Introduction to Network Calculus

John C.S. Lui

Department of Computer Science & Engineering
The Chinese University of Hong Kong

Outline

Introduction

What is network calculus?

- A theoretical framework to analyze performance guarantees (e.g., maximum delays, maximum buffer space requirements) in computer network.
- As traffic flows through a network, it is subject to constraints such as:
  - link capacity;
  - traffic shapers (e.g., *leaky buckets*);
  - congestion control;
  - background traffic.
- Express arrival, service and these constraints in a systematic manner (network calculus);
- key idea is to use the **min-plus algebra**.

Outline

Various Sections

- The basic (σ, ρ) constraints, performance bounds of single queue.
- General constraint of deterministic constraint.
- Application to service curves.
- Given the input traffic and service curves, how to derive maximum delay for any server that conforms to the service curve.
- Applications: (a) bounding the maximum delay of a priority queue; (b) scheduling service at a constant rate link with multiple input streams in order to achieve a specified service curve.

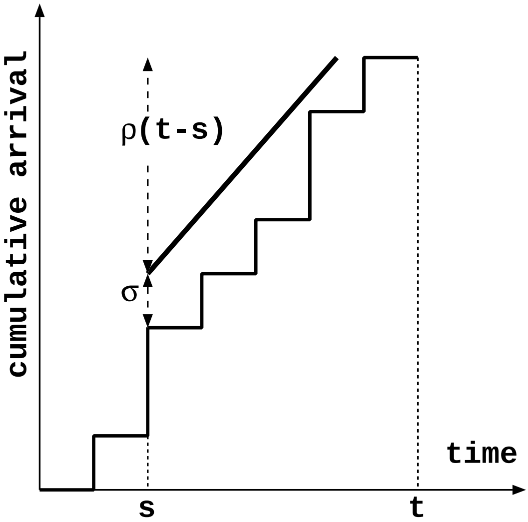
Upper Constrained Arrival Process

- For simplicity, assume equal length packets transmitted in discrete time.
- A *cumulative arrival process A* is a nondecreasing, integer-valued function on the nonnegative integer Z_+ such that $A(0) = 0$.
- $A(t)$ denotes the number of arrivals in slots $1, 2, \ldots, t$.
- $a(t)$ is the number of arrivals at time t and $a(t) = A(t) - A(t-1)$.
- We said A is *(σ, ρ)-upper constrained* (or $A \sim (\sigma, \rho)$) if

$$A(t) - A(s) \leq \sigma + \rho(t - s), 0 \leq s \leq t.$$

- In this lecture, σ, ρ are taken to be integer valued.

Example of A and (σ, ρ)



Token Bucket Filter

- Token bucket filter, a popular way to *regular* data streams and to generate $(\sigma, \rho)-$upper constrained traffic.

Definition

A token bucket filter (with no dropping) with (σ, ρ) operates like:

- The filter has infinite queue length and a token bucket.
- Events occur at integer time. New packets are added to the queue, and ρ new tokens are added to the token bucket.
- As many packets immediately depart if each packet has a token.
- If there are more than σ tokens in the bucket, drop some tokens until we have only σ tokens.

Observation

A token bucket filter with parameter (σ, ρ) is a (σ, ρ) regulator. Or for any input process A, the output process B is (σ, ρ)-upper constrained.

- Since at most σ tokens are in the bucket just before $s + 1$.
- And $\rho(t - s)$ tokens arrive in slots $s + 1, \ldots, t$.
- At most $\sigma + \rho(t - s)$ packets can depart from the filter in those slots.
- B is indeed upper constrained by (σ, ρ).

Multiplexing Rule

If constrained flows are merged, the output process is also constrained, or

$$A_i \sim (\sigma_i, \rho_i) \longrightarrow \sum A_i \sim \left(\sum \sigma_i, \sum \rho_i \right)$$

Performance bounds of constant server under $A \sim (\sigma, \rho)$

What are the performance bounds, i.e., duration of busy period, packet delay, for a constant server under $A \sim (\sigma, \rho)$?

- A single server with a constant service rate of C (positive integer).
- Let A be the cumulative arrival process.
- Let $q(t)$ be the queue length at time slot t. We have:

$$q(t+1) = (q(t) + a(t+1) - C)^+$$

with $q(0) = 0$.

Performance bounds: continue

- Using induction on t, we have

$$q(t) = \max_{0 \le s \le t} \{A(t) - A(s) - C(t-s)\} \qquad (1)$$

Show by induction: first, $q(0) = 0$.

$$q(1) = \max(0, q(0) + a(1) - C) = \max_{0 \le s \le 1} (A(1) - A(s) - C(1-s)).$$

Suppose it holds for t, it follows

$$\begin{aligned} q(t+1) &= \max\{0, \max_{0 \le s \le t} \{A(t) - A(s) - C(t-s)\} + a(t+1) - C\} \\ &= \max\{0, \max_{0 \le s \le t} \{A(t+1) - A(s) - C(t+1-s)\}\} \\ &= \max_{0 \le s \le t+1} \{A(t+1) - A(s) - C(t+1-s)\}. \end{aligned}$$

Performance bounds: continue:

- The output cumulative process B satisfies:

$$B(t) = A(t) - q(t) = \min_{0 \le s \le t} \{A(s) + C(t-s)\} \quad \forall t \ge 0.$$

- Queue Length Bound: Suppose A is (σ, ρ)-upper constrained, if $C \ge \rho$, Eq (1) implies $q(t) \le \sigma$ for all t. (*implication:* We obtain the bound, independent of the service order)
- Conversely, if $C = \rho$ and $q(t) \le \sigma$ for all t, then $A \sim (\sigma, \rho)$. (*implication:* if we can control $q(t)$, we specify the envelop of A)

Performance bounds: continue

We want to derive upper bound of

- **busy period**;
- **packet delay**

when $A \sim (\sigma, \rho)$.

Definition (Busy Period)

Given time s and t with $s \leq t$, a busy period is said to begin at s and end at t if $q(s-1) = 0, a(s) > 0, q(r) > 0$ for $s \leq r < t$ and $q(t) = 0$. The duration B of the busy period to be $B = t - s$ time units.

Performance bound: continue

Observation

Given such busy period, we must have

- C departures at each of the B times $\{s, \ldots, t-1\}$.
- At least one packet in the queue at time $t-1$.
- At least $CB+1$ packets must arrive at times $\{s, \ldots, t-1\}$ to *sustain* the busy period.

Since $A \sim (\sigma, \rho)$, we have at most $\sigma + \rho B$ packets arrive in B. We have $CB + 1 \leq \sigma + \rho B$, we have $B \leq \frac{\sigma - 1}{C - \rho}$. If B is an integer, it must be

$$B \leq \lfloor \frac{\sigma - 1}{C - \rho} \rfloor.$$

Performance bound: continue

Delay Bound

- The delay of a packet is the time the packet departs minus the time it arrives.
- The delay of *any* packet is less than or equal to the length of the busy period.

Thus, the upper of the packet delay d, *independent of service discipline*, is:

$$d \leq \lfloor \frac{\sigma - 1}{C - \rho} \rfloor.$$

If one unit of service time is added, we have

$$d + 1 \leq \lfloor \frac{\sigma - 1}{C - \rho} \rfloor \leq \lceil \frac{\sigma}{C - \rho} \rceil$$

Performance bound: continue

What if the service discipline is FIFO?

- If the packet has a nonzero waiting time, then it is carried over from the time it first arrived to the next time slot.
- The total number of packets carried over, including this packet, is less than or equal to σ (shown after Eq. (1)).
- The delay of FIFO is:

$$d_{FIFO} = \lceil \frac{\sigma}{C} \rceil.$$

- If service time is included, we have to add 1 to the above expression.

Output Analysis 1

If $A \sim (\sigma, \rho)$ and delay bound d, what about the output B?

- Let say we know the maximum delay of the queue is d.
- For $s < t$, any packets that departs from the queue at a time in $\{s+1, \ldots, t\}$ must arrive at one of the $t - s + d$ times in $\{s + 1 - d, \ldots, t\}$.
- Therefore, output process based on delay bound d is

$$B(t) - B(s) \leq A(t) - A(s - d) \leq \sigma + \rho d + \rho(t - s).$$

- Therefore, $B \sim (\sigma + \rho d, \rho)$-upper constrained.

Output Analysis 2

If $A \sim (\sigma, \rho)$ and queue length bound q, what about the output B?

- Let say we know the maximum queue length is q.
- Let $q(t)$ be the queue length at time t, we have

$$\begin{aligned} B(t) - B(s) &= A(t) - A(s) - (q(t) - q(s)) \\ &\leq A(t) - A(s) + q(s) \leq \sigma + \rho(t-s) + q \\ &\leq \sigma + q + \rho(t-s) \end{aligned}$$

- Therefore, output process based on queue length bound q is $B \sim (\sigma + q, \rho)$-upper constrained.
- Now we **characterized** B, B is fed into another queue and we can continue to do the delay analysis.

Output analysis 3

If $A \sim (\sigma, \rho)$ and the server is work conserving, what about the output B?

- Assume it is a work conserving link with capacity C, we have

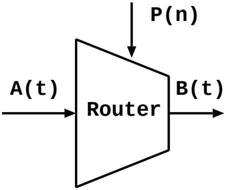
$$B(s) = \min_{0 \le \tau \le s} [A(\tau) + C(s - \tau)] \quad ; \quad B(t) = \min_{0 \le \tau \le t} [A(\tau) + C(t - \tau)]$$

- Let τ^* be the argument which achieves the minimum in $B(s)$. We have $B(s) = A(\tau^*) + C(s - \tau^*)$ and $B(t) \le A(t - s + \tau^*) + C(s - \tau^*)$ (by choosing $\tau = t - s + \tau^*$). We have

$$\begin{aligned} B(t) - B(s) &\le A(t - s + \tau^*) + C(s - \tau^*) - A(\tau^*) - C(s - \tau^*) \\ &= A(t - s + \tau^*) - A(\tau^*) \le \sigma + \rho(t - s) \end{aligned}$$

- Output process based on work conservation, B is $(\sigma, \rho)-$upper constrained

Routing



Definition

An ideal router is a network element with one input A, one control input P, one output B such that $B = P(A(t))$ where $A(t)$ is the cumulative number of arrival by time t, $P(n)$ is the number of arrivals that are selected among the first n arrivals, and $B(t)$ as the cumulative departures by time t. In other words, the cumulative number of output by time t is the cumulative number of arrivals selected by time t.

Characterization of router's output

Lemma

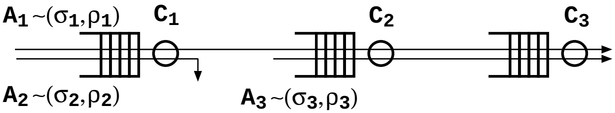
For an ideal router, if $A \sim (\sigma, \rho)-$*upper constrained and* $P \sim (\delta, \gamma)-$*upper constrained, then* $B \sim (\gamma\sigma + \delta, \gamma\rho)-$ *upper constrained.*

Proof

Observe that

$$\begin{aligned} B(t) - B(s) &= P(A(t)) - P(A(s)) \\ &\leq \delta + \gamma(A(t) - A(s)) \\ &\leq \delta + \gamma(\sigma + \rho(t - s)) \\ &= \delta + \gamma\sigma + \gamma\rho(t - s) \end{aligned}$$

Application: feed-forward network



Parameters

- $C_1 = C_2 = C_3 = 4$
- Arrival processes are $(\sigma_k, \rho_k)-$upper constrained with $(\sigma_1, \rho_1) = (1, 2)$, $(\sigma_2, \rho_2) = (2, 1)$, $(\sigma_3, \rho_3) = (3, 2)$.
- Routing, as indicated in the figure.

Analysis on the 1st communication link

- Based on the multiplexing rule, the overall arrival to link 1 is $(\sigma_1 + \sigma_2, \rho_1 + \rho_2)-$upper constrained, or $(3, 3)$.
- Since $\rho_1 + \rho_2 = 3 < C_1 = 4$, using the delay bound result after Eq. (1), the maximum queue length q_1 in the first link is upper bounded by $q_1 = \sigma_1 + \sigma_2 = 3$, and using the delay bound result, we have $d_1 = \lceil (\sigma_1 + \sigma_2)/(C_1 - \rho_1 - \rho_2) \rceil = 3$.
- Let B_1 be the output process. Since A_2 will not affect the second link, we only need to consider A_1. Using the *bounding output process based on queue length*, we have $B_1 \sim (\sigma_1 + q_1, \rho_1) = (4, 2)-$upper constrained.

Analysis on the 2nd communication link

- Based on the multiplexing rule, since $B_1 \sim (4, 2)$ and $A_3 \sim (3, 2)$, we have $A \sim (7, 4)-$upper constrained.
- Because 4 is equal to C_2, the maximum queue length $q_2 = 7$.
- Since $C_2 = 4$, we cannot use the delay bound result (since $C - \rho = 0$ in the denominator). If this link uses FCFS discipline, we have $d_2 = \lceil 7/4 \rceil = 2$.
- The output process B_2, based on *bounding output process based on work conserving link* is $B_2 \sim (7, 4)-$upper constrained.

Analysis on the 3rd communication link

- Arrival process to this link is same as B_2, therefore $(7,4)-$upper constrained.
- Based on the established theory, bound on $q_3 = \sigma = 7$.
- Note that this bound is too loose. Why?
- Since $C_2 = C_3 = 4$, it means at most 4 packets come out from the 2nd link and these packets will be *immediately* served at the 3rd link.

Interesting questions

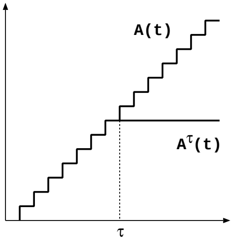
- Can we refine the theory to obtain tighter bound?
- What about communication systems with feedback?
- What about multi-class communication networks?

Single class non-feed-forward routing

Definition

For any increasing sequence A, we define its "stopped sequence" at time τ, denoted as A^τ, by

$$A^\tau(t) = \begin{cases} A(t) & \text{if } t \leq \tau, \\ A(\tau) & \text{otherwise} \end{cases} \quad (2)$$



comment

If A is an arrival process, then there are no further arrivals after time τ for the stopped sequence A^τ.

Traffic characterization of A^τ

Lemma

For every ρ, a stopped sequence A^τ is $(\sigma(\tau), \rho)-$upper constrained where

$$\sigma(\tau) = \max_{0 \le t \le \tau} \max_{0 \le s \le t} \left[A(t) - A(s) - \rho(t - s)\right]. \quad (3)$$

Proof

As the sequence A^τ is stopped at time τ, $\sigma(\tau)$ is the maximum queue length of a work conserving link with capacity ρ and input A^τ.

Traffic characterization of A^τ: continue

Corollary

If A^τ is $(\sigma, \rho)-$upper constrained, then $\sigma(\tau) \leq \sigma$, where $\sigma(\tau)$ is defined in Eq. (3).

Proof

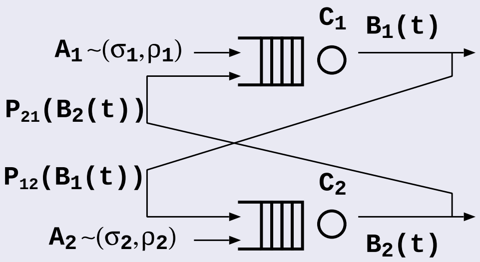
If A^τ is $(\sigma, \rho)-$upper constrained, then for all $0 \leq s \leq t \leq \tau$,

$$A(t) - A(s) = A^\tau(t) - A^\tau(s) \leq \sigma + \rho(t - s).$$

That $\sigma(\tau) \leq \sigma$ follows immediately from Eq.(3).

Example of feedback queues

- Consider the following network:



- $A_1 \sim (\sigma_1, \rho_1)$, $A_2 \sim (\sigma_2, \rho_2)$, $P_{12} \sim (\delta_{12}, \gamma_{12})$, $P_{21} \sim (\delta_{21}, \gamma_{21})$.
- What is the performance of the system? Say the queue length bound?

Analysis

- Let $\tilde{A}_1$ ($\tilde{A}_2$) be the overall arrival process of the first (second) link and B_1 (B_2) be the respective output process. We have

$$\tilde{A}_1(t) = A_1(t) + P_{21}(B_2(t)), \quad (4)$$
$$\tilde{A}_2(t) = A_2(t) + P_{12}(B_1(t)). \quad (5)$$

- The main idea to for the analysis is to *derive* the performance bounds for a finite time τ and show the bounds are *independent* of τ.
- Let B_1^τ (B_2^τ) be the stopped sequence of B_1 (B_2) at time τ.
- It follows that for "*any*" α_1, $B_1^\tau \sim (\sigma_1(\tau), \alpha_1)$.

$$\sigma_1(\tau) = \max_{0 \le t \le \tau} \max_{0 \le s \le t} [B_1(t) - B_1(s) - \alpha_1(t-s)].$$

- Similarly, for "*any*" α_2, $B_2^\tau \sim (\sigma_2(\tau), \alpha_2)$.

$$\sigma_2(\tau) = \max_{0 \le t \le \tau} \max_{0 \le s \le t} [B_2(t) - B_2(s) - \alpha_2(t-s)].$$

Analysis: continue

- Solve α_1 and α_2: $\alpha_1 = \rho_1 + \gamma_{21}\alpha_2; \qquad \alpha_2 = \rho_2 + \gamma_{12}\alpha_1$.
- Assume $\gamma_{12}\gamma_{21} < 1$, we have:

$$\alpha_1 = (1 - \gamma_{12}\gamma_{21})^{-1}(\rho_1 + \gamma_{21}\rho_2); \quad \alpha_2 = (1 - \gamma_{12}\gamma_{21})^{-1}(\rho_2 + \gamma_{12}\rho_1).$$

- Using the routing and multiplexing rules we discussed:

$$\begin{aligned} \tilde{A}_1 &\sim (\sigma_1 + \gamma_{21}\sigma_2(\tau) + \delta_{21}, \rho_1 + \gamma_{21}\alpha_2), \\ B_1^\tau &\sim (\sigma_1 + \gamma_{21}\sigma_2(\tau) + \delta_{21}, \rho_1 + \gamma_{21}\alpha_2) \end{aligned}$$

- Since we solved α_1, α_2, we can say that B_1^τ is $(\sigma_1 + \gamma_{21}\sigma_2(\tau) + \delta_{21}, \alpha_1)-$upper constrained. It follows that

$$\sigma_1(\tau) \leq \sigma_1 + \gamma_{21}\sigma_2(\tau) + \delta_{21}.$$

- Using similar argument, we can characterize B_2^τ:

$$\sigma_2(\tau) \leq \sigma_2 + \gamma_{12}\sigma_1(\tau) + \delta_{12}.$$

Analysis: continue

- Solving the above equations results in $\sigma_1(\tau) \leq \tilde{\sigma}_1$ and $\sigma_2(\tau) \leq \tilde{\sigma}_2$ where

$$\begin{aligned}\tilde{\sigma}_1 &= (1-\gamma_{12}\gamma_{21})^{-1}(\sigma_1+\gamma_{21}\sigma_2+\gamma_{21}\delta_{12}+\delta_{21}),\\ \tilde{\sigma}_2 &= (1-\gamma_{12}\gamma_{21})^{-1}(\sigma_2+\gamma_{12}\sigma_1+\gamma_{12}\delta_{21}+\delta_{12}).\end{aligned}$$

  These bounds are independent of τ !! So B_1 is $(\tilde{\sigma}_1, \alpha_1)-$upper constrained and B_2 is $(\tilde{\sigma}_2, \alpha_2)-$upper constrained.
- This in turn implies that $\tilde{A}_1$ is $(\sigma_1 + \gamma_{21}\tilde{\sigma}_2 + \delta_{21}, \alpha_1)-$ upper constrained and $\tilde{A}_2$ is $(\sigma_2 + \gamma_{12}\tilde{\sigma}_1 + \delta_{12}, \alpha_2)-$ upper constrained.
- Queue length in server 1 is bounded by $\sigma_1 + \gamma_{21}\tilde{\sigma}_2 + \delta_{21}$ if $\sigma_1 = (1-\gamma_{12}\gamma_{21})^{-1}(\rho_1+\gamma_{21}\rho_2) \leq C_1$,
- Queue length in server 2 is bounded by $\sigma_2 + \gamma_{12}\tilde{\sigma}_1 + \delta_{12}$ if $\sigma_2 = (1-\gamma_{12}\gamma_{21})^{-1}(\rho_2+\gamma_{12}\rho_1) \leq C_2$.

Generalization of (σ, ρ)

$f-$upper constraint processes

- Let f be a nondecreasing function from Z_+ to Z_+.
- An arrival process A is $f-$upper constrained if

$$A(t) - A(s) \leq f(t-s) \quad \text{for all } s, t \text{ with } 0 \leq s \leq t$$

- Rearranging, A is $f-$upper constrained iff $A(t) \leq A(s) + f(t-s)$ for $0 \leq s \leq t$, or $A \leq A \star f$, where $f \star g$ is the function on Z_+ defined as

$$(f \star g) = \min_{0 \leq s \leq t} g(s) + f(t-s). \qquad (6)$$

Similar to *"convolution"*, the min-plus algebra uses min instead of integration, $+$ instead of multiplication. **ILLUSTRATE**!

Comments on f^*

Some Comments

- Some functions can be reduced without changing the condition that an arrival process is $f-$upper constrained, e.g., $f(0) = 0$ because $A(t) - A(t) = 0$ anyway.
- Suppose A is $f-$upper constrained and $s, u \geq 0$, then $A(s+u) - A(s) \leq f(u)$ but a tighter bound may be implied. Let $n \geq 1$ and u is represented as $u = u_1 + \cdots + u_n$ where $u_i \geq 1$ and integer, then

$$
\begin{aligned}
A(s+u) - A(s) &= (A(s+u_1) - A(s)) + (A(s+u_1+u_2) - A(s+u_1)) \\
&\quad + \cdots + (A(s+u_1+\cdots+u_n) - A(s+u_1+\cdots+u_{n-1})) \\
&\leq f(u_1) + f(u_2) + \cdots + f(u_n).
\end{aligned}
$$

Sub-additive Closure

So $A(t) - A(s) \leq f^*(t-s)$, where f^* is the *sub-additive closure of f*, is defined by

$$f^*(u) = \begin{cases} 0 & \text{if } u = 0 \\ \min\{f(u_1) + \cdots + f(u_n) : n \geq 1, u_i \geq 1, \sum_i u_i = u\} & \text{if } u \geq 1. \end{cases}$$

+roperties on f^*

Properties

1. $f^* \leq f$
2. A is $f-$upper constrained iff A is f^*-upper constrained
3. f^* is sub-additive, $f^*(s+t) \leq f^*(s) + f^*(t)$ for all $s, t \geq 0$
4. If g is any other function with $g(0) = 0$ satisfying (1) and (3), then $g \leq f^*$

Illustrate

Maximal regulator for f

Definition (Regulator for f)

A regulator for f is a service center such that for any input A, the corresponding output B is $f-$upper constrained.

Definition (Maximal Regulator for f)

A regulator is said to be a maximal regulator for f if the following is true. For any input A, if B is the output of the regulator for input A and if $\tilde{B}$ is a cumulative arrival process such that $\tilde{B} \leq A$ and $\tilde{B}$ is $f-$upper constrained, then $\tilde{B} \leq B$.

Finding the maximal regulator for f

Theorem

A maximal regulator for f is determined by $B = A \star f^*$.

Proof

Let A, B, and $\tilde{B}$ be as in the definition of the maximal regulator, then

$$\tilde{B} = \tilde{B} \star f^* \leq A \star f^* = B \tag{7}$$

- First equality holds because $\tilde{B}$ is $f-$upper constrained.
- Inequality holds because $\star f^*$ is a monotone operation.
- The final equality holds by the definition of B.

Continue

Corollary

Suppose f_1 and f_2 are nondecreasing functions on Z_+ with $f_1(0) = f_2(0) = 0$. Two maximal regulators in series, consisting of a maximal regulator of f_1 followed by a maximal regulator for f_2, is a maximal regulator for $f_1 \star f_2$. In particular, the output is the same if the order of the two regulators is reversed.

Proof

Let A be in the input to the first regulator and B is the output of the second regulator, then

$$B = (A \star f_1^*) \star f_2^* = A \star (f_1^* \star f_2^*) = A \star (f_1 \star f_2)^*. \quad (8)$$

The last equality depends on the assumption $f_1(0) = f_2(0) = 0$.
The second part of the theorem is by the uniqueness of maximal regulators and the fact $f_1 \star f_2 = f_2 \star f_1$.

Maximal regulator for token bucket filter

Theorem

The token bucket filter with parameter (σ, ρ) *is the maximal* (σ, ρ) *regulator.*

Introduction

What we have learnt

- So far, we have considered passing a constrained process into a maximal regulator.
- Examples of maximal regulators are token bucket, queue with fixed service rate.
- Output of these servers is completely determined by the input.
- In practice, there can be many other types of servers, and server may have priority in selecting with traffic to serve first.
- What we need is a *flexible way* to specify a *guarantee* that a particular server offers.

Service curve

Most service centers are not fixed rate server or token bucket filter, we need a flexible way to specify a service behavior.

Definition (Service Curve)

A service curve is a nondecreasing function from Z_+ to Z_+. Given a service curve f, a server is an $f-$server if for any input A, the output B satisfies $B \geq A \star f$. That is:

$$B(t) \geq \min_{0 \leq s \leq t} \{A(s) + f(t-s)\}$$

comment on service curves under study

Comments

- For "*regulator*", we can completely specify the output process B. For "*service curve*", we get the "*inequality*" only.
- We only consider servers such that $B(t) \leq A(t)$ (or causality), and it is assumed that $A(0) = 0$. Take $s = t$ in the above equation implies $B(t) \geq A(t) + f(0)$ and this can only happen when $f(0) = 0$.

Examples

Different servers

- Given an integer $d \geq 0$, define

$$O_d(t) = \begin{cases} 0 & \text{for } t \leq d, \\ +\infty & \text{for } t > d. \end{cases}$$

  Then a FIFO device is an O_d-server iff the delay for every packets is less than or equal to d, independent of A.
- A server with constant service rate C is an f server for $f(t) = Ct$.
- A leaky bucket regulator is an f server for $f(t) = (\sigma + \rho t)I_{\{t \geq 1\}}$.
- The maximal regulator for a function f is an f^*-server.

Some definitions

Suppose an f_1-upper constrained process A passes through an f_2-server. We define:

Definition (d_V)

$$d_V = \max_{t\geq 0}\left(f_1^*(t) - f_2(t)\right),$$

or d_V is the maximum vertical distance that the graph f_1^* is above f_2.

Definition (d_H)

$$d_H = \max\left\{d \geq 0 : f_1(t) \leq f_2(t+d)\text{for all } t \geq 0\right\},$$

or d_H is the maximum horizontal distance that the graph d_2 is to the right of f_1^*.

Characterization

Theorem

Let A be f_1-upper constrained passing through an f_2-server with output process B. The queue size $A(t) - B(t)$ is less than or equal to d_V for any $t \geq 0$, and if the order of service is FIFO, the delay of any packet is less than or equal to d_H.

Proof

- Let $t \geq 0$. Since it is an f_2-server, there exists an s^* with $0 \leq s^* \leq t$ such that $B(t) \geq A(t-s^*) + f_s(s^*)$. Because A is f_1-upper constrained, $A(t) \leq A(t-s^*) + f_1^*(s^*)$. Thus, $A(t) - B(t) \leq f_1^*(s^*) - f_2(s^*) \leq d_v$.
- Suppose a packet arrives at time t and departs at time $\bar{t} > t$. Then $A(t) > B(\bar{t}-1)$. Since the server is f_2-server, there exists an s^* with $0 \leq s^* \leq \bar{t}-1$ such that $B(\bar{t}-1) \geq A(s^*) + f_2(\bar{t}-1-s^*)$. Because $A(t) > B(\bar{t}-1)$, it must be that $0 \leq s^* \leq t$. Since A is f_1-upper constrained, we have:

$$A(s^*) + f_1(t-s^*) \geq A(t) \geq B(\bar{t}-1) \geq A(s^*) + f_2(\bar{t}-1-s^*).$$

so that $f_1(t-s^*) > f_2(\bar{t}-1-s^*)$. Hence $\bar{t}-1-t < d_H$, so $\bar{t}-t < d_H$.

Example

Consider a server of constant rate C which servers input streams 1 and 2, giving priority to packets from input 1, and serves the packets within a single input stream in FIFO order. Let A_i be the cumulative arrival stream of input i.

Claim

If A_1 is f_1-upper constrained, the link is an $\tilde{f}_2-$server for type 2 stream, where

$$\tilde{f}_2(t) = (Ct - f_1(t))^+ .$$

Example

- Suppose A_i is (σ_i, ρ_i) constrained for each i. Then $\tilde{f}(t) = ((C - \rho_1)t - \sigma_1)^+$. Applying previous result to yield the delay for any packet in input 2 is less than or equal to $(\sigma_1 + \sigma_2)/(C - \rho_1)$. Since packets in input 1 are not affected by the packets from input 2, the delay for any packet in input 1 is less than or equal to $\lceil \sigma_1/C \rceil$.
- If queue were served in FIFO order, then the maximum delay for packets from either input is $\lceil \frac{\sigma_1+\sigma_2}{C} \rceil$. If σ_1 is much smaller than σ_2, the delay for input 1 packets is much smaller for the priority server than for the FIFO server.

Service Curve Earliest Deadline (SCED)

Suppose there are multiple input streams on the constant rate link to meet specified service curves for each input. Let the i^{th} input has a cumulative arrival process A_i which is known to be g_i-upper constrained and supposed that the i^{th} input wants to receive service conforming to a specified service curve f_i. The question is, what is the algorithm to achieve the above goal?

Service Curve Earliest Deadline (SCED)

SCED

- Under SCED, for each input stream i let $N_i(t) = A_i(t) \star f_i$. Then $N_i(t)$ is the minimum number of type i packets that must depart by time t in order that the input i see service curve f_i. Based on this, the deadline can be computed for each packet of input i. Specifically, the deadline for the k^{th} packet from input i is the minimum t such that $N_i(t) \geq k$. In other words, if all packets are scheduled by their deadlines, then all service curve constraints are met.
- Given any arrival sequence with deadlines, if it is possible for an algorithm to meet all the deadlines, then the **earliest deadline first (EDF)** scheduling can do it. The SCED is to put deadlines on the packets and use EDF.

SCED Scheduling

Theorem

Given g_i, f_i for each i and capacity C satisfying

$$\sum_{i=1}^{n} (g_i \star f_i)(t) \leq Ct \quad \forall t,$$

the service to each input stream i provided by the SCED scheduling algorithm conforms to service curve f_i.

Proof

- Fix a time $t_0 \geq 1$. Let say all packets with deadline t_0 or earlier are colored red, while all other packets are colored white. For any time $t \geq 0$, let $q_0(t)$ denote the number of red packets that are not scheduled by t_0. Since EDF is used, red packets have pure priority over the white packets, so q_0 is the queue length process in case all white packets are all ignored. We need to show $q_0(t_0) = 0$.
- For $0 \leq s \leq t_0 - 1$, the number of red packets that arrive from stream i in the set of times $\{s+1, \cdots, t_0\}$ is $(N_i(t_0) - A_i(s))^+$. Also, $N_i(t_0) \leq A_i(t_0)$. Therefore

$$q_0(t_0) = \max_{0 \leq s \leq t_0} \left\{ \sum_i (N_i(t_0) - A_i(s))^+ \right\} - C(t_0 - s).$$

Continue:

Proof: continue

- For any s with $1 \leq s \leq t_0$,

$$\begin{aligned} N_i(t_0) &= (A_i \star f_i)(t_0) \leq (A_i \star g_i \star f_i)(t_0) \\ &= \min_{0 \leq u \leq t_0} A_i(u) + (g_i \star f_i)(t_0 - u) \leq A_i(s) + (g_i \star f_i)(t_0 - s) \end{aligned}$$

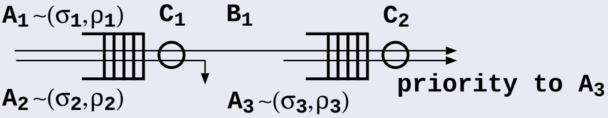
- It follows that

$$\sum_i (N_i(t_0) - A_i(s))^+ \leq \sum_i (g_i \star f_i)(t_0 - s) \leq C(t_0 - s).$$

So $q_0(t_0) = 0$ and the proposition is proved.

Example

- Consider the following network:



- We have $\rho_1 + \rho_2 \leq C_1$ and $\rho_1 + \rho_3 \leq C_2$. The first server is FIFO server. The second server gives priority to A_3 but is FIFO in within each class. $(\sigma_i, \rho_i) = (4, 2)$ for $1 \leq i \leq 3$ and $C_1 = C_2 = 5$.
- (a) What is the maximum delay d_1 ? (b) Characterize B_1. (c) What is the maximum delay of stream 1 in the second queue?

Solution:

- The total arrival stream to the first queue is $(\sigma_1 + \sigma_2, \rho_1 + \rho_2)-$upper constrained. Since the server is FIFO, $d_1 = \lceil (\sigma_1 + \sigma_2)/C_1 \rceil = 2$.
- $B \sim (\sigma_1 + \rho_1 d_1, \rho_1)-$upper constrained.
- Let d_1^2, be the delay of the lower priority stream for a FIFO server is:

$$d_1^2 \leq \frac{(\sigma_1 + \rho_1 d_1) + \sigma_3}{C_3 - \rho_3} = \frac{12}{3} = 4.$$