Network calculus

João Luís Sobrinho
Instituto Superior Técnico
September 2009

I. MIN-PLUS CALCULUS

A. The semi-ring of order-preserving functions

The min-plus algebra is the bounded-complete idempotent dioid \((\mathbb{N}_0 \cup \{+\infty\}, \min, +, +\infty, 0)\). Often-
times, we use the infix operator \(\cap\) rather than the prefix operator \(\min\). Let \(\mathcal{F}\) be the set of order-preserving
functions, according to order \(\leq\), from \(\mathbb{N}_0\) to \(\mathbb{N}_0 \cup \{+\infty\}\). Define the minimum operation, \(\ominus\), on \(\mathcal{F}\) by
\[
(f \ominus g)(t) = f(t) \cap g(t),
\]
and the convolution operation, \(\otimes\), by
\[
(f \otimes g)(t) = \bigcap_{0 \leq s \leq t} \{f(s) + g(t - s)\},
\]
for all \(t \geq 0\). Further, let \(\epsilon(t)\) and \(\delta(t)\) be the following two special functions of \(\mathcal{F}\).
\[
\epsilon(t) = +\infty \quad \text{for all } t \geq 0.
\]
\[
\delta(t) = \begin{cases} 
0 & : t = 0 \\
+\infty & : t > 0.
\end{cases}
\]

The goal of this section is to show that \((\mathcal{F}, \ominus, \otimes, \epsilon, \delta)\) is a bounded-complete idempotent dioid.

**Proposition 1.1:** The triple \((\mathcal{F}, \cap, \otimes, \epsilon, \delta)\) is an idempotent monoid.

**Proof:** Let us show that \(f \cap g \in \mathcal{F}\) for all \(f, g \in \mathcal{F}\). Take \(s \leq t\). We have both \((f \cap g)(s) \leq f(s) \leq f(t)\) and \((f \cap g)(s) \leq g(s) \leq g(t)\) implying that \((f \cap g)(s) \leq f(t) \cap g(t) = (f \cap g)(t)\). Associativity, commutativity and the idempotent law are easy to establish as it is easy to see that \(\epsilon(t)\) is the identity of \(\cap\). \(\blacksquare\)

**Proposition 1.2:** The idempotent monoid \((\mathcal{F}, \cap, \epsilon)\) is bounded-complete. Moreover, given any set of functions \(\mathcal{S}, \mathcal{S} \subseteq \mathcal{F}\), we have
\[
\left(\bigcap \mathcal{S}\right)(t) = \bigcap \{f(t) | f \in \mathcal{S}\}.
\]

**Proof:** We have that \(g \leq f\) for all \(f \in \mathcal{S}\) if and only if \(g(t) \leq f(t)\) for all \(f \in \mathcal{S}\) and \(t \geq 0\). But \(g(t) \leq f(t)\) if and only if \(g(t) \leq \bigcap \{f(t) | f \in \mathcal{S}\}\) and this if and only if \(g \leq f'\) where \(f'\) is the function such that \(f'(t) = \bigcap \{f(t) | f \in \mathcal{S}\}\) for all \(t \geq 0\). Therefore, \(\cap \mathcal{S} = f'\). \(\blacksquare\)
Proposition 1.3: The triple \((\mathcal{F}, \otimes, \delta)\) is a commutative monoid.

Proof: Let us show that \(f \otimes g \in \mathcal{F}\) for all \(f, g \in \mathcal{F}\). Take \(s \leq t\). Because \(f\) and \(g\) are order-preserving, we have both

\[
f(u) + g(s - u) \leq f(u) + g(t - u) \quad \text{for } 0 \leq u \leq s
\]

and

\[
f(s) + g(0) \leq f(u) + g(t - u) \quad \text{for } s < u \leq t,
\]

implying

\[
(f \otimes g)(s) = \bigwedge_{0 \leq u \leq s} \{f(u) + g(s - u)\} \leq \bigwedge_{0 \leq u \leq t} \{f(u) + g(t - u)\} = (f \otimes g)(t).
\]

Let us show that \(\delta\) is the identity of \(\otimes\).

Proposition 1.4: The quintet \((\mathcal{F}, \cap, \otimes, \epsilon, \delta)\) is an idempotent, commutative semi-ring.

Proof: We have already shown that \((\mathcal{F}, \cap, \epsilon)\) is a commutative and idempotent monoid, and that \((\mathcal{F}, \otimes, \delta)\) is a commutative monoid. It is easy to see that \(\epsilon\) is an annihilator of \(\mathcal{F}\), so we are left to show that \(\otimes\) distributes over \(\cap\). We have

\[
(f \otimes (g \cap h))(t) = \bigwedge_{0 \leq s \leq t} \{f(s) + g(t - s) \cap h(t - s)\}
\]

\[
= \bigwedge_{0 \leq s \leq t} \{(f(s) + g(t - s)) \cap (f(s) + h(t - s))\}
\]

\[
= \left(\bigwedge_{0 \leq s \leq t} \{f(s) + g(t - s)\}\right) \cap \left(\bigwedge_{0 \leq s \leq t} \{f(s) + h(t - s)\}\right)
\]

\[
= ((f \otimes g) \cap (f \otimes h))(t).
\]
Proposition 1.5: The idempotent, commutative semi-ring \((\mathcal{F}, \sqcap, \otimes, \epsilon, \delta)\) is complete.

B. Classes of order-preserving functions

A function \(f \in \mathcal{F}\) is sub-unitarian if \(f \leq \delta\), which is equivalent to stating that \(f(0) = 0\).

Proposition 1.6: Given two sub-unitarian functions \(f, g \in \mathcal{F}\), we have \(f \otimes g \leq f \sqcap g\).

Proof: We have
\[
(f \otimes g)(t) = \bigcap_{0 \leq s \leq t} \{f(s) + g(t - s)\} \\
\leq (f(0) + g(t)) \sqcap (f(t) + g(0)) \\
= (f \sqcap g)(t).
\]

A function is concave if
\[
f(u) \geq \frac{t - u}{t - s} f(s) + \frac{u - s}{t - s} f(t) \quad \text{for all } s \leq u \leq t.
\]

Proposition 1.7: Given two sub-unitarian and concave functions \(f, g \in \mathcal{F}\), we have \(f \otimes g = f \sqcap g\).

Proof: We have
\[
f(s) + g(t - s) \geq \frac{t - s}{t} f(0) + \frac{s}{t} f(t) + \frac{s}{t} g(0) + \frac{t - s}{t} g(t) \\
= s f(t) + \frac{t - s}{t} g(t) \\
\geq f(t) \sqcap g(t),
\]

for all \(0 \leq s \leq t\). Thus,
\[
(f \otimes g)(t) = \bigcap_{0 \leq s \leq t} \{f(s) + g(t - s)\} \geq (f \sqcap g)(t),
\]

which together with the proposition above implies \(f \otimes g = f \sqcap g\).

A function is sub-additive if \(f(s + t) \leq f(s) + f(t)\) for all \(s\) and \(t\), which is equivalent to stating that \(f \otimes f \geq f\).

Proposition 1.8: A function \(f \in \mathcal{F}\) is sub-unitarian and sub-additive if and only if \(f \otimes f = f\).

Proof: The forward implication is obvious from Proposition 1.6. For the reverse implication, suppose first that \(f\) is not sub-additive. Then there are \(0 \leq s \leq t\) such that \(f(s) + f(t - s) < f(t)\), implying that \((f \otimes f)(t) < f(t)\), and, so, that \(f \otimes f \neq f\). Now suppose that \(f\) is not sub-unitarian. Then there is \(t\) such that \((f \otimes f)(t) > f(t)\), again implying that \(f \otimes f \neq f\).

Proposition 1.9: A function \(f \in \mathcal{F}\) is sub-unitarian and sub-additive if and only if \(f^* = f\), where \(f^*\) is the quasi-inverse of \(f\).
Proof: If \( f \) is sub-unitarian and sub-additive then, from Proposition 1.8, \( f \otimes f = f \). This implies \( f^k = f \) for \( k \geq 1 \) and, thus, \( f^* = \delta \cap f = f \). For the converse statement, note that \((\mathcal{F}, \wedge, \otimes, \epsilon, \delta)\) being a complete idempotent semi-ring, we have that \( f^* \otimes f^* = f^* \). Since, by hypothesis, \( f^* = f \), we get \( f \otimes f = f \). Invoking Proposition 1.8, we conclude that \( f \) is both sub-unitarian and sub-additive. \( \blacksquare \)

**Proposition 1.10**: Every sub-unitarian concave function is sub-additive.

**Proof**: Let \( f \in \mathcal{F} \) be sub-unitarian and concave. Then,

\[
f(t) \geq \frac{s}{t+s}f(0) + \frac{t}{t+s}f(t+s) = \frac{t}{t+s}f(t+s),
\]

and

\[
f(s) \geq \frac{s}{t+s}f(t+s).
\]

Adding the two inequalities yields the statement of the proposition. \( \blacksquare \)

**C. Especial functions**

The rate function \( \alpha_c, c \in \mathbb{N} \), is defined by

\[
\alpha_c(t) = ct \quad \text{for } t \geq 0.
\]

The rate functions are sub-unitarian, concave, and sub-additive. The affine function \( \gamma_{b,r}, b \in \mathbb{N}_0 \) and \( r \in \mathbb{N} \), is defined by

\[
\gamma_{b,r}(t) = \begin{cases} 
0 & : t = 0 \\
 b + rt & : t > 0.
\end{cases}
\]

The affine functions are sub-unitarian, concave, and sub-additive. Therefore,

\[
\gamma_{b,r} \otimes \gamma_{b',r'} = \gamma_{b,r} \cap \gamma_{b',r'},
\]

and

\[
\gamma^*_{b,r} = \gamma_{b,r}.
\]

The rate-latency function \( \beta_{d,r}, rd \in \mathbb{N}_0 \) and \( r \in \mathbb{N} \), is defined by

\[
\beta_{d,r}(t) = \max\{0, rt - rd\}.
\]

The rate-latency functions are sub-unitarian. We have

\[
\beta_{d,r} \otimes \beta_{d',r'} = \beta_{d+d',r\cap r'}.
\]

Note that

\[
\alpha_c = \gamma_{0,c} = \beta_{0,c}.
\]
II. QUEUES, BACKLOG AND DELAY

A queue is a device at which packets arrive, are possibly stored for some time, and then depart. Let \( A(t) \) describe the number of packets that arrived to a queue in the interval of time between 0 and \( t \) and \( B(t) \) describe the number of those packets that departed from the queue in the interval of time between 0 and \( t \). By convention, \( A(0) = 0 \). We assume that every queue is causal: a packet can only leave the queue after it has arrived. This implies \( B(t) \leq A(t) \) for all \( t \geq 0 \). We also assume that every queue can store an infinite number of packets but never loses a packet: every packet arriving at the queue will eventually depart. This implies that for any \( t \) there is \( t' \geq t \) such that \( B(t') \geq A(t) \). Functions \( A \) and \( B \) belong to \( F \) and are sub-unitarian.

The backlog, or queue length, at time \( t \) is

\[
Q(t) = A(t) - B(t)
\]

and the delay at time \( t \) is

\[
D(t) = \min\{\tau \mid A(t) \leq B(t + \tau)\}.
\]

Note that this delay corresponds to delay experienced by packets only if they depart from the queue in the order in which they arrived. This means that packets are scheduled for transmission in a first-in-first-out (FIFO) basis.

III. ARRIVAL CURVES AND REGULATORS

Given \( f \in F \) and \( f \) sub-unitarian, we say that a flow with arrival process \( A(t) \) is \( f \)-constrained if

\[
A(t) - A(s) \leq f(t - s),
\]

for all \( 0 \leq s \leq t \). Thus, the number of packet arrivals in any time interval of duration \( (t - s) \) cannot be greater than \( f(t - s) \). The condition is equivalent to

\[
A \leq A \otimes f.
\]

Because \( f \) is sub-unitarian, \( f \leq \delta \), and \( \otimes \) is isotone for \( \leq \), we also have \( A \otimes f \leq A \), from which we conclude that the flow is \( f \)-constrained if and only if

\[
A = A \otimes f.
\]

An \( f \)-regulator is a queue such that for any input \( A \) the corresponding output \( B \) is \( f \)-constrained. Because of causality, \( B \leq A \), we have \( B \sqcap A = B \). Therefore, the output \( B \) of an \( f \)-constrained regulator corresponding to input \( A \) satisfies the fixed-point equation

\[
B = (B \otimes f) \sqcap A.
\]
This equation has a maximal solution given by

\[ B = A \otimes f^*. \]

A regulator with this input-output relationship is called the maximal \( f \)–regulator. Equivalently, a maximal \( f \)–regulator is a regulator that never retains a packet that does not violate constraint function \( f \). Maximal regulators smooth the departure process of packets with minimal delay.

**Proposition 3.1:** The series of a maximal \( f \)–regulator with a maximal \( g \)–regulator is a maximal \( f \otimes g \)–regulator.

*Proof:* Let \( A \) be the input to the maximal regulator for \( f \). Its output is \( B = A \otimes f^* \) which is also the input to the maximal \( g \)–regulator. The output of this regulator is, thus, \( C = B \otimes g^* = A \otimes (f^* \otimes g^*) \).

Now, because \((F, \sqcap, \otimes, \epsilon, \delta)\) is a complete idempotent semi-ring and both \( f \) and \( g \) are sub-unitarian, we have \( f^* \otimes g^* = (f \otimes g)^* \).

**IV. SERVICE CURVES AND SCHEDULERS**

Given \( f \in F \) and \( f \) sub-unitarian, an \( f \)–server is a queue such that for any input \( A \) the corresponding output \( B \) satisfies

\[ B \geq A \otimes f. \]

**Proposition 4.1:** A queue is an \( f \)–server if and only if for every \( t \geq 0 \) there is \( s, 0 \leq s \leq t \), such that \( B(t) - A(s) \geq f(t - s) \).

**Proposition 4.2:** The series of an \( f \)–server with a \( g \)–server is an \( f \otimes g \)–server.

**Proposition 4.3:** Suppose that an \( f \)–constrained flow is input to a \( g \)–server. Then, the output is \( h \)–constrained with

\[
    h(t) = \begin{cases} 
        0 & : t = 0 \\
        \max_{0 \leq \tau \leq s} \{g(t + \tau) - f(\tau)\} & : t > 0.
    \end{cases}
\]

*Proof:* Denote the input to the server as \( A \) and the output as \( B \). We have, for \( s < t \),

\[
    B(t) - B(s) \leq A(t) - B(s) \\
    \leq A(t) - \min_{0 \leq \tau \leq s} \{A(\tau) + f(s - \tau)\} \\
    = \max_{0 \leq \tau \leq s} \{A(t) - A(\tau) - f(s - \tau)\} \\
    \leq \max_{0 \leq \tau \leq s} \{g(t - \tau) - f(s - \tau)\} \\
    = \max_{0 \leq \tau \leq s} \{g(t - s + \tau) - f(\tau)\} \\
    \leq \max_{0 \leq \tau} \{g(t - s + \tau) - f(\tau)\} \\
    = h(t - s).
\]
Proposition 4.4: Suppose that an \( f \)-constrained flow is input to a \( g \)-server. Then,
\[
Q(t) \leq \max_{0 \leq s} \{ f(s) - g(s) \},
\]
for all \( t \geq 0 \).

Proof: Denote the input to the server as \( A \) and the output as \( B \). We have
\[
Q(t) = A(t) - B(t) \\
\leq A(t) - \min_{0 \leq s \leq t} \{ A(s) + g(t - s) \} \\
= \max_{0 \leq s \leq t} \{ A(t) - A(s) - g(t - s) \} \\
\leq \max_{0 \leq s \leq t} \{ f(t - s) - g(t - s) \} \\
\leq \max_{0 \leq s \leq t} \{ f(s) - g(s) \}.
\]

Proposition 4.5: Suppose that an \( f \)-constrained flow is input to a \( g \)-server. Then,
\[
D(t) \leq \max_{0 \leq s} \min \{ \tau \mid f(s) \leq g(s + \tau) \},
\]
for all \( t \geq 0 \).

Proof: Denote the input to the server as \( A \) and the output as \( B \). From the definition of delay, we have that \( A(t) > B(t + D(t) - 1) \). From the definition of a \( g \)-server we know that there is \( s \), \( 0 \leq s \leq t + D(t) - 1 \), such that
\[
A(t) > B(t + D(t) - 1) \\
\geq A(s) + g(t + D(t) - 1 - s).
\]
Note that \( A(s) \geq A(t) > B(t + D(t) - 1) \) for \( s \geq t \). Therefore, we have \( s < t \). The flow is \( f \)-constrained. Hence,
\[
A(s) + f(t - s) \geq A(t) \\
\geq A(s) + g(t - s + D(t) - 1),
\]
which is equivalent to
\[
f(t - s) > g(t - s + D(t) - 1).
\]
We conclude that
\[
D(t) - 1 < \min \{ \tau \mid f(t - s) \leq g(t - s + \tau) \},
\]
and, finally,
\[
D(t) \leq \max_{0 \leq s} \min \{ \tau \mid f(s) \leq g(s + \tau) \}.
\]
V. APPLICATIONS

A. Token bucket regulators

A *token bucket regulator* with parameter \((b, r)\) is a queue which operates as follows. It has a token bucket which initially holds \(b\) tokens, and then, at any given time:

1) New packet arrivals are added to the queue and \(r\) new tokens are added to the token bucket;
2) As many packets as possible depart with the only restriction that each packet consumes one token;
3) Tokens in excess of \(b\) in the token bucket are discarded.

**Proposition 5.1:** A token bucket regulator is a maximal \(\gamma_{b,r}\)−regulator.

**Proof:** At any time \(s \geq 0\) the token regulator holds at most \(b\) tokens. Between time \(s\) and time \(t \geq s\) at most \(r(t - s)\) new tokens arrive at the token bucket. Therefore, between time \(s\) and time \(t\) at most \(b + r(t - s)\) packets depart: the token bucket regulator is a \(\gamma_{b,r}\)−regulator. Moreover, if the token bucket regulator is initially full of tokens then it never retains a packet that does not violate constraint function \(\gamma_{b,r}\). In conclusion, the token bucket regulator is a maximal \(\gamma_{b,r}\)−regulator. ■

B. Work conserving links

A *work conserving link* of capacity \(c\) operates as follows. At any given time:

1) New packet arrivals are added to the queue;
2) As many as possible packets depart from the queue up to a maximum of \(c\) packets.

**Proposition 5.2:** Let \(A\) be the arrival process at a work conserving link of capacity \(c\). Then, the departure process is given by

\[
B(t) = (A \otimes \alpha_c)(t) = \min_{0 \leq s \leq t} \{A(s) - c(t - s)\}.
\]

**Proof:** Noting that a work conserving link of capacity \(c\) behaves as a token bucket regulator with parameter \((0, c)\), we immediately have \(B = A \otimes \alpha_c^c = A \otimes \alpha_c\).

We now provide a direct proof. The number of packets departing at time \(t\) is the minimum between \(c\) and the number of packets backlogged at \((t - 1)\), \(A(t - 1) - B(t - 1)\), added to the number of packets arriving at \(t\), \(A(t) - A(t - 1)\). From this we can write

\[
B(t) = B(t - 1) + \min\{A(t) - B(t - 1), c\} = \min\{A(t), B(t - 1) + c\}.
\]

Solving this recursion yields the statement of the proposition. ■

A work conserving link is an \(\alpha_c\)−server. Suppose that an \(\gamma_{b,r}\)−constrained flow is input to a work conserving link of capacity \(c\), \(r \leq c\). From the results of the previous section, we conclude that the output flow is also \(\gamma_{b,r}\)−constrained, the backlog is upper bounded by \(b\) and the delay by \(b/c\). Slightly better bounds for the backlog and delay are \(b + r - c\) and \((b + r - c)/c\), respectively.
C. Generalized Processor Scheduling (GPS) schedulers

Suppose that two flows, 1 and 2, arrive at a work conserving link of capacity $c$. This queue is a GPS scheduler if it dispatches at least $c_i$, $i = 1, 2$, packets of flow $i$ at any given time when flow $i$ is backlogged, with $c_1 + c_2 = c$. It is easy to see that a GPS scheduler is a server for $c/c_i$ for flow $i$. If flow $i$ is $\gamma_{b_i,r_i}$-constrained, with $r_i \leq c_i$, then that the backlogged is upper bounded by $b_i$ and the delay is upper bounded by $b_i/c_i$.

D. Priority schedulers

Suppose that two flows, 1 and 2, arrive at a work conserving link of capacity $c$. This queue is a priority scheduler if at any given time it always dispatches packets of flow 1 before packets of flow 2.

Proposition 5.3: Consider a priority scheduler for flows 1 and 2 with cumulative arrival processes $A_1$ and $A_2$, respectively. If flow 1 is $f_1$-constrained, then the queue behaves as a $g_2$-server for flow 2 with

$$g_2(t) = \max\{0, ct - f_1(t)\}.$$ 

Proof: Packets of flow 1 are dispatched by the priority scheduler as if there were no packets of flow 2. We have

$$B_1(t) = \min_{0 \leq s \leq t} \{A_1(s) + c(t - s)\}$$
$$B_1(t) + B_2(t) = \min_{0 \leq s \leq t} \{A_1(s) + A_2(s) + c(t - s)\}.$$ 

Therefore, we can write

$$B_2(t) = \min_{0 \leq s \leq t} \{A_1(s) + A_2(s) - c(t - s)\} - \min_{0 \leq \tau \leq t} \{A_1(\tau) + c(t - \tau)\}$$
$$= \min_{0 \leq s \leq t} \{A_2(s) + \max_{0 \leq \tau \leq t} \{A_1(s) + c(t - s) - A_1(\tau) - c(t - \tau)\}\}$$
$$\geq \min_{0 \leq s \leq t} \{A_2(s) + \max\{0, c(t - s) + A_1(s) - A_1(t)\}\},$$

where we took $\tau = s$ and $\tau = t$ in the last inequality. Because flow 1 is $f_1$-constrained, we finally have

$$B_2(t) \geq \min_{0 \leq s \leq t} \{A_2(s) + \max\{0, c(t - s) - f_1(t - s)\}\},$$

concluding the proof. ■

Suppose that flow 1 is $\gamma_{b_1,r_1}$-constrained and flow 2 is $\gamma_{b_2,r_2}$-constrained, with $r_1 + r_2 \leq c$. Then, the service curve for flow 2 is $\max\{0, (c - r_1)t - b_1\} = \beta_{b_1/(c-r_1),c-r_1}$, which is a rate-latency curve with delay $b_1/(c - r_1)$ and rate $c - r_1$. The backlog and delay for flow 1 are upper bounded by $b_1$ and $b_1/c$, respectively. The backlog for flow 2 is upper bounded by

$$b_2 + \frac{r_2 b_1}{c - r_1},$$

and the delay is upper bounded by

$$\frac{b_1 + b_2}{c - r_1}.$$
A busy period for flow 1 begins at time $s$ and ends at a time $t$ if $Q(s - 1) = 0$, $A(s) > A(s - 1)$, $Q(u) > 0$ for $s \leq u < t$, and $Q(t) = 0$. The length of the busy period is $t - s$. In a busy period for flow 1, we witness $c$ departures in each of the instants of time $s, s + 1, \ldots, t - 1$ for a total of $c(t - s)$ departures during the busy period. At least that many packets must arrive during the same period. Since flow 1 is $\gamma_{b_1, r_1}$-constrained at most $b_1 + r_1(t - s)$ can arrive during the busy period. It must be the case that
\[
c(t - s) \leq b_1 + r_1(t - s),
\]
which yields
\[
t - s \leq \frac{b_1}{c - r_1}.
\]
Since the scheduler gives priority to packets of flow 1, a packet from flow 2 may have to wait for the duration of a busy period for flow 1 to be served. Hence, the scheduler behaves as a $\beta_{b_1/(c - r_1), c - r_1}$-server for flow 2.

E. Earliest Deadline First (EDF) schedulers

Suppose that two flows, 1 and 2, arrive at a work conserving link of capacity $c$. Each packet is assigned a deadline which is no sooner than its arrival time. The queue is an EDF scheduler if it schedules packets for transmission in non-decreasing order of their deadlines. Let $A_i$ be the cumulative arrival process of flow $i = 1, 2$. We assume that the deadlines within a flow are non-decreasing. Let $N_i(t)$ be the number of packets from flow $i$ with deadlines less than or equal to time $t$. The $k$-th packet from flow $i$ is assigned deadline $D_{i,k}$ with
\[
D_{i,k} = \min\{t \mid N_i(t) \geq k\}.
\]

Proposition 5.4: Consider an EDF scheduler for flows 1 and 2 with cumulative arrival processes $A_1$ and $A_2$, respectively. If every packet departs within its deadline, then
\[
N_1 + N_2 \leq (A_1 + A_2) \otimes \alpha_c.
\]

Proof: Let $B$ be the cumulative departure process. We have seen before that
\[
B = (A_1 + A_2) \otimes \alpha_c.
\]
The assumption that every packet is dispatched within its deadline implies
\[
N_1 + N_2 \leq B = (A_1 + A_2) \otimes \alpha_c.
\]

Proposition 5.5: Consider an EDF scheduler for flows 1 and 2 with cumulative arrival processes $A_1$ and $A_2$, respectively. If
\[
\sum_{i \in S} N_i \leq \sum_{i \in S} A_i \otimes \alpha_c,
\]
for every $S$ a subset of $\{1, 2\}$, then all packets depart within their deadlines.

**Proof:** We prove the contrapositive statement. Let $t$ be the first instant of time at which a packet, say from flow $i$, is known to miss its deadline. This packet has deadline $t$ and it is left backlogged at that time. Let $s'$ be the last instant of time before $t$ such that there are no backlogged packets with deadline less than or equal to $t$ at time $s'$. Because there is a backlogged packet at $t$ with deadline $t$, we gave $s' < t$. Let $S$ be the set of flows for which at least one packet with deadline less than or equal to $t$ arrives in any of the instants of time $s' + 1, \ldots, t$. Clearly, $i \in S$. Since the deadlines within a flow are non-decreasing, there are no backlogged packets of flow $j \in S$ at time $s'$, implying

$$B_j(s') = A_j(s').$$

The $c(t - s')$ packets that depart during the instants $s' + 1, \ldots, t$ all have deadlines less than or equal to $t$ and each must have belong t a flow of $S$. Therefore,

$$c(t - s') = \sum_{j \in S} (B_j(t) - B_j(s')) = \sum_{j \in S} (B_j(t) - A_j(s')).$$

Again because the deadlines are non-decreasing, no packet from a flow $j \in S$ with deadline greater than $t$, could have departed before or at time $t$:

$$B_j(t) \leq N_j(t).$$

Since the packet that missed its deadline is backlogged at time $t$, we further have

$$B_i(t) < N_i(t).$$

Summing up the inequalities above yields

$$\sum_{j \in S} N_j(t) > \sum_{j \in S} B_j(t) = \sum_{j \in S} A_j(s') - c(t - s') \geq \min_{0 \leq s \leq t} \left\{ \sum_{j \in S} A_j(s) - c(t - s) \right\},$$

thereby concluding the proof.

In the Service Curve Earliest Deadline (SCED) first algorithm, each flow $i$ has an associated function $g_i$, belonging to $F$ and sub-unitarian, and assigns deadlines to its packets such that $N_i = A_i \otimes g_i$. The packets are submitted to an EDF scheduler.

**Proposition 5.6:** Suppose that $N_i = A_i \otimes g_i$, with $g_i \in F$ and sub-unitarian. If

$$g_1 + g_2 \leq \alpha_c,$$
then all packets depart within their deadlines.

In the Service Curve Earliest Deadline (SCED) first algorithm, each flow $i$ has an associated function $g_i$, belonging to $F$ and sub-unitarian, and assigns deadlines to its packets such that $N_i = A_i \otimes g_i$. The packets submitted to an EDF scheduler.

**Proposition 5.7:** Suppose that $N_i = A_i \otimes g_i$, with $g_i \in F$ and sub-unitarian. If $g_1 + g_2 \leq \alpha_c$, then all packets depart within their deadlines.

**Proof:** We verify the condition of Proposition 5.4.

\[
\sum_{j \in S} N_j(t) = \sum_{j \in S} \min_{0 \leq s \leq t} \{A_j(s) + g_j(t - s)\} \\
\leq \min_{0 \leq s \leq t} \{\sum_{j \in S} A_j(s) + \sum_{j \in S} g_j(t - s)\} \\
\leq \min_{0 \leq s \leq t} \{\sum_{j \in S} A_j(s) + g_1(t - s) + g_2(t - s)\} \\
\leq \min_{0 \leq s \leq t} \{\sum_{j \in S} A_j(s) + c(t - s)\}.
\]

Moreover, if all packets depart within their deadlines, then $B_i \geq N_i = A_i \otimes g_i$ meaning that the EDF scheduler is a $g_i$-server for flow $i$. The next proposition shows that the deadlines of the packets can be computed in real-time.

**Proposition 5.8:** Suppose that under SCED, the $k$-packet from flow $i$ arrives at time $t$. Then,

\[
D_{i,k} = \min \{\tau \mid \tau \geq t \text{ and } \min_{0 \leq s < t} \{A_i(s) + g_i(\tau - s)\} \geq k\}.
\]

**Proof:** By definition

\[
D_{i,k} = \min \{\tau \mid (A_i \otimes g_i)(\tau) \geq k\}.
\]

The $k$-th packet from flow $i$ arrives at $t$ so that $A_i(u) < k$ for $u < t$. We have $(A_i \otimes g_i)(u) \leq A_i(u) < k$ for $u < t$ implying that $\tau \geq t$.

For all $s$ such that $t \leq s \leq \tau$, we have $A_i(s) + g_i(\tau - s) \geq A_i(t) \geq k$. Therefore,

\[
\min_{0 \leq s \leq \tau} \{A_i(s) + g_i(\tau - s)\} < k
\]

if and only if

\[
\min_{0 \leq s < t} \{A_i(s) + g_i(\tau - s)\} < k,
\]

and so we can write $D_{i,k}$ as stated in the proposition.
Proposition 5.9: Suppose that the arrival curve $A_i$ of the $i$-th flow, $i = 1, 2$, is $f_i$-constrained. Under SCED, the server is an $g_i$-server for $A_i$ if

$$\sum_{i=1}^{2} f_i \otimes g_i \leq \alpha_c.$$ 

Proof: We have $A_i = A_i \otimes f_i$ and $N_i = A_i \otimes g_i$. Therefore,

$$N_i = A_i \otimes g_i = A_i \otimes (f_i \otimes g_i).$$

Hence, if

$$\sum_{i=1}^{2} f_i \otimes g_i \leq \alpha_c.$$ 

all packets depart within their deadlines. In this case, $B_i \geq N_i = A_i \otimes g_i$ and the server is a $g_i$-server for flow $i$. 

Exercises

- Consider the network of Figure 1 with three arrival flows. Suppose $A_i$ is $\gamma_{b_i,r_i}$-constrained, for $i = 1, 2, 3$, that $r_1 + r_2 \leq c_1$, and that $r_1 + r_3 \leq c_2$. The first queue is FIFO, and the second gives priority to flow 3. For all three questions below, express your answers in terms of the given parameters, and also give numerical answers for the case $\gamma_{4,2}$, for $i = 1, 2, 3$, and $c_1 = c_2 = 5$.

1) Give the maximum delay $d_1$ for customers of the first queue.
2) Seek the smallest value of $b$ such that $B_1$ is $\gamma_{b,r_1}$-constrained.
3) Find an upper bound on the delay that flow 1 suffers in the second queue.

- Let

$$f_1(t) = 20 + t \quad \text{for } t > 0$$

$$f_2(t) = 8 + 4t \quad \text{for } t > 0$$

$$g_1(t) = \max\{0, 5(t - 4)\}$$

$$g_2(t) = \max\{0, 6(t - 20)\}$$

1) Sketch $f_i \otimes g_i$ for $i = 1, 2$. These functions are piecewise linear. Label the breakpoints and the values of the functions $f_i \otimes g_i$ at the breakpoints.
2) Suppose that $n_1$ flows, each $f_1$-constrained, and $n_2$ flows, each $f_2$-constrained share a link of capacity $c = 100$. We require that the link be an $g_1$-server for each of the $n_1$ flows and a $g_2$-server for each of the $n_2$ flows. Using SCED, it is possible to accommodate the flows if $n_1 f_1 \otimes g_1 + n_2 f_2 \otimes g_2 \leq \alpha_c$. Find and sketch the region of admissible $(n_1, n_2)$ pairs. [Hint: it is enough to check the inequalities at the breakpoints and at $t \to \infty$.]
Fig. 1. Example network