Worst Case Analysis - Network Calculus

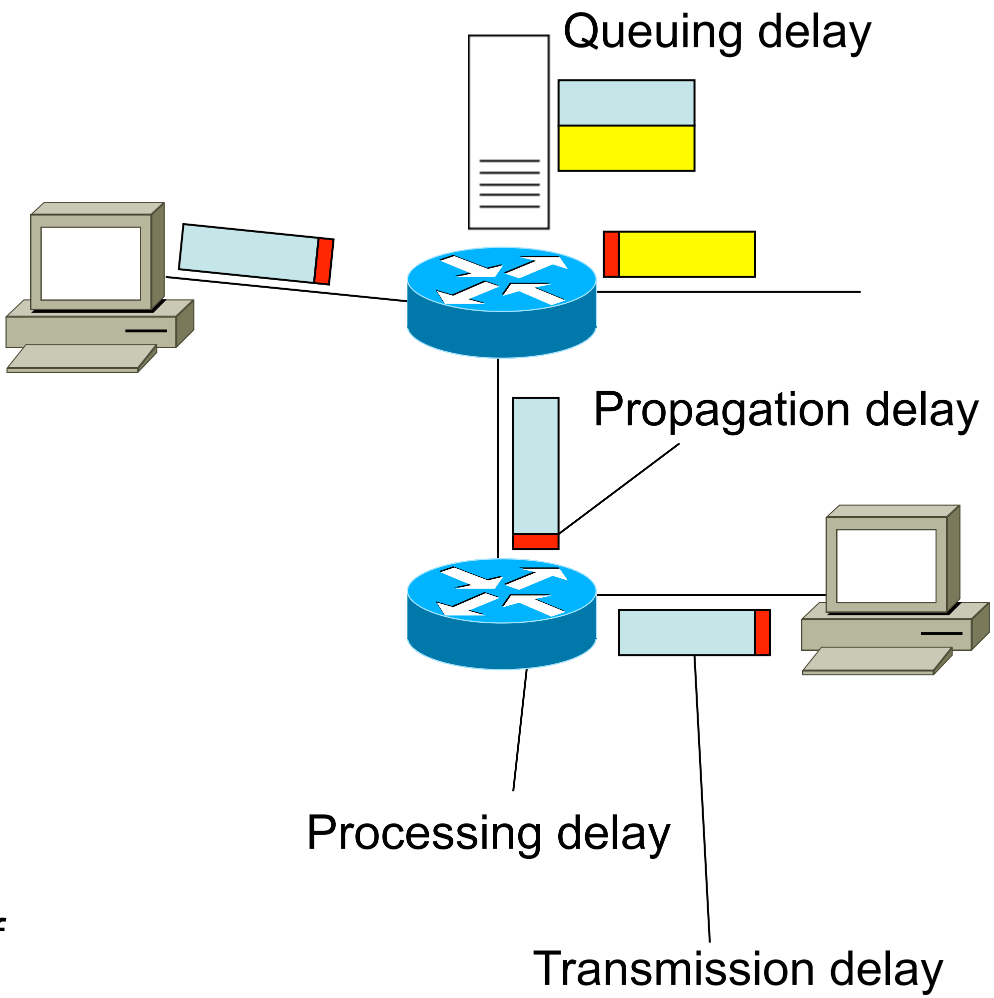
- Network Calculus Primer
- Networks in Network Calculus
  - Packet Based Networks
  - Tandem Networks
  - Feed Forward Networks
  - Non Feed Forward Networks
- Traffic Models in Network Calculus
- Formalisms in the Network Calculus
  - Min-plus Algebra
  - Arrival and Service Curves
  - Latency and Backlog Bounds
- Tightening Bounds
  - Convolution Form Network /
    Pay Bursts Only Once
  - Pay Multiplexing Only Once
- Excursion to Network Analysis Tools

Network Calculus Primer

- Latencies in Networks
- Sample Delay Calculation
- Origins of Network Calculus

Latencies in Networks

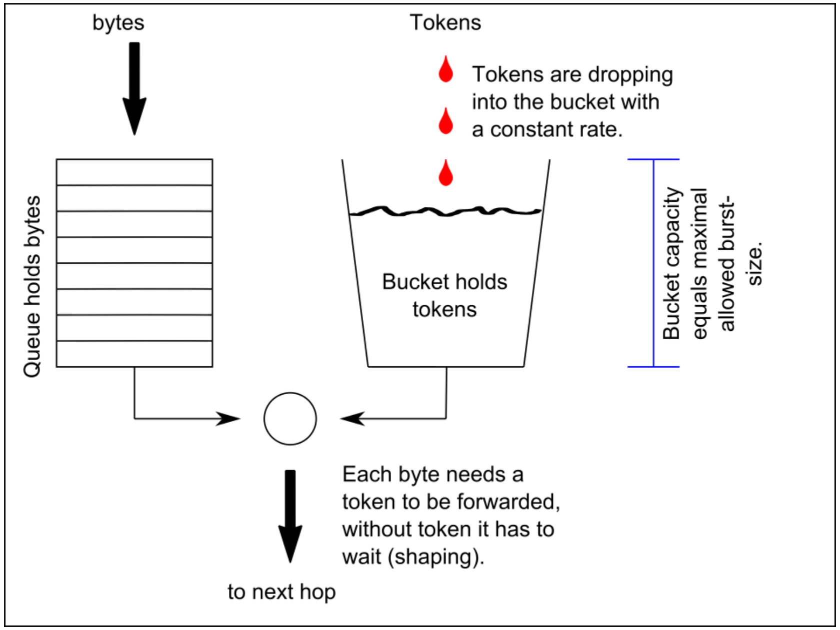
- Propagation Delay
  - stable and almost negligible
  - speed of light
- Processing Delay
  - Hardware dependent
  - relatively stable
- Transmission Delay
  - Time it takes to transmit the whole frame
- Queuing Delay
  - If output port is busy, frames must be queued
  - Sum of transmission delay of other frames, that have to be served before



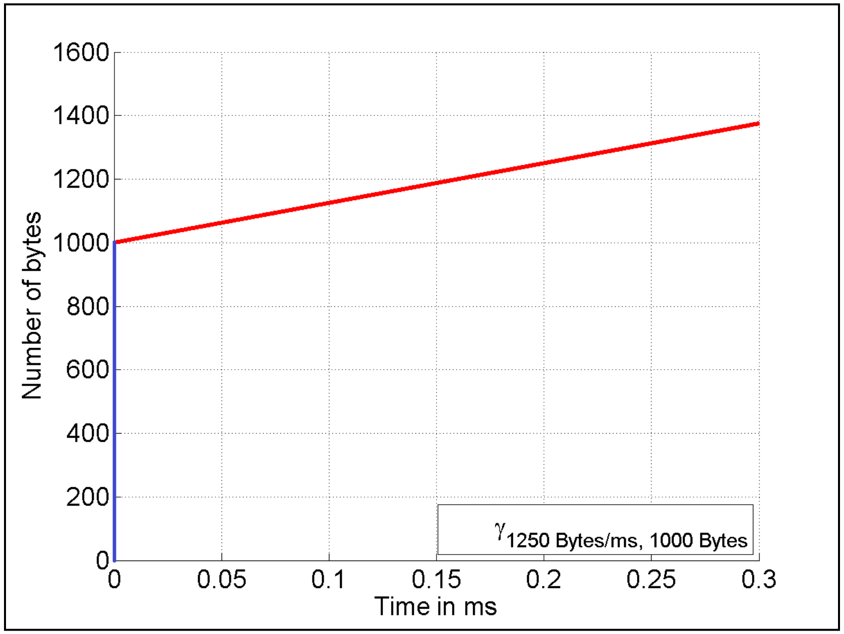
Refreshing Token Bucket Model

- Extended version of [Tan2002] and [Sta2001] which allows some burstiness
- Shaping does not occur until burst is consumed

Token Bucket Scheme



Network Calculus Representation (accumulated arrivals)



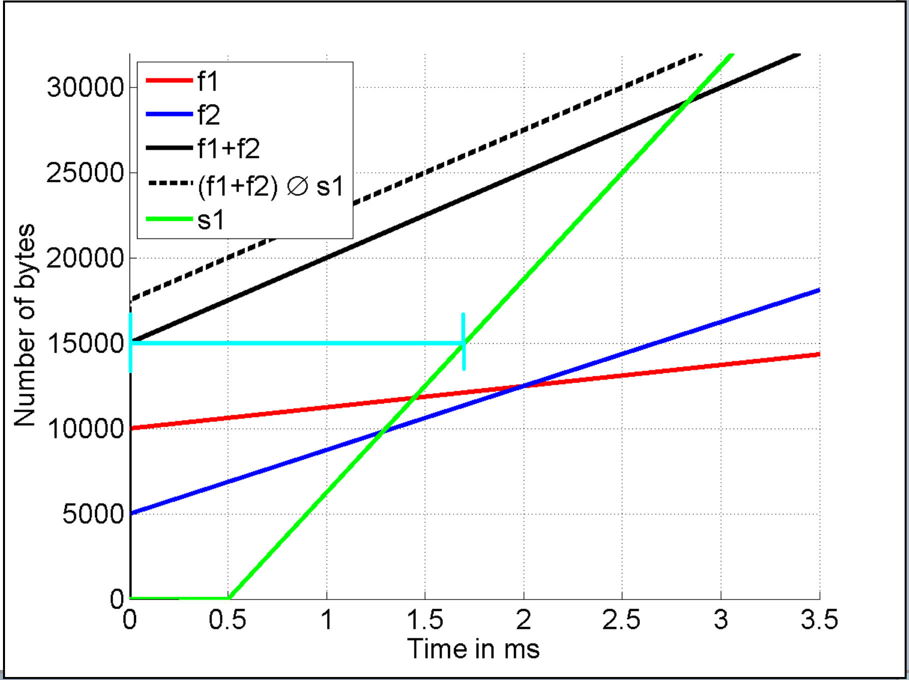
Short Introduction to Network Calculus (I)

- Flows in terms of Arrival Envelopes / Arrival Curves
- Service experienced by switch in terms of Service Curve

Example of fluid flows, preemptive

- f1 and f2 are multiplexed and traverse two servers / switches
- Flow of interest is f1
- Delay given by horizontal deviation

Node-by-Node Analysis



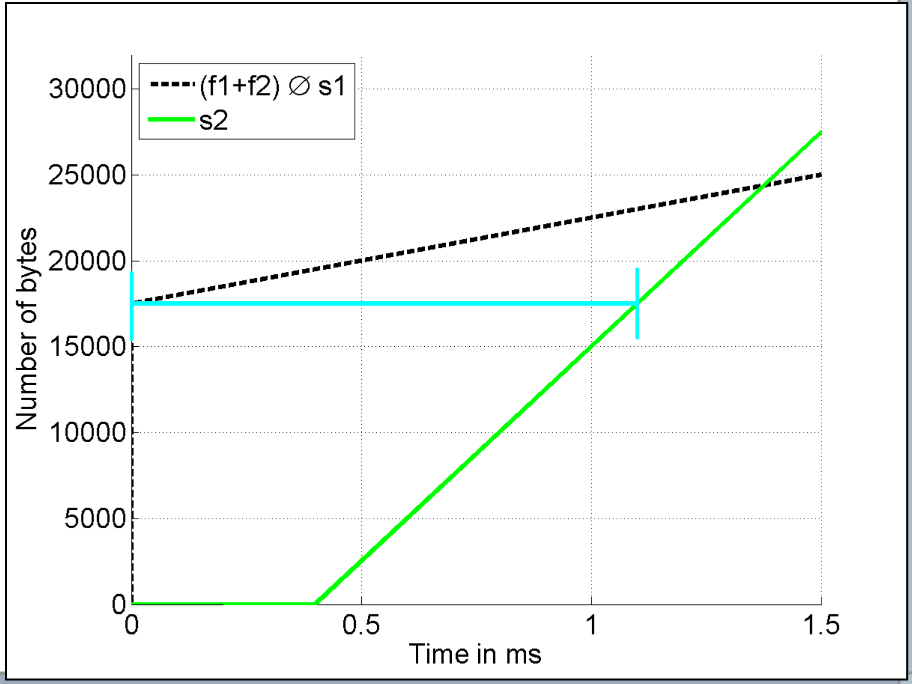
Short Introduction to Network Calculus (II)

- Flows in terms of Arrival Envelopes / Arrival Curves
- Service experienced by switch in terms of Service Curve

Example of fluid flows, preemptive

- f1 and f2 are multiplexed and traverse two servers / switches
- Flow of interest is f1
- Delay given by horizontal deviation

Node-by-Node Analysis



Origins of Network Calculus

- Cruz introduced Network Calculus as alternative to Queuing Theory
  - Investigated burstiness of traffic flows and the impact on the delay
  - Investigated buffer requirements
- Le Boudec shifted Network Calculus towards (min, +)-algebra
  - Reuse of convolution and deconvolution known from system theory
  - Instead of integration, use infimuum and supremum
- Network Calculus is a competitor to classical queuing theory, but focused on the worst case
- Latest techniques try to bring stochastic into the network calculus
  - Stochastic Network Calculus

Evolution of Network Calculus

- Evolution
  - From basic calculus over (min,+)-Algebra to
    - Stochastical extensions (SNC)
    - Tightness / Convolution-form networks
    - Linear optimization based approaches

early 90ies

Cruz

late 90ies

Le Boudec

since 2000

Jiang

Bouillard

Liebeherr

Schmitt

Fidler

SNC

DNC

Networks in Network Calculus

- Packet Based Networks
- Tandem Networks
- Feed Forward Networks
- Non Feed Forward Networks

Networks in Network Calculus

Networks we know so far

- Packet Based Networks
- Overlay Networks

But we abstract in the Network Calculus

- Tandem Networks
- Feed Forward Networks
- Non Feed Forward Networks

We differentiate here, because

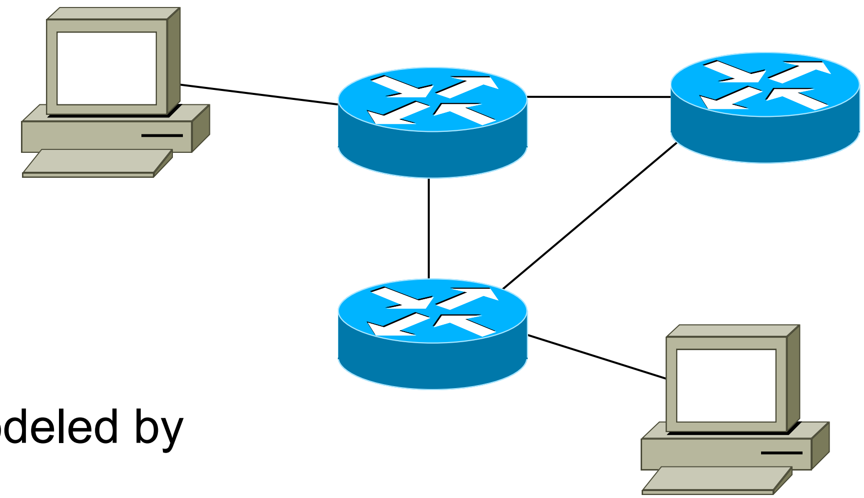
  - the type of such abstracted networks has impact on tightness of bounds

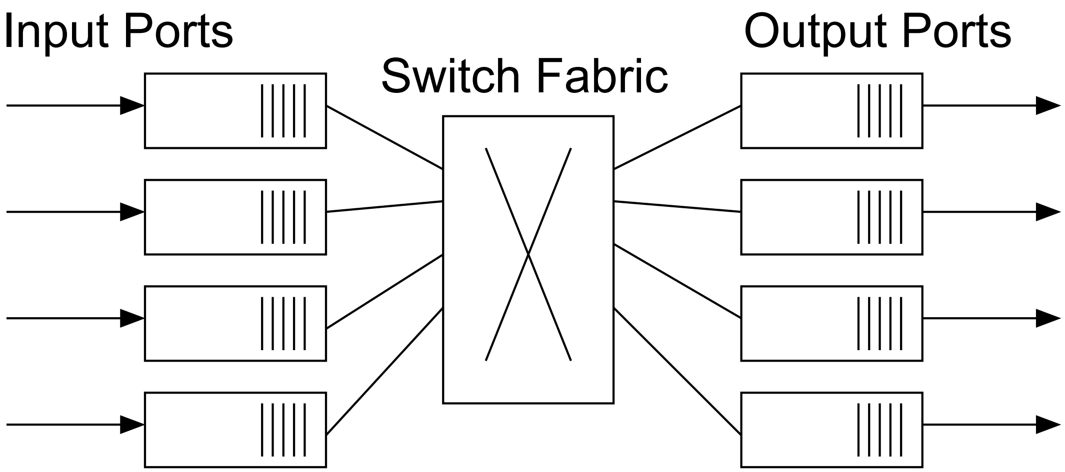
What are tight bounds ?

- Tight bounds: Can this worst case constellation ever be reached ?

Packet Switched Networks

- Example Network
- Basically, switches consist of input queues, switch fabric, and output queues
- IP routers/gateways modeled quite similar
- Each output queue will be later modeled by a Service Curve



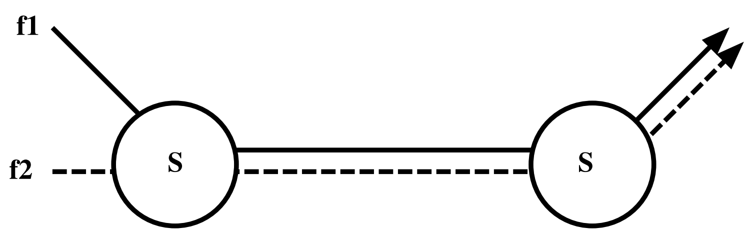


Tandem Network

- Tandem Network
  - Tandem of Servers
  - Multiplexing occurs only once
  - Exactly one path from source to destination

- Pay Burst Only Once (PBOO) should be applied here
- Also known as Convolution-Form Network

- There exist polynomial algorithms to determine tightest bounds
  - Optimization Based Approaches
  - Linear Programming

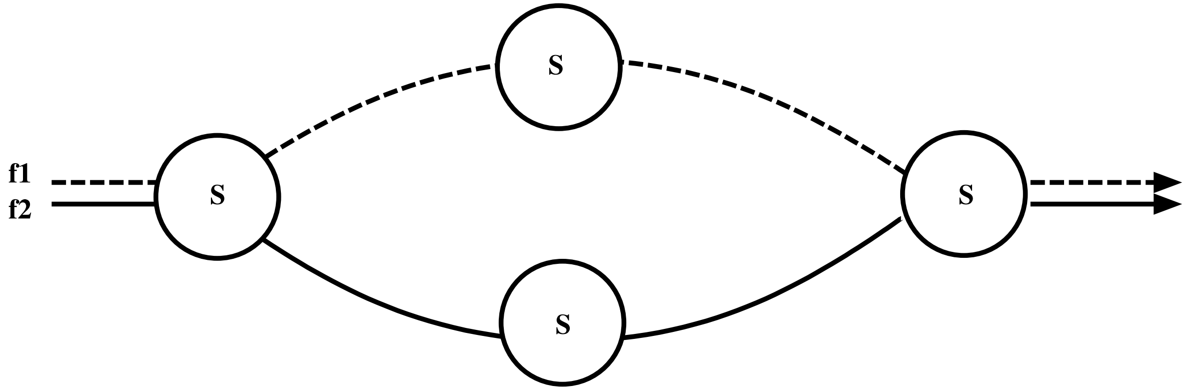


Feed-Forward Network

- Feed-Forward Network
- Directed Acyclic Graph
  - Multiplexing can occur several times
  - Several paths from source to destination possible

- Pay Multiplexing Only Once (PMOO) should be applied here

- Determining tight bounds was shown to be NP-hard
- There exist linear optimization based approaches



Non Feed-Forward Network

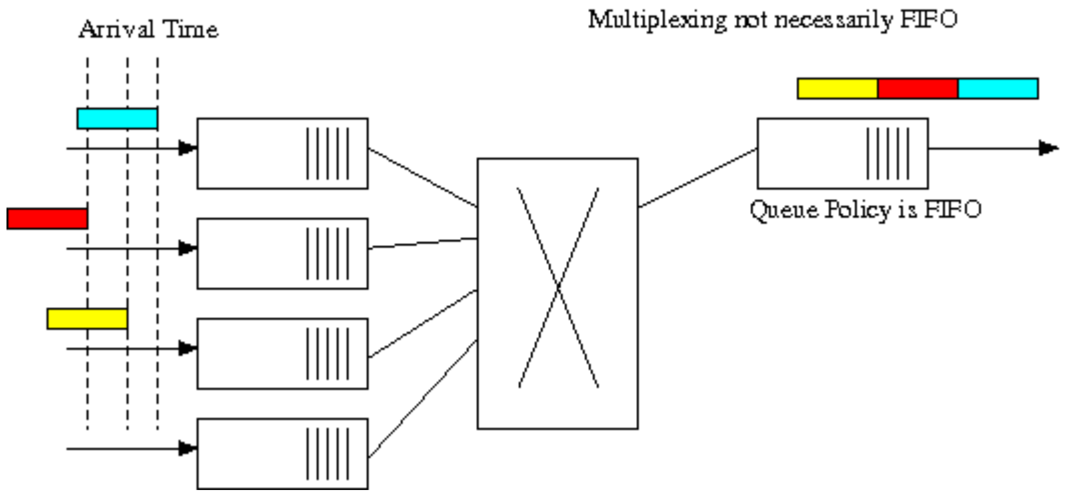
- Non Feed-Forward Network
- Cycles can occur in Non Feed-Forward Networks

- Network Calculus tends to deliver infinite bounds for those networks

- However there are some algorithms to turn Non Feed-Forward Networks to Feed-Forward Networks
  - Spanning Tree
    - restrictive, eliminates links
    - Creates tree
  - Turn Prohibition Algorithm [Sta2003]
    - Retain Graph Topology

Non-FIFO bounds

- Why do we talk about Non-FIFO bounds, queuing discipline should be FIFO ?
- Consider following situation in a packet switch



- Usually switch fabric tries to find maximum matching in order to serve as many input ports as possible
- For FIFO multiplexing, switches would have to store arrival time
- So, no FIFO (with respect to packet forwarding) is guaranteed

Traffic Models in the Network Calculus

- Token / Leaky Bucket in the Wild
- Token Bucket for Traffic Limiting
- Leaky Bucket for Traffic Shaping
- Token Bucket for Traffic Shaping

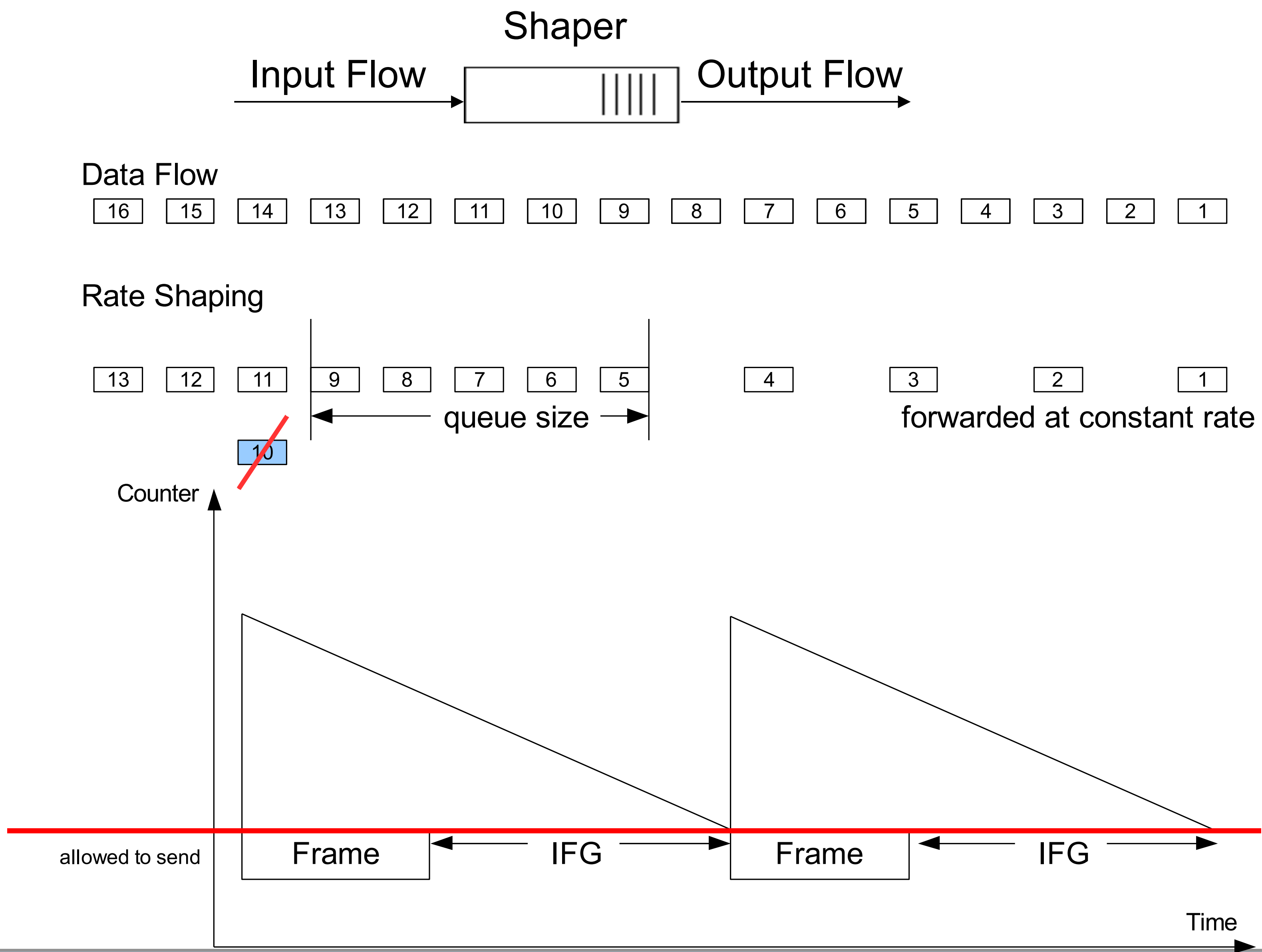
Token Bucket / Leaky Bucket in the Wild (1)

- Traffic Shapers shape traffic
  - In packet based networks we guarantee that the inter frame gap is greater or equal to the corresponding bandwidth

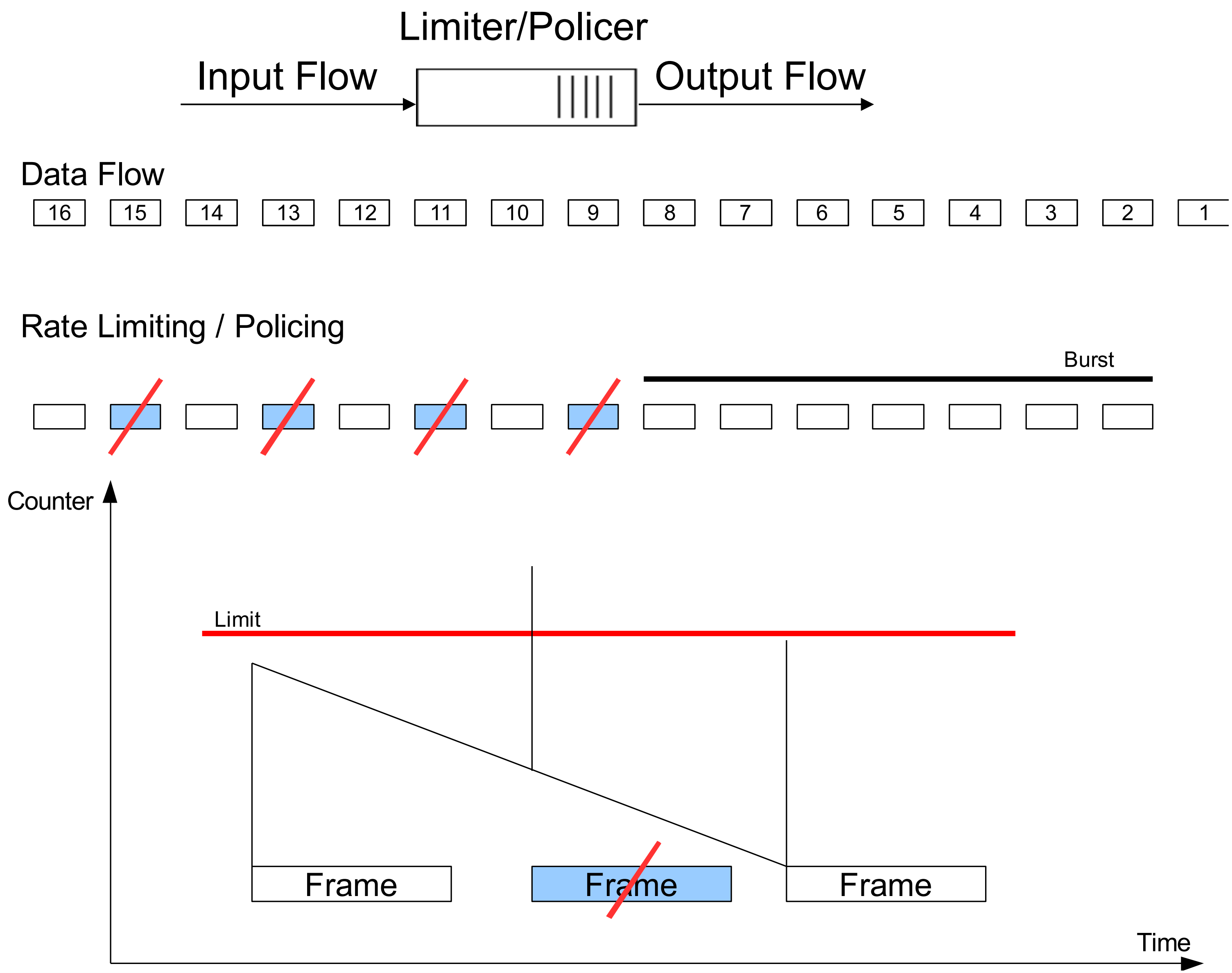
- Traffic Policers determine, whether a traffic flow is in accordance with a specified traffic pattern
  - If the burst is consumed, policer might trigger actions as
    - Dropping frames
    - Send PAUSE Message for Flow Control (According to IEEE 802.3x)

- Major difference: Traffic policers do not manipulate the inter frame gap while traffic shapers might do

Token Bucket / Leaky Bucket in the Wild (2)



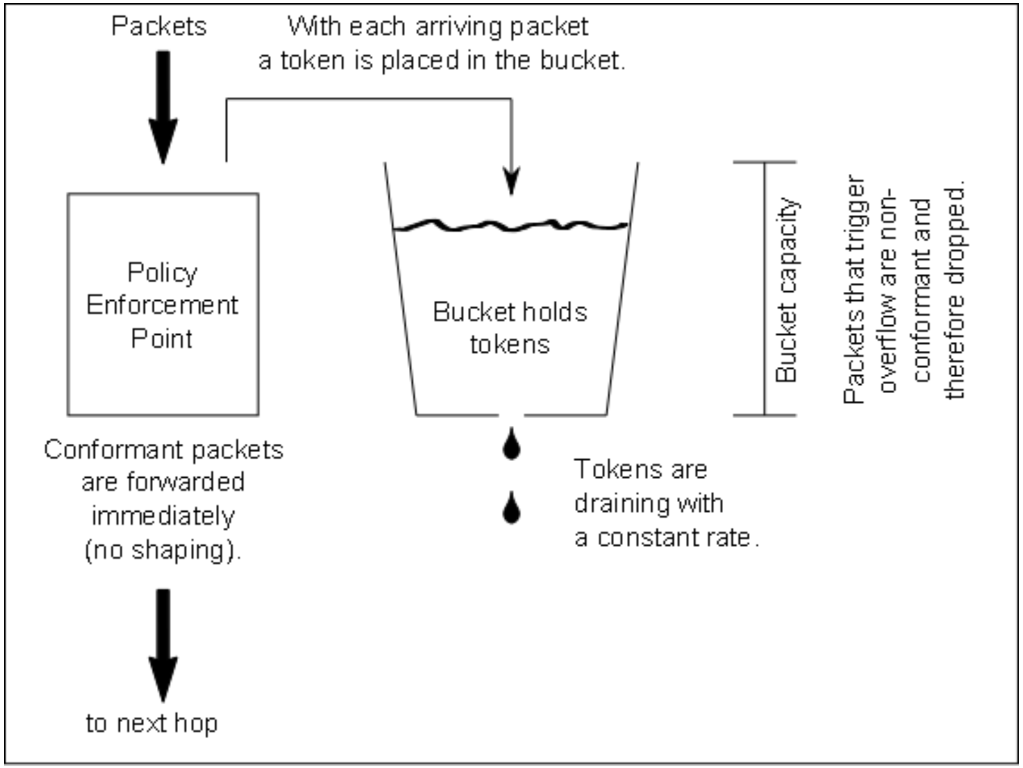
Token Bucket / Leaky Bucket in the Wild (3)



Token Bucket for Traffic Limiting

- Used to constrain flows in Network Calculus [Tan2002]
- Can be enforced by some sort of hardware limiters

- Packets are not delayed but
  - either removed from the traffic flow
  - or some flow control mechanism is triggered

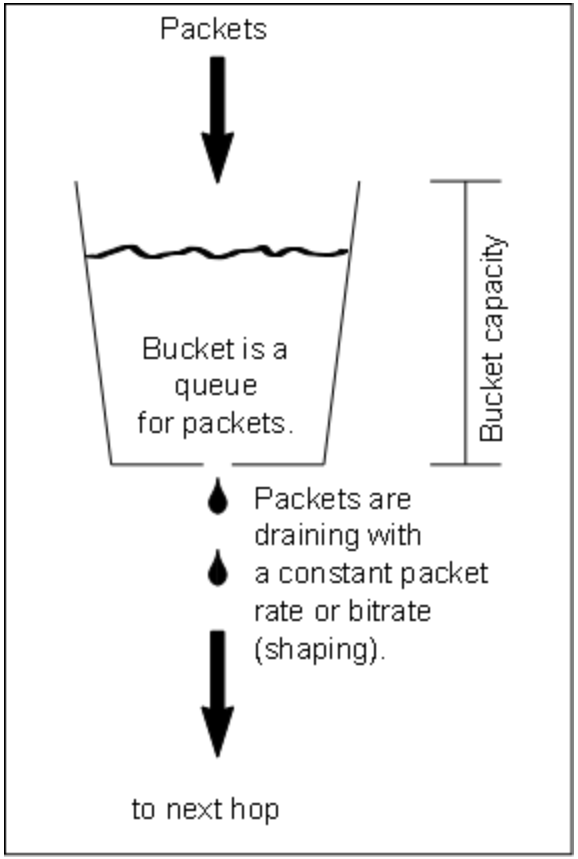


Leaky Bucket for Traffic Shaping

- Used to constrain flows in Network Calculus [Sta2001]
- Can be enforced by some sort of hardware shapers

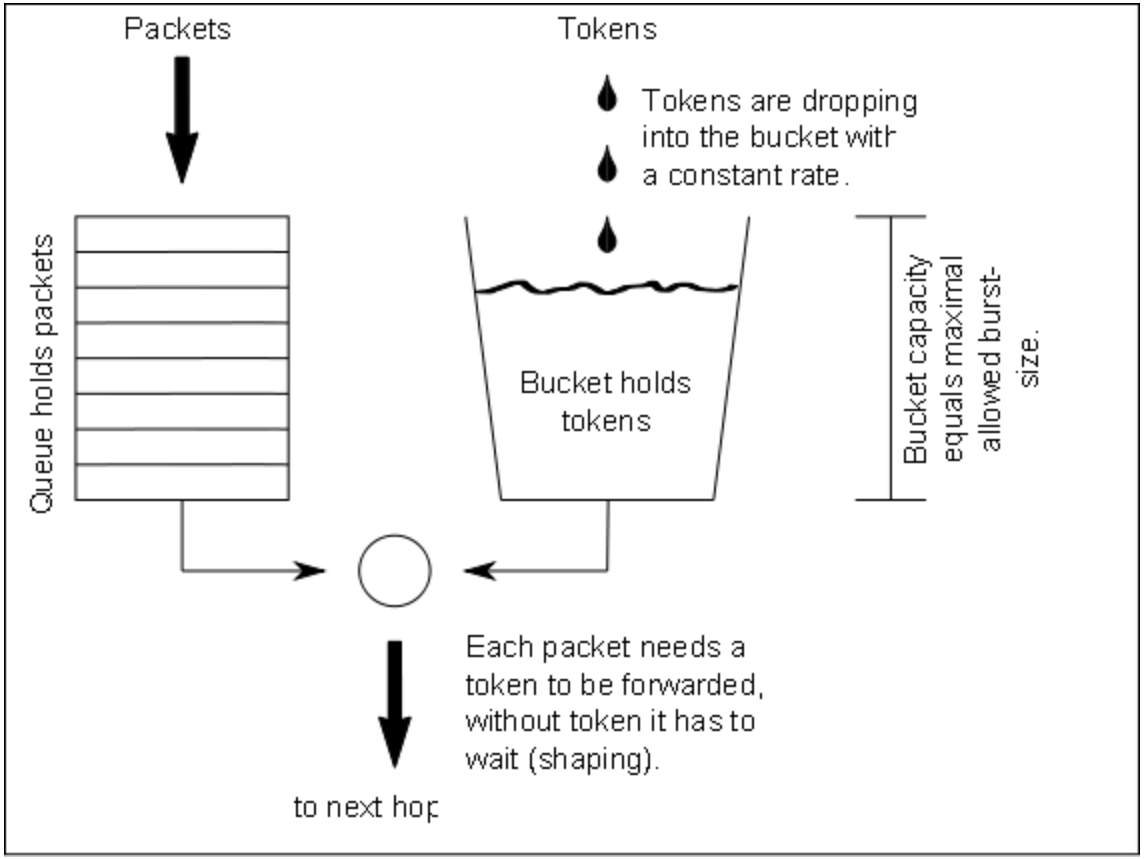
- Traffic is shaped, i.e., inter frame gap (IFG) is guaranteed to be greater than some specified value

- For this version, shaping already occures at first packet



Token Bucket for Traffic Shaping

- Extended version which allows some burstiness
- Shaping does not occur until burst is consumed



Multiple Token Bucket Models

Sometimes it is not reasonable, to model traffic with only one (peak) rate

$\Rightarrow$ we need several buckets

Other examples

- Model packets at line rate
  - You do not want the token bucket model with parameters
    - (200 Bytes, 100MBit/s)
    - Better: (200 Bytes, 100MBit/s, 10000 Bytes, 800kBit/s)
      - E.g. VoIP with input buffers, also buffers in protocol stack
- Intserv‘s TSPEC
  - Peak rate
  - Burst at peak rate
  - Average rate
  - Burst at average rate

Formalisms in the Network Calculus

- Min-plus Algebra
- Arrival and Service Curves
- Latency and Backlog Bounds

Min,+ Algebra

- Min,+ Algebra is a semi-ring, dioid on $(\mathbb{R} \cup \{+\infty\}, \wedge, +)$, so

- Closure and Associativity of $\wedge$
- Zero element existent for $\wedge$
- Idempotency and Commutativity of $\wedge$

- Closure and Associativity of +

- Zero element for $\wedge$ is absorbing for +
- Neutral element existent for +
- Distributivity of + with respect to

$\wedge$ is infimuum (or minimum if exists)
$\vee$ is supremum (or maximum if exists)

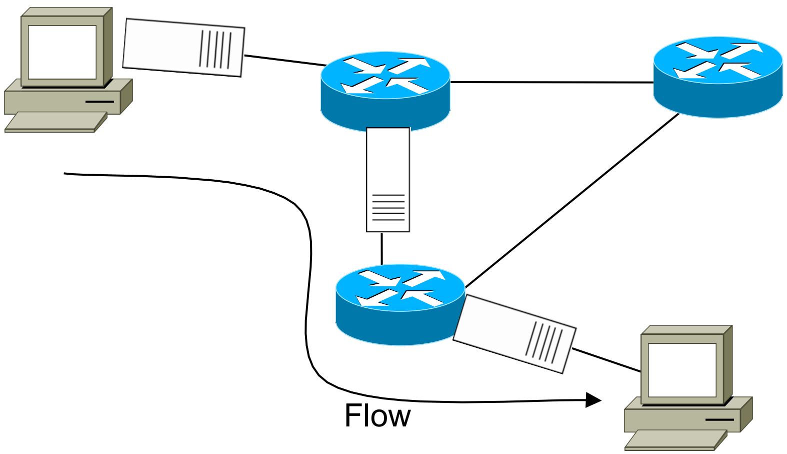
$$[x]^+ \equiv \max\{x, 0\}$$
$$[x]_1 \equiv \min\{x, 1\}$$

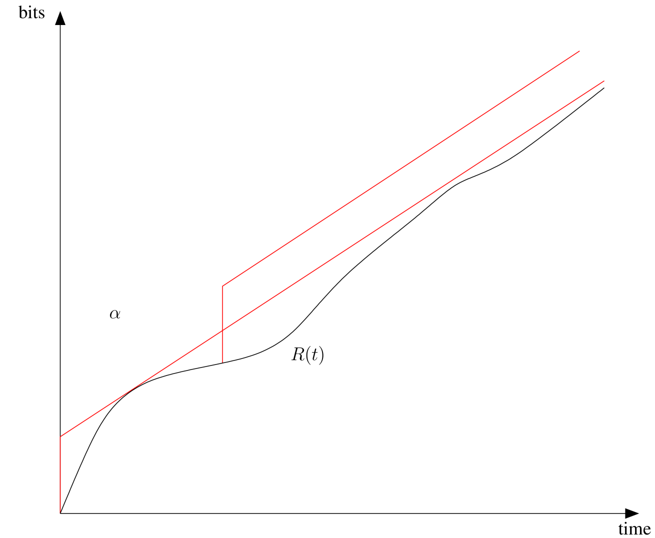
- [Bou2004]

- **(Closure of $\wedge$)** For all $a, b \in \mathbb{R} \cup \{+\infty\}$, $a \wedge b \in \mathbb{R} \cup \{+\infty\}$.
- **(Associativity of $\wedge$)** For all $a, b, c \in \mathbb{R} \cup \{+\infty\}$, $(a \wedge b) \wedge c = a \wedge (b \wedge c)$.
- **(Existence of a zero element for $\wedge$)** There is some $\mathrm{e} = +\infty \in \mathbb{R} \cup \{+\infty\}$ such that for all $a \in \mathbb{R} \cup \{+\infty\}$, $a \wedge \mathrm{e} = a$.
- **(Idempotency of $\wedge$)** For all $a \in \mathbb{R} \cup \{+\infty\}$, $a \wedge a = a$.
- **(Commutativity of $\wedge$)** For all $a, b \in \mathbb{R} \cup \{+\infty\}$, $a \wedge b = b \wedge a$.
- **(Closure of $+$)** For all $a, b \in \mathbb{R} \cup \{+\infty\}$, $a + b \in \mathbb{R} \cup \{+\infty\}$.
- **(Associativity of $+$)** For all $a, b, c \in \mathbb{R} \cup \{+\infty\}$, $(a + b) + c = a + (b + c)$.
- **(The zero element for $\wedge$ is absorbing for $+$)** For all $a \in \mathbb{R} \cup \{+\infty\}$, $a + \mathrm{e} = \mathrm{e} = \mathrm{e} + a$.
- **(Existence of a neutral element for $+$)** There is some $u = 0 \in \mathbb{R} \cup \{+\infty\}$ such that for all $a \in \mathbb{R} \cup \{+\infty\}$, $a + u = a = u + a$.
- **(Distributivity of $+$ with respect to $\wedge$)** For all $a, b, c \in \mathbb{R} \cup \{+\infty\}$, $(a \wedge b) + c = (a + c) \wedge (b + c) = c + (a \wedge b)$.

Arrival Curve

- Arrival Curve specifies a traffic envelope to arrivals
- Used for nodes creating traffic
- but also for forwarding nodes at output
- How do we model arrivals of traffic in networks ?



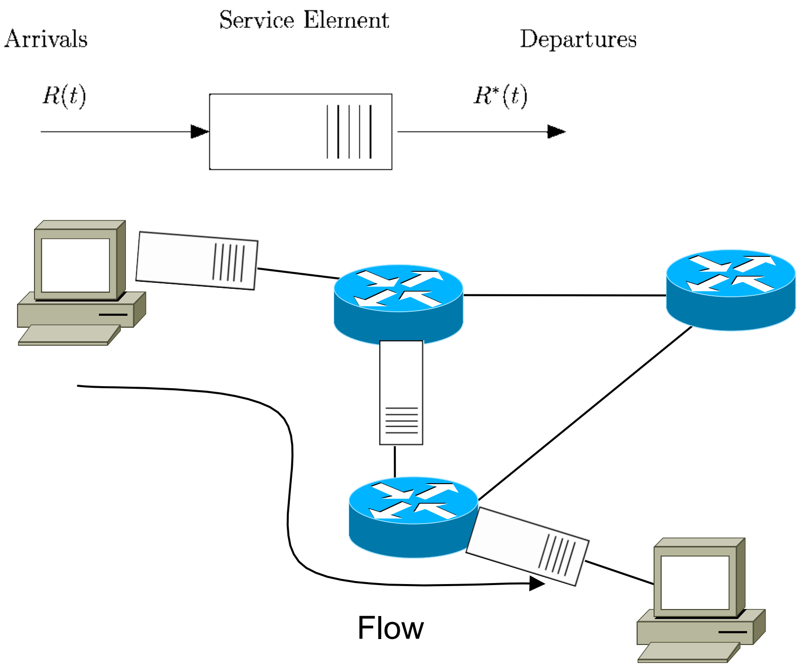


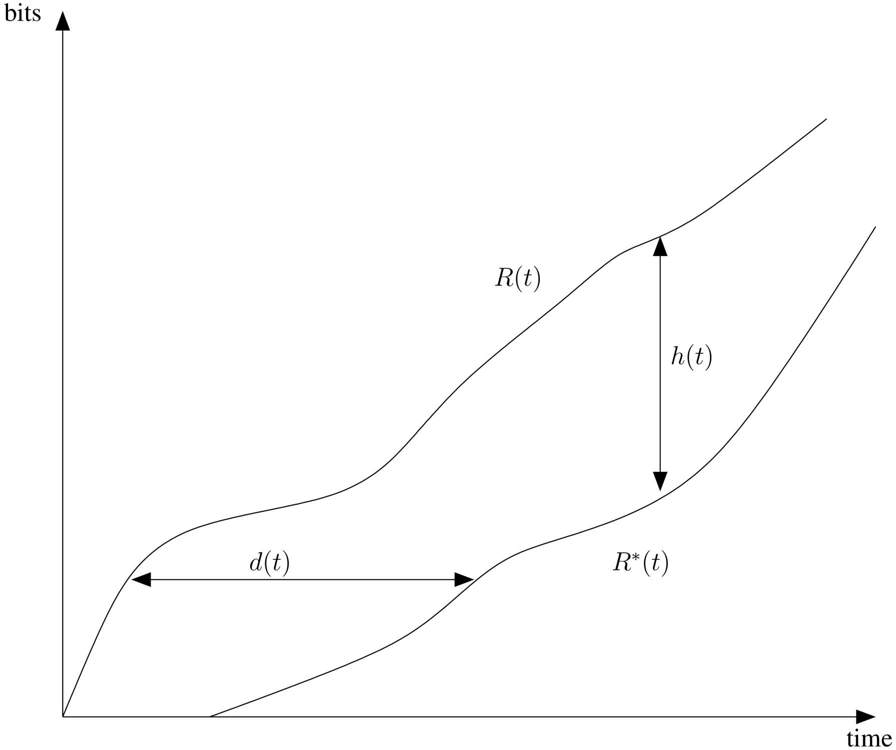
- Given a wide-sense increasing functions α defined for $t \geq 0$
  A flow R is constrained by α if and only if for all $s \leq t$

$$R(t) - R(s) \leq \alpha(t - s)$$

Output Arrival Curve

- Horizontal deviation d(t) gives FIFO bound on delay
  - One bit in Arrival corresponds to exactly one bit in Departure
- Vertical deviation h(t) gives maximum backlog



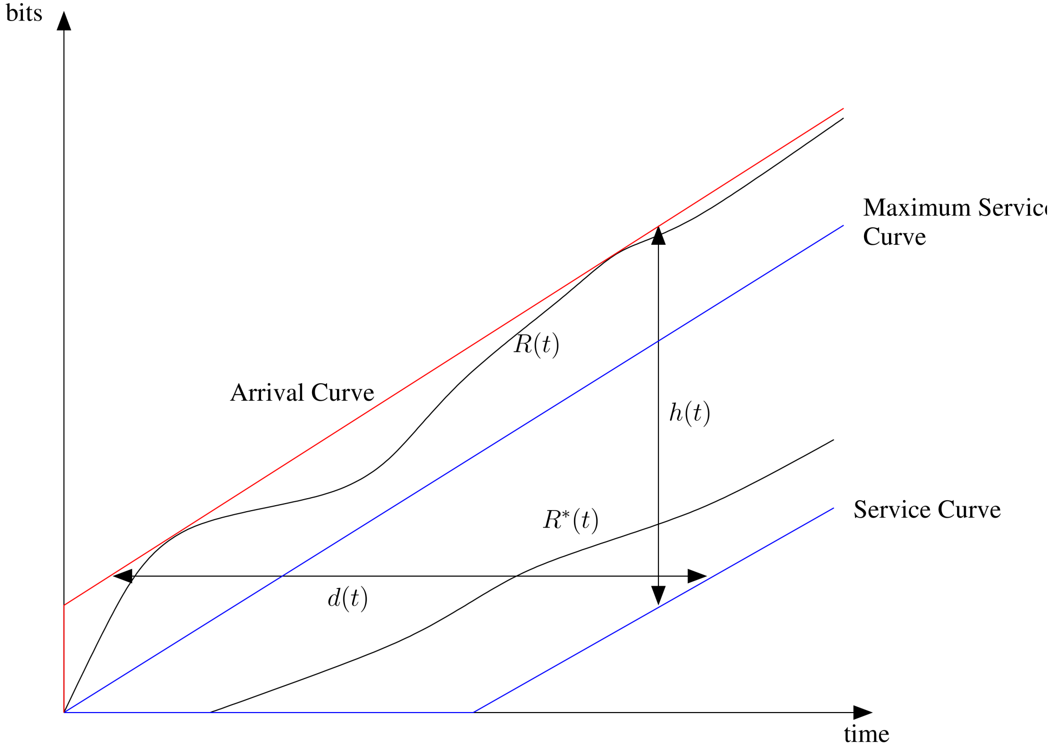


- How to determine bound for departures ?

Service Curve

- Concept to abstract service offered from systems
- In accordance to scheduling (GPS, EDF) disciplines

- Consider a system S and a flow through S with input and output function R and R^*. S offers to the flow a service curve β if and only if β is wide sense increasing, $\beta(0) = 0$ and $R^* \geq R \otimes \beta$.



Strict Service Curve

- We say that system S offers a strict service curve β to a flow if, during any backlogged period of duration u, the output of the flow is at least equal to $\beta(u)$.
- Backlogged Period: Timespan where backlog is greater than 0.

Token Bucket Constrained Arrival Curve

- Token Bucket, (σ, ρ) -constrained

- Burst often gives maximum packet size
- Rate gives the average rate
- If peak rate and average rate should also be modeled, we use dual or even multiple Token Buckets

- Example for (σ, ρ)-constrained arrival curve with

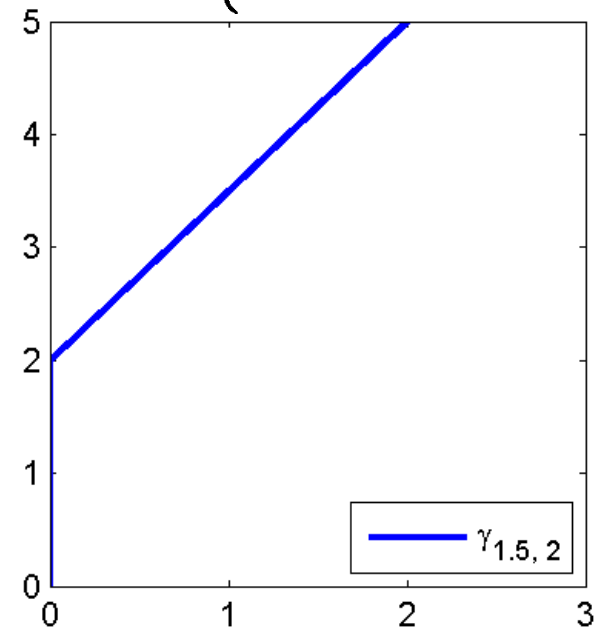
  $$\sigma = 1.5$$
  $$\rho = 2$$

- Example for dual Token Bucket IntServ TSpec (r,b,p,M)
  - average rate r, burst b, peak rate p, maximum packet size M
  - formal

    $\gamma_{r,b} \wedge \gamma_{p,M}$

    $\wedge$ is minimum

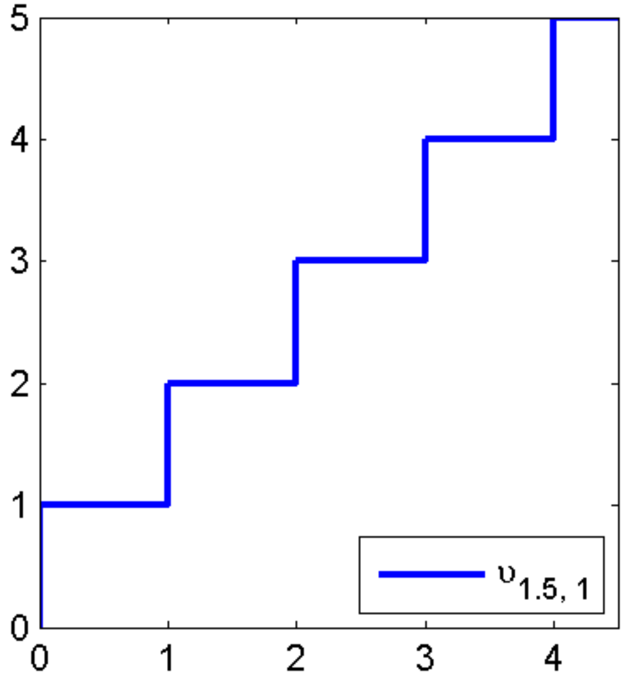
$$\gamma_{r,b}(t) = \begin{cases} rt + b & \text{if } t > T \\ 0 & \text{otherwise} \end{cases}$$



Staircase Arrival Curve

- Periodic Arrival Curve
- Used for Discrete Events such as
  - packet bursts
  - periodic messages

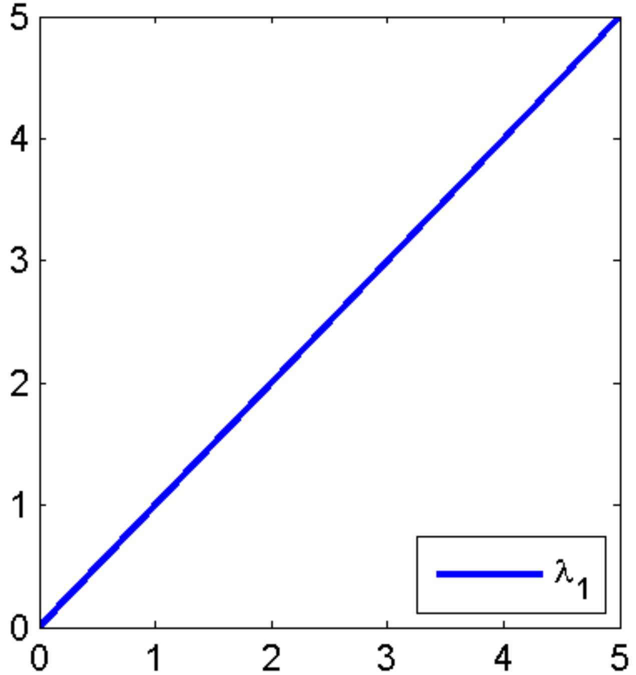
$$v_{T,\tau}(t) = \begin{cases} \lceil \frac{t+\tau}{T} \rceil & \text{if } t > T \\ 0 & \text{otherwise} \end{cases}$$



Service Curves – Strict Service Curve

- Service Curve – Strict Service Curve
  - Generalized Processor Sharing (GPS)
  - Theoretical model to serve several flows in parallel
  - Practical implementation requires different service curve

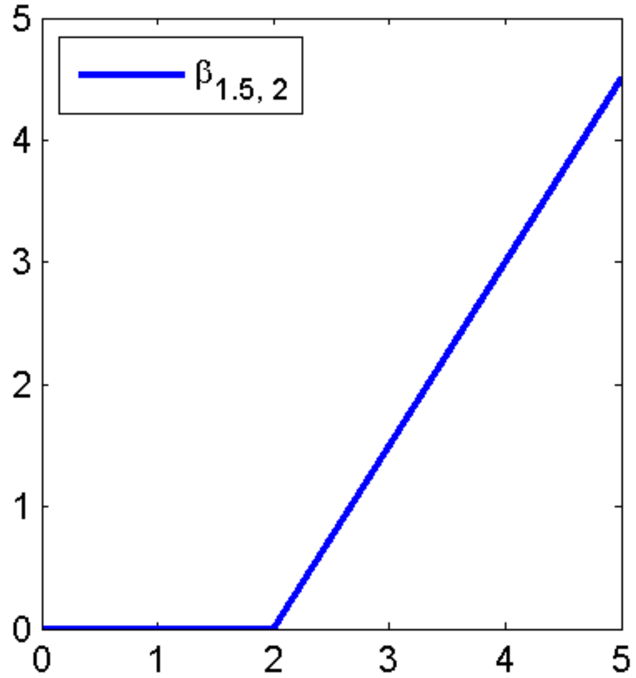
$$\lambda_R(t) = \begin{cases} Rt & \text{if } t > T \\ 0 & \text{otherwise} \end{cases}$$



Service Curves - Rate Latency

- Service Curve – Service of a Forwarding Node / Switch / Router
  - Packetized GPS
  - Weighted Fair Queuing
  - Intserv Guaranteed Service

$$\beta_{R,T} = R[t-T]^+ = \begin{cases} R(t-T) & \text{if } t > T \\ 0 & \text{otherwise} \end{cases}$$



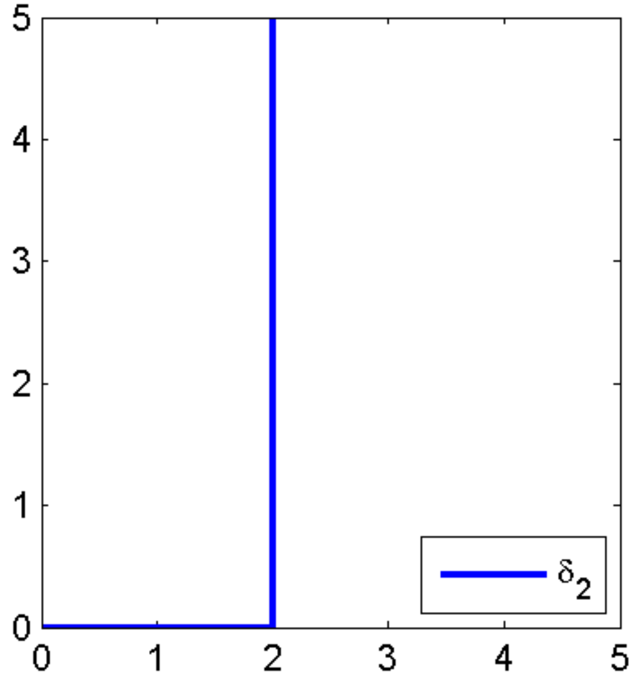
Service Curve - Non-Preemtive Priority Node

- Service Curve Non-Preemptive Priority Node
  - Constant Rate C
  - High priority flow $\beta_{C,l_{max}/C}$
  - Low priority flow $\beta_{C-r,b/(C-r)}$
- Rate Latency Service Curve is the standard tool to model
  - Guaranteed Rate Servers
  - Practical implementations of GPS
  - Non-Preemptiveness
  - Store-and-Forward delay

Service Curves - EDF

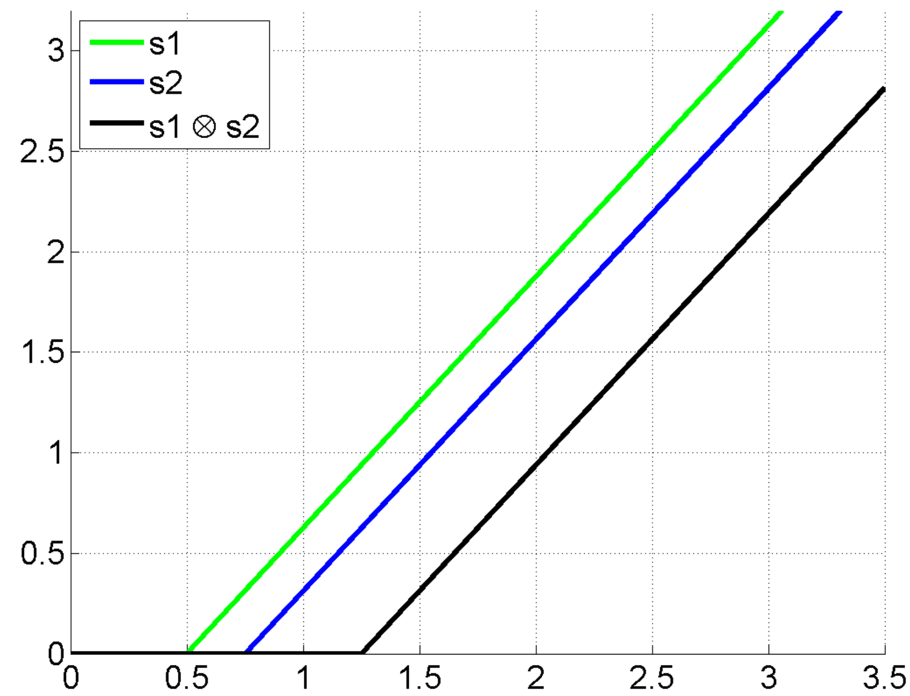
- Service Curve
  - Burst Delay
  - Earliest Deadline First (EDF)
  - Guaranteed Delay Node

$$\delta_T(t) = \begin{cases} +\infty & \text{if } t > T \\ 0 & \text{otherwise} \end{cases}$$



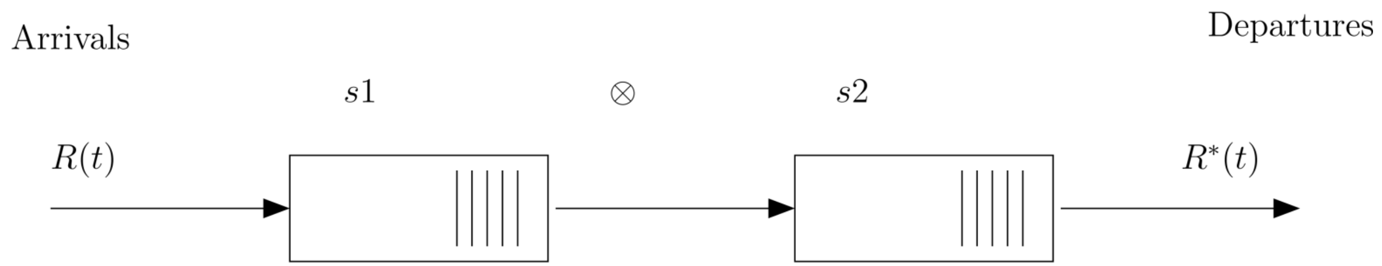
Convolution (I)

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



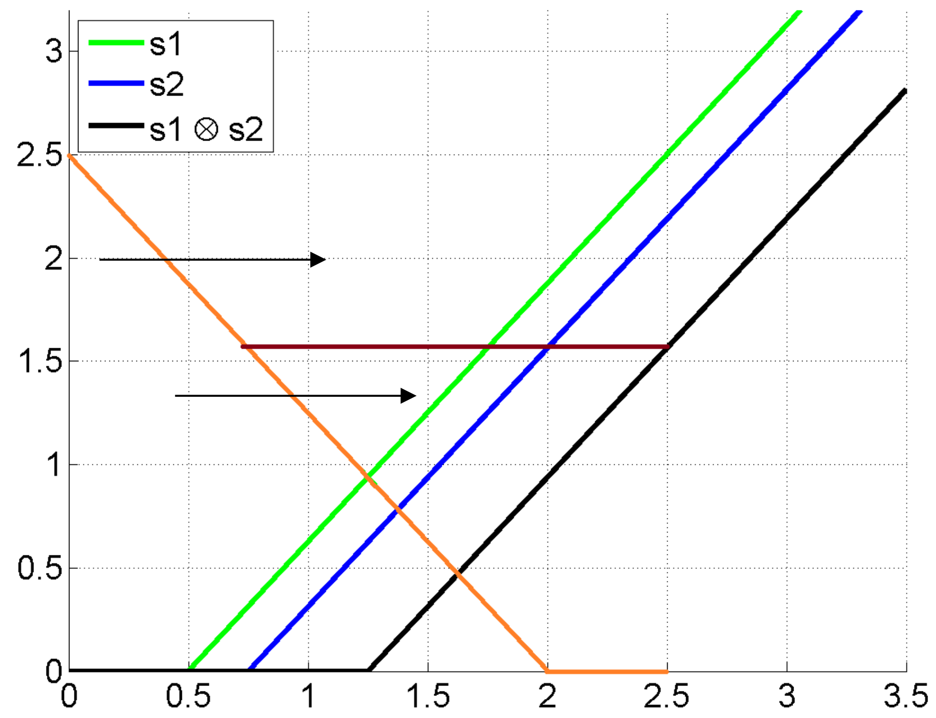
Application

– Infimuum (lower) bound for output curve
– Latest appearance of bits at output
– Concatenation of Servers
– Convolution of tandem of service curves (convolution-form networks)



Convolution (II)

- Move mirrored green curve to the right (orange curve)
- Determine minimum of sum of orange and blue curve

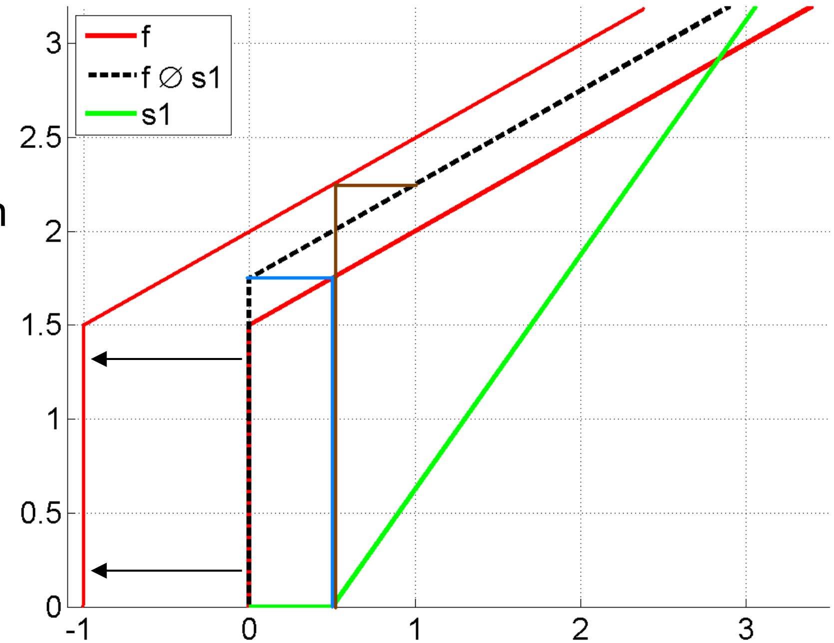


$$(f \otimes g)(t) = \inf_{0 \leq s \leq t} \{f(t-s) + g(s)\}$$

Deconvolution

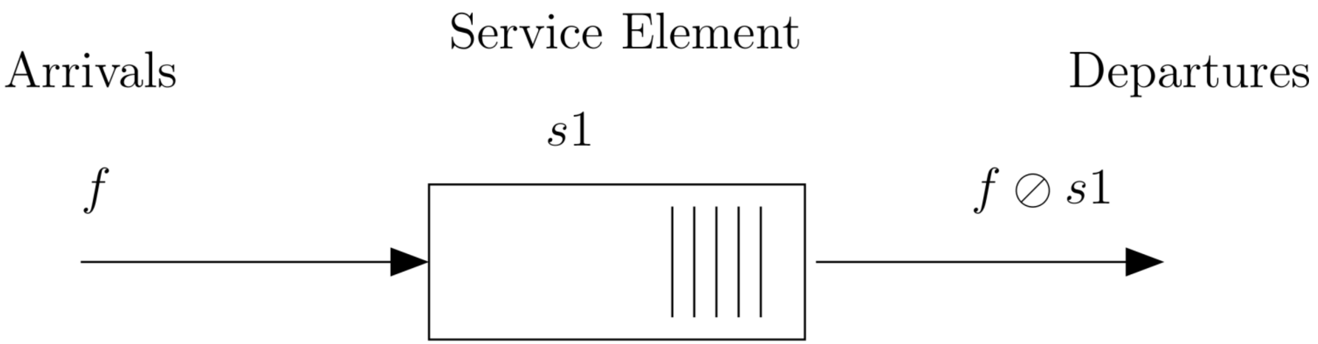
$$(f \oslash g)(t) = \sup_{u \geq 0}\{f(t+u) - g(u)\}$$

- Move red curve to the left
- Determine maximum of difference between red curve and green curve



Application

- Supremum (upper) bound for output curve
- Earliest appearance of bits at output



Tighten Bounds

- Convolution Form Network / Pay Bursts Only Once
- Pay Multiplexing Only Once

Node-by-Node Edge Analysis

- Obvious way to calculate worst case delays
  - Calculate delay per traversing node and add them up
  - Also known under the term
    - **Node-by-Node Analysis**
    - Implemented as Total Flow Analysis (TFA) in DISCO
  - Problem in terms of overestimation:
    In reality, burst should only be paid at the first node. With Node-by-Node, it will be paid at every traversing node.

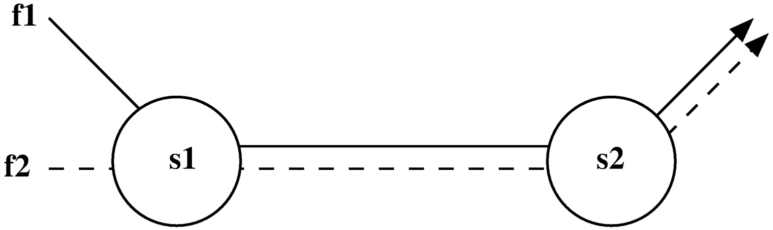
Tightness of Network Calculus Bounds

However, with the so called Node-by-Node Analysis (as seen before)

- Latency is determined at each node, such that burst is paid at every server, i.e., s1 as well as s2
- Also known as algorithm: Total Flow Analysis (TFA)

Tightening bounds

- „Pay Bursts Only Once“ [RIZ2005]
  - Burst will only be paid at first node
  - Edge-by-Edge Analysis (First: Service Curve over all edges, Then: horizontal deviation)
  - Also known as algorithm: Separated Flow Analysis (SFA)
  - Addresses the following case

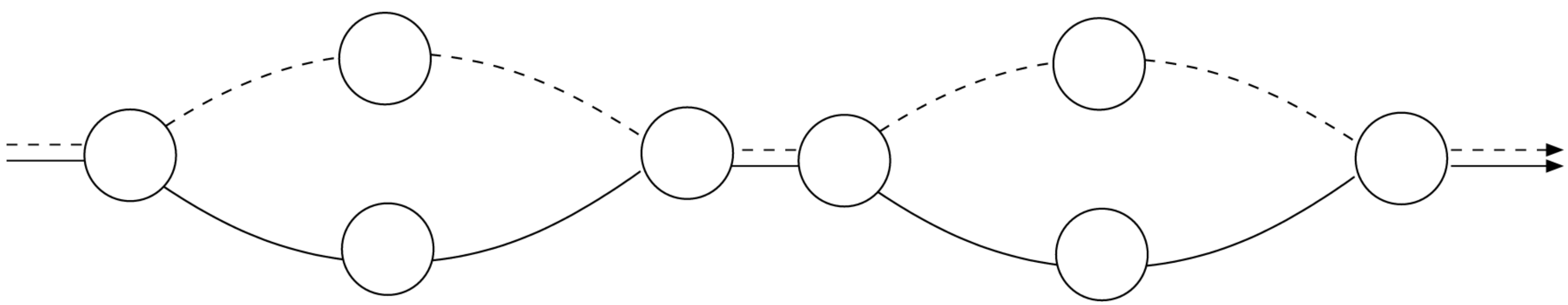


$$\beta_{SFA} = [s_1 - f_1]^+ \otimes [s_2 - s_1 \oslash f_1]^+$$

Tightness of Network Calculus Bounds

Tightening bounds

- „Pay Multiplexing Only Once“ [SCH2008]
  - If flow is multiplexed several times, SFA will pay too much at each multiplexing
  - Also known as algorithm: PMOO-SFA
  - Edge-by-Edge Analysis (First: Service Curve over all edges, Then: horizontal deviation)
    - Idea: Eliminate rejoining flows from service curve
    - Addresses the following case

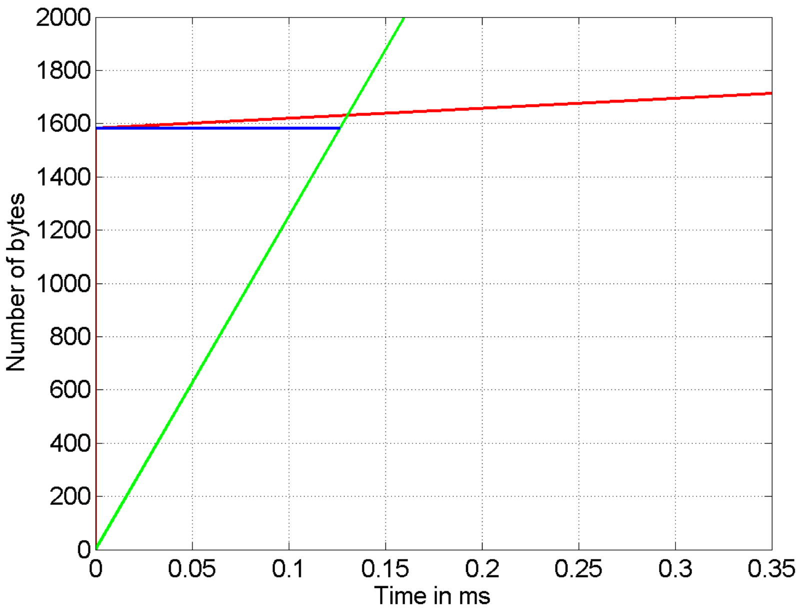


Non-Preemptiveness of Switched Ethernet (I)

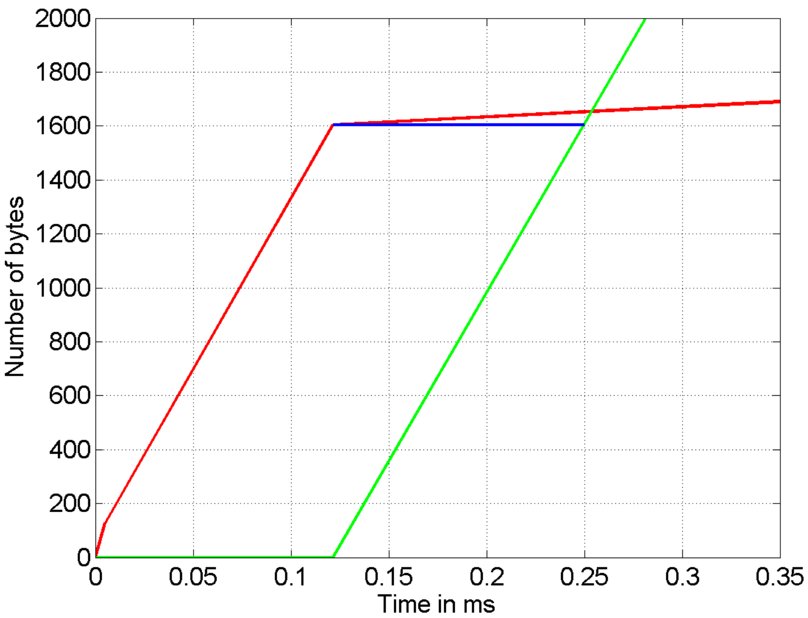
- We showed how to determine worst cases in fluid flow models
- But how to deal with non-preemptiveness of Switched Ethernet ?

- Mapping by

Discrete Sized Bursts



Additional latency in Rate Latency Service Curve (Packetizer)



Non-Preemptiveness of Switched Ethernet (II)

- The discrete bursts approach in switched Ethernet has some pitfalls:

⇒ Packet bursts must be preserved when not modeled by additional rate latency

⇒ Okay for TFA

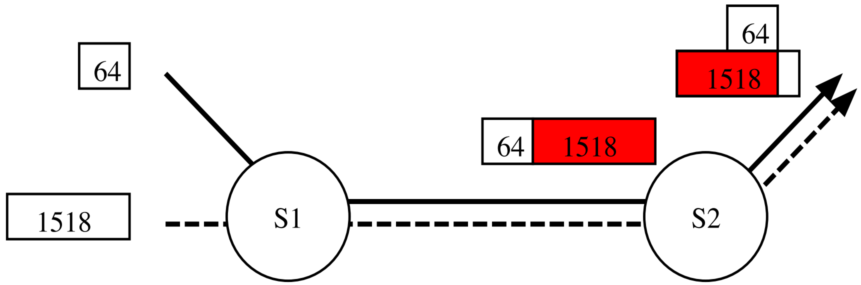
⇒ Store-and-forward delay of flow of interest must be added to SFA result (If *n* nodes, add *(n-1)* times the store-and-forward delay)

⇒ PMOO does not preserve packet bursts

Non-Preemptiveness of Switched Ethernet (III)

However, NC cannot map the following situation accurately (speed is Fast Ethernet):

- Assume a small packet being delayed by a larger packet
- At Server/Switch S1, the small packet is delayed by the full large packet
- At Server/Switch S2, the small packet is delayed only by the remaining 1454 bytes

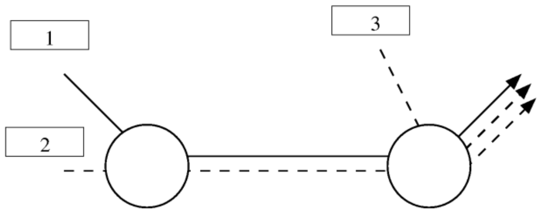


Approach	Calculation
TFA	$f = f_1 + f_2$
	$g = s_1,\ h = s_2$
	$v(f,g),\ v(f,h)$
RL	$v(f,g) + v(f,h) = 0.2566\text{ms}$
DB	$v(f,g) + v(f,h) = 0.2531\text{ms}$
SFA	$f = f_2$
	$g = [s_1 - f_1]^+ \otimes [s_2 - f_1 \oslash s_1]^+$
RL	$v(f,g) = 0.4914\text{ms}$
DB	$v(f,g) = 0.2518\text{ms}$
PMOO-SFA	$f = f_2, g = [s_1 \otimes s_2 - f_1]^+$
RL	$v(f,g) = 0.3681\text{ms}$
DB	$v(f,g) = 0.1285\text{ms}$

- But for FastEthernet, exact worst case is 0.2480 ms (omitting IFG and preamble)

Non-Preemptiveness of Switched Ethernet (IV)

- Additionally, NC can not map the following situation accurately:
- Assume three equally sized frames in a simple network
- Packet of interest is 1



1. Packet 1 and 2 arrive at first switch, Packet 1 is delayed by Packet 2

2. Packet 2 is transmitted and arrives at second switch as Packet 3 does

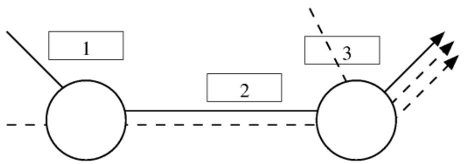
3. Packet 2 waits at second switch until transmission of Packet 3 finished

4. Packet 1 will be delayed by Packet 2

⇒ Additional delay by 2 packets
⇒ But NC gives additional delay of 3 packets

Non-Preemptiveness of Switched Ethernet (IV)

- However, NC can not map the following situation accurately:
- Assume three equal sized frames in a simple network
- Packet of interest is 1



1. Packet 1 and 2 arrive at first switch, Packet 1 is delayed by Packet 2

2. Packet 2 is transmitted and arrives at second switch as Packet 3 does

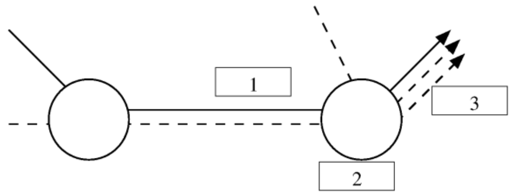
3. Packet 2 waits at second switch until transmission of Packet 3 finished

4. Packet 1 will be delayed by Packet 2

⇒ Additional delay by 2 packets
⇒ But NC gives additional delay of 3 packets

Non-Preemptiveness of Switched Ethernet (IV)

- However, NC can not map the following situation accurately:
  - Assume three equal sized frames in a simple network
  - Packet of interest is 1



1. Packet 1 and 2 arrive at first switch, Packet 1 is delayed by Packet 2

2. Packet 2 is transmitted and arrives at second switch as Packet 3 does

3. Packet 2 waits at second switch until transmission of Packet 3 finished

4. Packet 1 will be delayed by Packet 2

$\Rightarrow$ Additional delay by 2 packets
$\Rightarrow$ But NC gives additional delay of 3 packets

Non-Preemptiveness of Switched Ethernet (IV)

- However, NC can not map the following situation accurately:
- Assume three equal sized frames in a simple network
- Packet of interest is 1

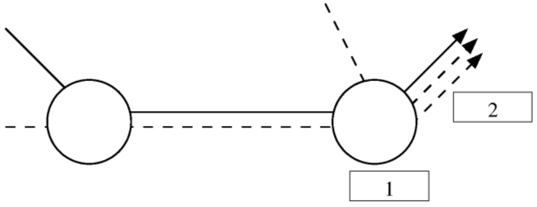
1. Packet 1 and 2 arrive at first switch, Packet 1 is delayed by Packet 2

2. Packet 2 is transmitted and arrives at second switch as Packet 3 does

3. Packet 2 waits at second switch until transmission of Packet 3 finished

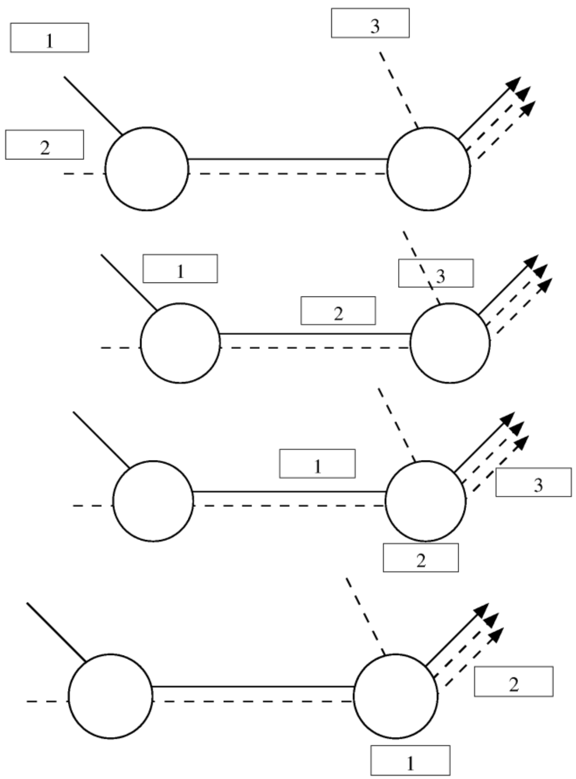
4. Packet 1 will be delayed by Packet 2

⇒ Additional delay by 2 packets
⇒ But NC gives additional delay of 3 packets



Non-Preemptiveness of Switched Ethernet (IV)

- However, NC can not map the following situation accurately:
  - Assume three equal sized frames in a simple network
  - Packet of interest is 1



1. Packet 1 and 2 arrive at first switch, Packet 1 is delayed by Packet 2

2. Packet 2 is transmitted and arrives at second switch as Packet 3 does

3. Packet 2 waits at second switch until transmission of Packet 3 finished

4. Packet 1 will be delayed by Packet 2

$\Rightarrow$ Additional delay by 2 packets
$\Rightarrow$ But NC gives additional delay of 3 packets

Excursion to Network Analysis Tools

- DISCO Network Analyzer
http://disco.informatik.uni-kl.de/

- COINC
http://perso.bretagne.ens-cachan.fr/~bouillar/coinc/

- CyNC
http://www.control.aau.dk/~henrik/CyNC/

- Real Time Calculus
http://www.mpa.ethz.ch/

References

[Tan2002] Computer Networks (4th Edition), 2002, Andrew S. Tanenbaum

[Sta2001] High-Speed Networks and Internets: Performance and Quality of Service (2nd Edition), William Stallings

[Sta2003] Starobinski, D.; Karpovsky, M. & Zakrevski, L. A.
Application of network calculus to general topologies using turn-prohibition, IEEE/ACM Transaction on Networking, IEEE Press, 2003, 11, 411-421

[Sch2008] Schmitt, J. B.; Zdarsky, F. A. & Martinovic, I. Improving Performance Bounds in Feed-Forward Networks by Paying Multiplexing Only Once Measuring, Modelling and Evaluation of Computer and Communication Systems (MMB), 2008 14th GI/ITG Conference -, Proc. of the 14th Conference on Measuring, Modelling and Evaluation of Computer and Communication Systems (MMB), 2008

[Sch2005] Schmitt, J. B. & Zdarsky, F. A. The DISCO Network Calculator: A Toolbox for Worst Case Analysis Proc. of the 1st international conference on Performance evaluation methodolgies and tools. ValueTools 2006, 2006

[Bou2004] Le Boudec, J. Network Calculus: A Theory of Deterministic Queuing Systems for the Internet 2004

[Jia2008] Jiang, Y. & Liu, Y. Stochastic network calculus Springer-Verlag New York Inc, 2008