Guaranteed Rate Scheduling

ECE 1545

University of Toronto

(J. Liebeherr)

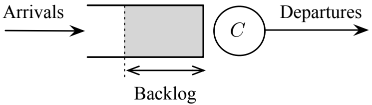
Road Map

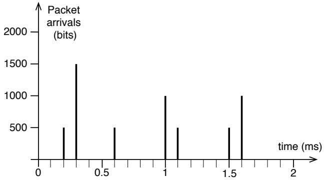
Goal: **Design a scheduler that gives rate guarantees to traffic**

1. VirtualClock Scheduling ($\rightarrow$ Powerpoint slides)
2. Service curves of the form $S(t) = Rt$
3. SCED scheduling with rate guarantees
4. Rate guarantees over long-term and short-term intervals
   $\Rightarrow$ lower vs. strict vs. adaptive service curves
5. Putting it all together
   $\Rightarrow$ VirtualClock, Packet Scale Rate Guarantees (PSRG)

Re-visit Work-conserving Link

C	Transmission rate
$T_A^p(n)$	Arrival time of n-th packet
$T_D^p(n)$	Departure time of n-th packet
ℓ_n	Size of n-th packet (in bits)
L_n	$= \sum_{j=1}^{n} \ell_n$





Recursion for Departure time of nth Packet

$$
\begin{aligned}
T_D^p(n) &= \max\Big\{T_A^p(n), T_D^p(n-1)\Big\} + \frac{\ell_n}{C} \\
&= \max\Big\{T_A^p(n) + \frac{\ell_n}{C}\ ,\ T_A^p(n-1) + \frac{\ell_{n-1} + \ell_n}{C}\ ,\ \ldots \\
&\qquad\qquad \ldots\ ,\ T_A^p(1) + \frac{\ell_1 + \ldots + \ell_n}{C}\Big\} \\
&= \max_{0 \le k \le n-1}\Big\{T_A^p(n-k) + \frac{\ell_{n-k} + \ldots + \ell_n}{C}\Big\}
\end{aligned}
$$

with $T_D^p(0) = 0$.

Bit-level View of Packets

We number the bits of the packets

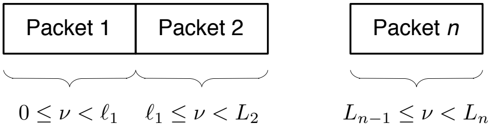
$$\underbrace{0, 1, \ldots, \ell_1 - 1}_{\text{Packet } 1}, \underbrace{\ell_1, \ldots, L_2 - 1}_{\text{Packet } 2}, \quad \cdots \quad , \underbrace{L_{n-1}, \ldots, L_n - 1}_{\text{Packet } n}$$

$$\boxed{T_D^p(n) = T_D(L_n - 1)}$$

$$\begin{aligned} T_D^p(n) &= \max\Big\{T_A(L_n - 1), T_D(L_n - 2)\Big\} + \frac{1}{C} \\ &= \max\Big\{T_A(L_n - 1) + \frac{1}{C}, T_A(L_n - 2) + \frac{2}{C}, \ldots \\ &\qquad\qquad\qquad \ldots, T_A(0) + \frac{L_n}{C}\Big\} \\ &= \max_{\kappa=0,1,\ldots,L_n-1}\Big\{T_A(L_n - 1 - \kappa) + \frac{\kappa + 1}{C}\Big\} \end{aligned}$$

Continuous-space View of Packets

Viewing bits as real numbers:



$$T_D^p(n) = T_D(L_n^-)$$

$$T_D^p(n) = T_D(L_n^-) = \sup_{0 \le \kappa \le L_n^-} \left\{ T_A(L_n^- - \kappa) + \frac{\kappa}{C} \right\}$$

$$= T_A \overline{\otimes} \gamma(L_n^-)$$

with $\gamma_S(\nu) = \frac{\nu}{C}$

Important relationship

We have shown:

$$T_D^p(n) = \max\Big\{T_A^p(n), T_D^p(n-1)\Big\} + \frac{\ell_n}{C}$$
$$= T_A \overline{\otimes}\, \gamma(L_n^-)$$

with $\gamma_S(\nu) = \frac{\nu}{C}$

Implication:

- For the departure time of packet n, the packet-level recursion is equal to the max-plus convolution.
- We can exploit this to design a SCED scheduler.

Max-Plus SCED

- Recall max-plus service curve: $T_D \leq T_A \overline{\otimes} \gamma_S$
- Max-plus SCED:
  Set deadlines to $D\ell(\nu) = T_A \overline{\otimes} \gamma_S(\nu)$
- Then:

$$T_D \leq T_A \overline{\otimes} \gamma_S \quad \Longleftrightarrow \quad D\ell \geq T_D$$

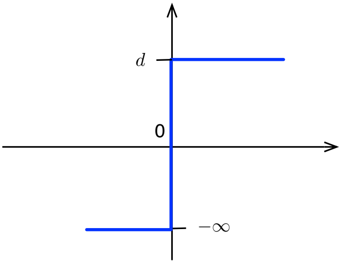
- Deadline of nth packet: $D\ell^p(n) = D\ell(L(n^-))$

Max-Plus SCED for Delay Guarantees

Delay service curve ($\nu \geq 0$):

$$\gamma_S(\nu) = \overline{\delta}_d(\nu) = T$$

Delay function

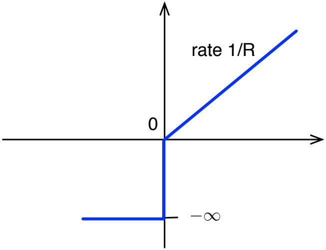


SCED implementation: $D\ell(\nu) = T_A \overline{\otimes} \gamma_S(\nu) = T_A(\nu) + T$

$\Longrightarrow$ This is EDF !

Max-Plus SCED for Rate Guarantees

Rate function



Rate service curve ($\nu \geq 0$):

$$\gamma_S(\nu) = \frac{\nu}{R}$$

SCED implementation:

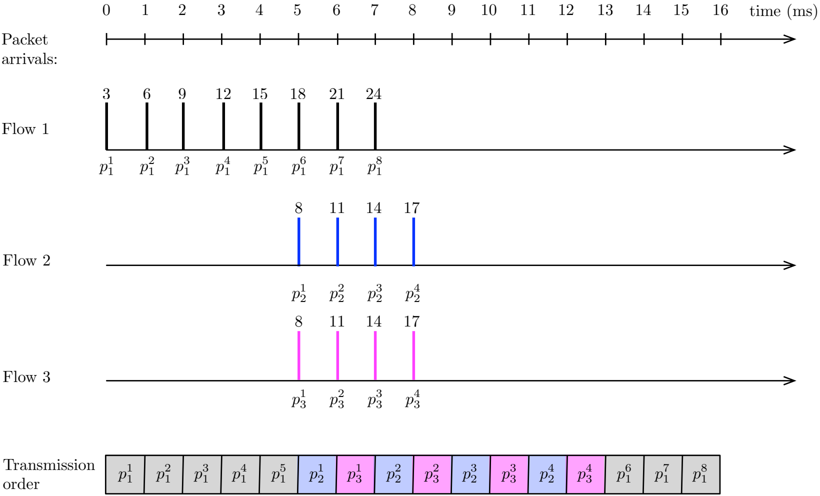
$$D\ell^p(n) = T_A \overline{\otimes} \gamma_S(L_n^-) = \max\left\{T_A^p(n), D\ell^p(n-1)\right\} + \frac{\ell_n}{C}$$

$\Longrightarrow$ This is VirtualClock (VC)

Note, however, that the deadline $D\ell^p(n)$ must hold for all $\nu \neq L_n^-$. To ensure this, we need to relax the service curve to $\gamma_S(\nu) = \frac{\nu + \ell_{\max}}{R}$

Scheduling with VirtualClock

$R_1 = R_2 = R_3 = \frac{1}{3}$ Mbps, $C = 1$ Mbps, $\ell = 1000$ bits



Drawback of VirtualClock

- VirtualClock implements a lower service curve, which gives guarantees for all $\nu > 0$ that

$$T_D(\nu) \geq T_A \overline{\otimes} \gamma_S(\nu)$$

- This is a guarantee for intervals $[0, \nu]$
- Drawback: Receiving more than the guarantee earlier, can result in less than the guarantee later.
- **Solution :** Need to look at different types of service curves.

Rate Guarantees

Consider the rate guarantee achieved with service curve $S(t) = Rt$:

- **exact service curve:** $D = A \otimes S$
  - good for expressing service at work-conserving buffered link
  - does not allow a flow to obtain more output than Rt in a time interval of length $t \ldots$

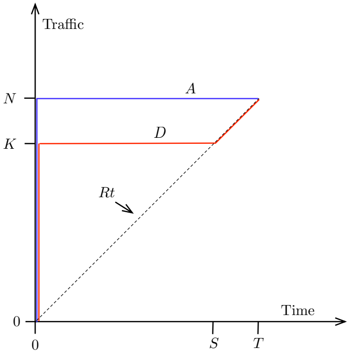
    $\ldots$ even if bandwidth is available
- **lower service curve:** $D \geq A \otimes S$
  - guarantees that the the service is **at least** as given by $D \geq A \otimes S$

    $\Longrightarrow$ can use more bandwidth if it is available

Lower service curve $S(t) = Rt$

Scenario: Large burst arrival,
a lot of service initially,
then no service in $(0, S]$

$A(t) = N = RT$,
for $0 \leq t \leq T$



In this scenario, for all $t \leq T$:

$$D(t) \geq A \otimes S(t) = Rt$$

$\Longrightarrow$ lower service curve $S(t) = Rt$ is always satisfied

What is the problem?

Lower service curves give guarantees over the time interval $[0, t)$. This is fine for delay guarantees. However, for rate guarantees, we want a rate guarantee for any time interval $(s, t]$, where there is a backlog

$\Longrightarrow$ **strict service curve**

A strict (min-plus) service curve S satisfies for all s, t such that there is a backlog in $(s, t]$ that

$$D(t) - D(s) \geq S(t - s)$$

$\Longrightarrow$ No penalty (at later time) when flow gets more service than guaranteed

Issue with strict service curves

- Strict service curves are great for rates ($S(t) = Rt$)
- They are not good for delay guarantees ($S(t) = \delta_T(t)$)

Consider flow that is backlogged in $[s, t]$:

- We must have $D(t) - D(s) \geq \delta_T(t - s)$
- If $s - t > T$, we get $D(t) - D(s) \geq \infty$
  $\implies$ busy period cannot be longer than T

.

What we want:

What we get:

Adaptive (Min-Plus) Service Curves

A network element offers an adaptive min-plus service curve S, if arrival and departure functions A and D satisfy for all $s \leq t$ that

$$D(t) \geq \min\left\{ D(s) + S(t-s), \inf_{\mathbf{s} \leq x \leq t} \{A(x) + S(t-x)\} \right\}$$

This can be re-written as a single condition:

$$D(t) \geq \sup_{s \leq t} \left[\min\left\{ D(s) + S(t-s), \inf_{\mathbf{s} \leq x \leq t} \{A(x) + S(t-x)\} \right\} \right]$$

Adaptive (Min-Plus) Service Curves

A network element offers an adaptive min-plus service curve S, if arrival and departure functions A and D satisfy for all $s \leq t$ that

$$D(t) \geq \min\left\{D(s) + S(t-s), \inf_{\mathbf{s} \leq x \leq t}\{A(x) + S(t-x)\}\right\}$$

Combines advantages of lower and strict service curves

- $S(t) = Rt$ guarantees

  $$D(t) - D(s) \geq R(t-s)$$

  whenever there is a backlog in $[s, t]$
- $S(t) = \delta_T(t)$ enforces

  $$D(t) \geq A(t-d)$$

  without limiting the lengths of busy periods

Adaptive Max-Plus Service Curves

Define:

$$F \overline{\underset{\mu}{\otimes}} G(\nu) = \sup_{\mu \leq \kappa \leq \nu} \{F(\kappa) + G(\nu - \kappa)\}$$

An adaptive max-plus service curve γ_S satisfies for all $\nu \geq 0$ and $\mu \leq \nu$:

$$T_D(\nu) \leq T_D(\mu) + \gamma_S(\nu - \mu) \quad \text{or} \quad T_D(\nu) \leq T_A \overline{\underset{\mu}{\otimes}} \gamma_S(\nu)$$

This can be re-written as a single condition:

$$T_D(\nu) \leq \inf_{\mu \leq \nu} \left\{ \max \left[T_D(\mu) + \gamma_S(\nu - \mu) \, , T_A \overline{\underset{\mu}{\otimes}} \gamma_S(\nu) \right] \right\}, \quad \forall \nu \geq 0$$

Adaptive Max-Plus SCED

- Set deadlines to

$$D\ell(\nu) = \inf_{\mu \le \nu} \left\{ \max \left[T_D(\mu) + \gamma_S(\nu - \mu) \, , T_A \overline{\underset{\mu}{\otimes}} \gamma_S(\nu) \right] \right\}$$

- Then:

$$T_D(\nu) \le \inf_{\mu \le \nu} \left\{ \max \left[T_D(\mu) + \gamma_S(\nu - \mu) \, , T_A \overline{\underset{\mu}{\otimes}} \gamma_S(\nu) \right] \right\}$$

$$\Longleftrightarrow$$

$$D\ell(\nu) \ge T_D(\nu)$$

- Deadline of nth packet: $D\ell^p(n) = D\ell(L(n^-))$

Adaptive Max-Plus SCED

We can show

$$T_D^p(n) \le \underbrace{\min_{1\le m\le n}\left\{\max\left[T_D^p(m-1)+\frac{1}{R}\sum_{j=m}^{n}\ell_j\,,\ \max_{m\le k\le n}\{T_A^p(k)+\frac{1}{R}\sum_{j=k}^{n}\ell_j\}\right]\right\}}_{F^p(n)=}$$

. . . and

$$T_D^p(n) \le F^p(n) = \max\left\{T_A^p(n), \min\left[T_D^p(n-1), F^p(n-1)\right]\right\} + \frac{\ell_n}{R}\,,$$

SCED implementation

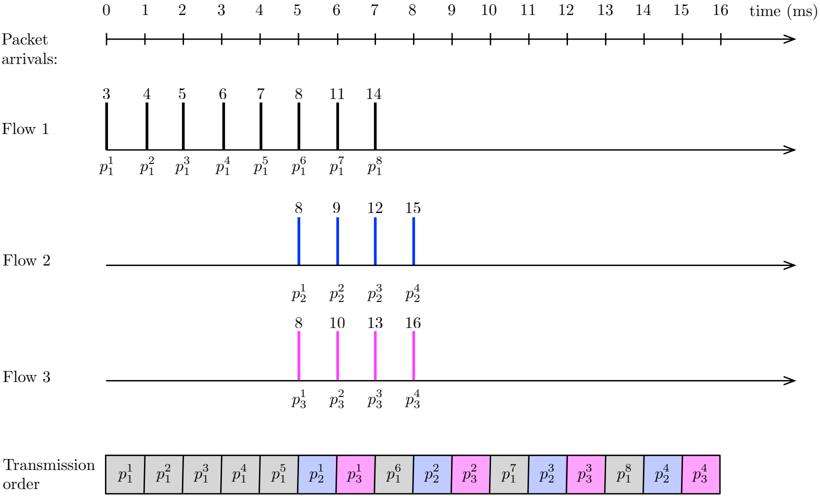
$$D\ell^p(k) = \max\left\{T_A^p(k), \min\{D\ell^p(k-1), T_D^p(k-1)\}\right\} + \frac{\ell_k}{R}$$

$\Longrightarrow$ This is Packet Scale Rate Guarantees (PSRG)

As with VirtualClock, we need to relax the service curve to $\gamma_S(\nu) = \frac{\nu+\ell_{\max}}{R}$

Scheduling with PSRG

$R_1 = R_2 = R_3 = \frac{1}{3}$ Mbps, $C = 1$ Mbps, $\ell = 1000$ bits



Issue with PSRG

- Deadline of nth packet requires knowledge of departure time of $(n-1)$th packet
- This means that deadlines cannot be computed at arrival time
- Deadline of a packet is computed is computed when previous packet from the same flow departs
- **Open Problem:** Schedulability condition for PSRG

Implementation of PSRG

PSRG for flow j with with guaranteed rate R and delay d:

- Counter $\texttt{VC}_j$ keeps track of $D\ell_j^p(k)$
- FIFO queue $\texttt{FIFO}_j$ for packets that are not assigned a deadline upon their arrival
- Each packet gets a timestamp (used as deadline)
- A transmission queue for packets with assigned deadlines (transmit in increasing order of deadlines)

Implementation of PSRG

If k-th packet from flow j with size ℓ arrives at time t:

- $t > T^p_{D_j}(k-1)$:
  - $\texttt{VC}_j \leftarrow t + \frac{\ell}{R}$
  - Timestamp packet: $\texttt{TS}^k_j = \texttt{VC}_j + d$
  - Add packet to transmission queue
- $t \leq T^p_{D_j}(k-1)$:
  - Timestamp packet: $\texttt{TS}^k_j = t$
  - Add packet to $\texttt{FIFO}_j$

If k-th packet from flow j departs at time t:

- $t > T^p_{A_j}(k+1)$:
  - $(k+1)$-th packet with length ℓ is at head of $\texttt{FIFO}_j$
  - $\texttt{VC}_j \leftarrow \max\{\texttt{TS}^{k+1}_j, \min\{\texttt{VC}_j, t\}\} + \frac{\ell}{R}$
  - $\texttt{TS}^{k+1}_j = \texttt{VC}_j + d$
  - Move $(k+1)$-th packet to transmission queue