for instance
$(i) : [\beta^I(t) - \alpha_2^I(t - 0)]^+ \otimes \beta^{II}(t)$ or
$(ii) : \beta^I(t) \otimes [\beta^{II}(t) - \alpha_2^{II}(t - 0)]^+$?

Fig.11 explains in (i) and (ii) the semantic equivalent of first or second expression. In picture (i) the single flow x_2 with arrival curve α_2 leaves the system after served in node I, and in (ii) flow x_2 enters the system being served by node II only.

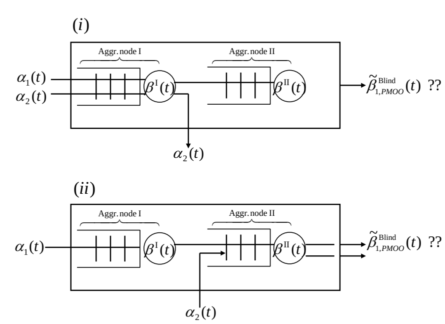

Fig. 11. Service curves of flow x_1 in (i) and (ii)

Using these definitions we come to the following results of the

Different scenarios:

Within FIFO:
$\tilde{\beta}_{1,PMOO}^{FIFO} \geq \beta_{1,PMOO}^{FIFO} \geq \beta_1^{FIFO}$

Within Blind:

- $\beta_{1,PMOO}^{Blind} \geq \beta_1^{Blind}$ if above condition (*) or (**) is given

 ISBN: 978-1-61208-256-1

- $\tilde{\beta}^{Blind}_{1,PMOO} \geq \beta^{Blind}_{1,PMOO}$ if $t \to \infty$
- $\tilde{\beta}^{Blind}_{1,PMOO} \geq \beta^{Blind}_{1,PMOO} \geq \beta^{Blind}_{1}$ if $t \to \infty$ and at condition (*) or (**)

Figure 12 shows the relations of left-over service curves between FIFO- and Blind-scheduling. Herein, the relations between the service curves symbolized by '?' are to be computed from case to case. Depending on the parameters $R^I, R^{II}, \tau, \vartheta$ and the concrete value of parameter t – both inequations are possible, either '$\geq$' or '$\leq$' respectively. Of course,

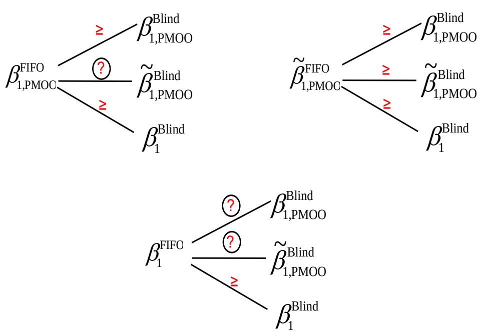

Fig. 12. Comparison between FIFO and Blind

one has to ask how to deal with more complex scenarios, e.g. in case of more than two aggregated flows or more than two service nodes. In principle we can apply an approach resulting from aggregation Theorems 3, 4 and 5 together with the concatenation Theorem 6. However one has to check whether the solutions are of practical benefit, may be the service curves $\beta_i(t)$ of single service x_i are too pessimistic which means they create to large worst-case delay bounds based on Theorem 2.

An example setting from [4] for FIFO scheduling with 3 flows and 3 server nodes is given here, where in Fig. 13 (*i*) flow 3 enters node II and after service is given out immediately. The other both flows are served by all 3 nodes. According to the aggregation and concatenation theorem we get:

$\beta_1(t) = [\beta^I(t) \otimes [(\beta^{II}(t) - \alpha_3^{II}(t-\tau))]^+ \otimes \beta^{III}(t) - \alpha_2^I(t-\vartheta)]^+$.

As before taking rate-latency service curves and token bucket arrival curves, $\tau = T^{II} + \frac{b_3^{II}}{R^{II}}$, and $\vartheta = T^I + T^{II} + T^{III} + \frac{b_2^I}{\min(R^I, R^{II}-r_3, R^{III})}$ thereupon resulting in $\beta_1(t) = [\min(R^I, R^{II} - r_3, R^{III}) - r_2] \cdot [t - T^I - T^{II} - T^{III} - \frac{b_3^{II}}{R^{II}} - \frac{b_2^I}{\min(R^I, R^{II}-r_3, R^{III})}]^+$.

The scenario in Fig. 13 (*ii*) w.r.t. left-over service of flow 2 leads to the end-to-end service curve

$\beta_2(t) = \min(R^I - r_1, R^{II} - r_3 - r_1, R^{III} - r_3) \cdot [t - T^I - T^{II} - T^{III} - \frac{b_1^I}{\min(R^I, R^{II}-r_3)} - \frac{b_3^{III}}{\min(R^{II}-r_1, R^{III})}]^+$.

Hereby flow 1 get service by node I and node II and leaves the system whereas flow 3 is served by node II and node III

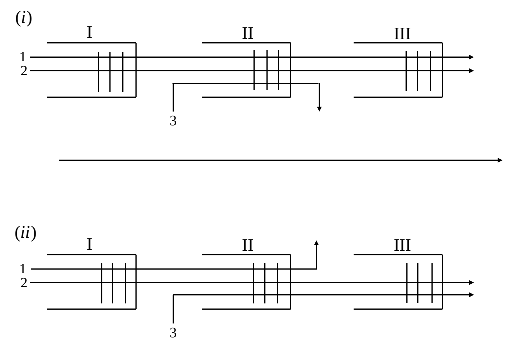

Fig. 13. Server networks with more flows and nodes

before leaving the server system. Only flow 2 get service by all 3 nodes.

IV. Conclusion

In this paper, we considered the subject of service curves in connection with aggregate scheduling mechanisms. Based on these service curves the maximum end-to-end delays of single flows x_i (left-over flow) after being demultiplexed are computable. In particular we discussed different scenarios of multiple aggregated nodes - which are typical for practical applications: Token bucket input flows and rate-latency service curves together with the main scheduling principles FIFO and Blind multiplexing. In a sense of case study we computed corresponding formulas and compared the results w.r.t. 'best service curves', i.e. the largest one and such producing the shortest worst-case end-to-end delays, which have great practical benefit for many hard real-time server systems.

In conclusion, for FIFO and Blind-scheduling of concatenated aggregation systems we computed service curves of demultiplexed single flows and compared them in different practice-relevant scenarios, which so far in the literature are not given. With our formulas and comparisons for single end-to-end service curves we move a step closer to allowing the design of complex systems.

References

[1] J.-Y. Le Boudec and P. Thiran, *Network Calculus*. Springer Verlag LNCS 2050, 2001.

[2] R. Cruz, "A calculus for network delay, part i: Network elements in isolation," *IEEE Trans. Inform. Theory*, vol. 37-1, pp. 114–131, 1991.

[3] A. Charny and J.-Y. Le Boudec, *Delay Bounds in a Network with Aggregate Scheduling*. Springer Verlag LNCS 1922, 2000.

[4] M. Fidler and V. Sander, "A parameter based admission control for differentiated services networks," *Computer Networks*, vol. 44, pp. 463–479, 2004.

[5] J. Schmitt, F. Zdarsky, and I. Martinovic, "Improving Performance Bounds in Feed-Forward Networks by Paying Multiplexing Only Once," in *Measurements, Modelling and Evaluation of Computer and Communication Systems(14th GI/ITG Conference), Dortmund*, March 2008.

[6] U. Klehmet and K.-S. Hielscher, "Strictness of Rate-Latency Service Curves," in *Data Communication Networking(3rd DCNETS Conference), DCNET/ICE-B/OPTICS, page 75-78. SciTePress, Rome*, July 2012.

 ISBN: 978-1-61208-256-1