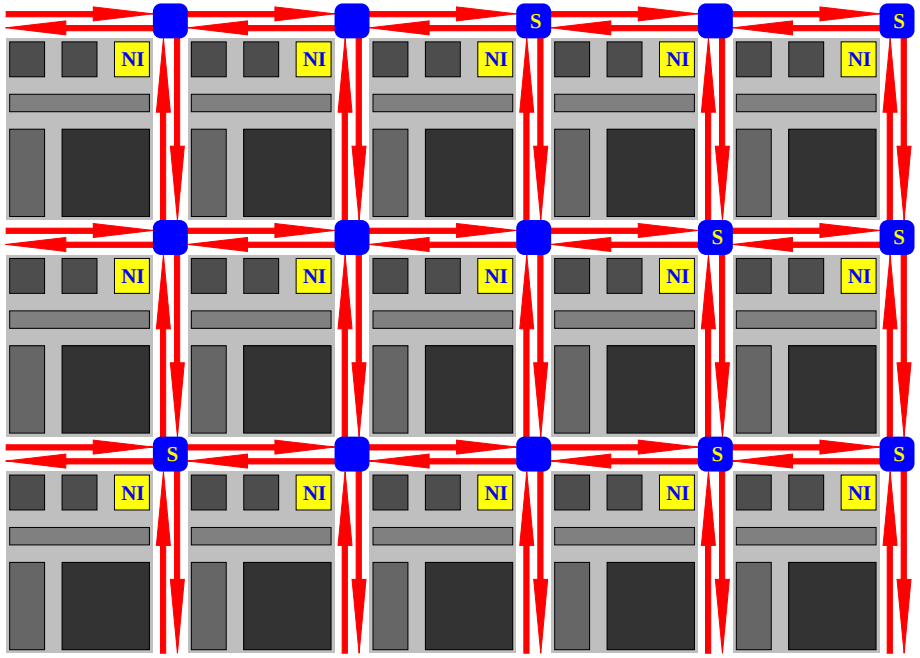
The Nostrum Network on Chip



Axel Jantsch, Zhonghai Lu, Mikael Millberg, Rikard Thid, Johnny Öberg, Erland Nilsson, Shashi Kumar, Ahmed Hemani, et al.

Royal Institute of Technology, Stockholm

November 2007

Overview

Topology and Structure

The Network Layer and the Switch

TDM Allocation for Quality of Service

Regulated Flows

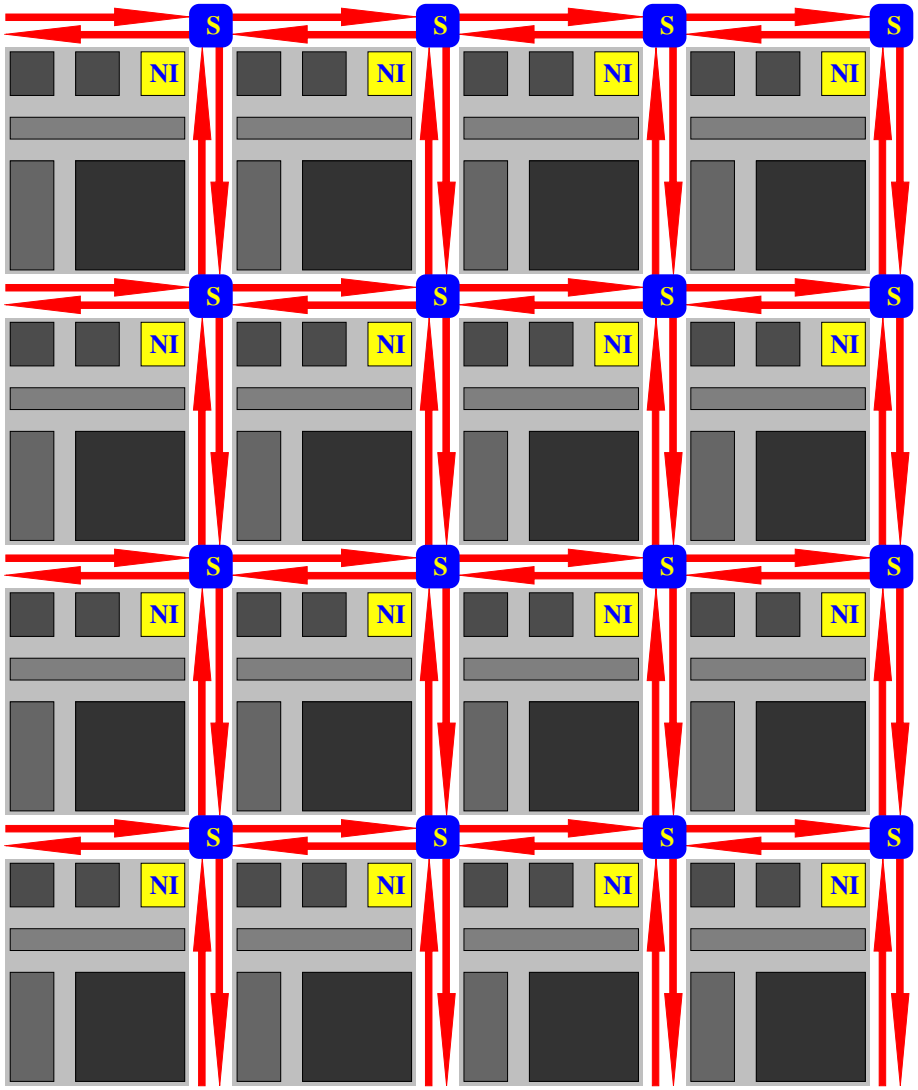
Data Protection

Clocking

Dynamic Voltage Scaling

Network Simulator

Nostrum Topology: Mesh



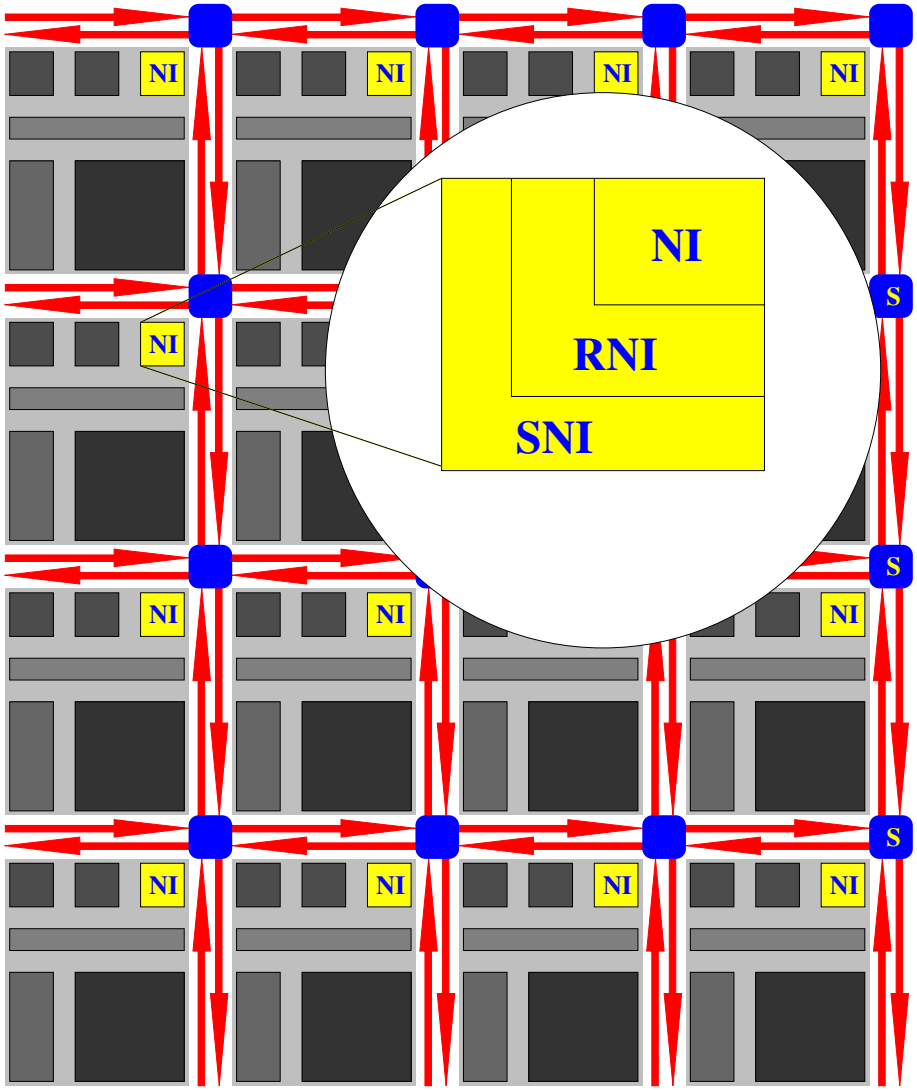
Characteristics:

- Resource-to-switch ratio: 1
- A switch is connected to 4 switches and 1 resource
- A resource is connected to 1 switch
- Average distance: $2/3n$
- Bisection bandwidth: $2n$

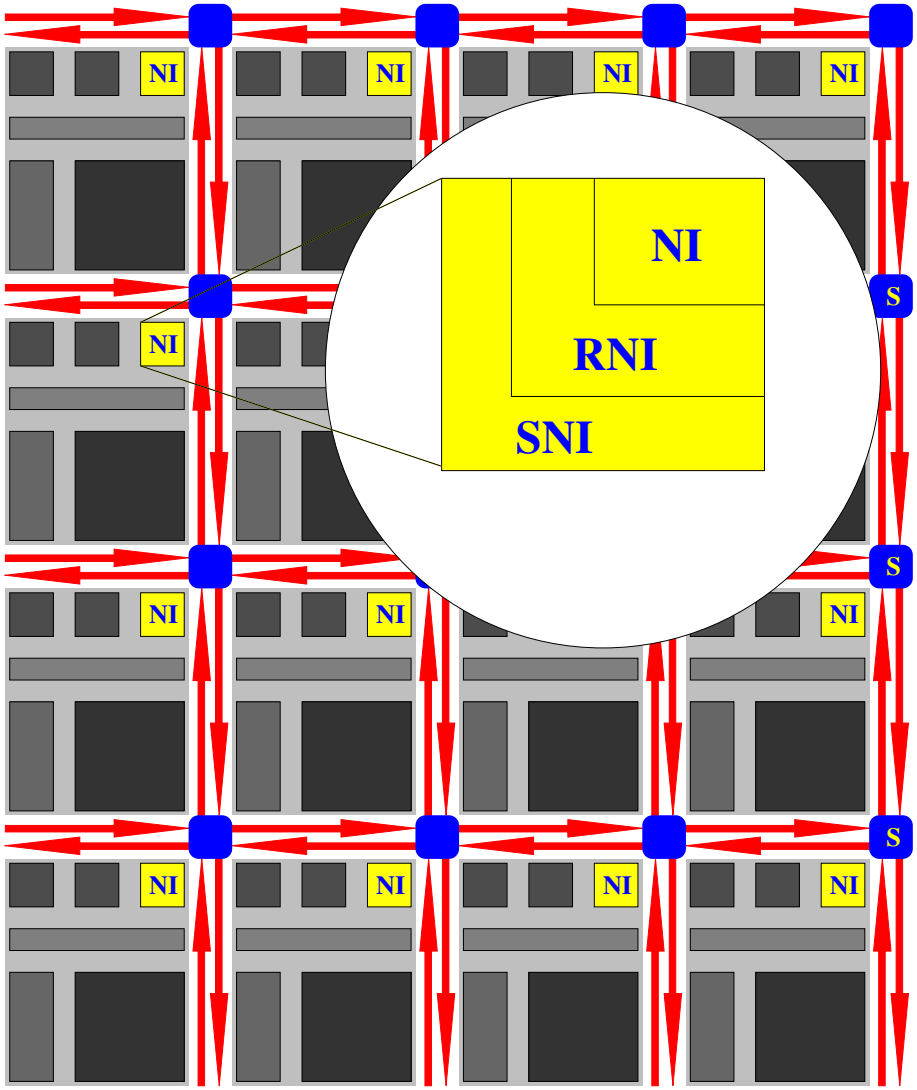
Motivation:

- Regularity of layout; predictable electrical properties
- Expected locality of traffic

The Node in a Mesh



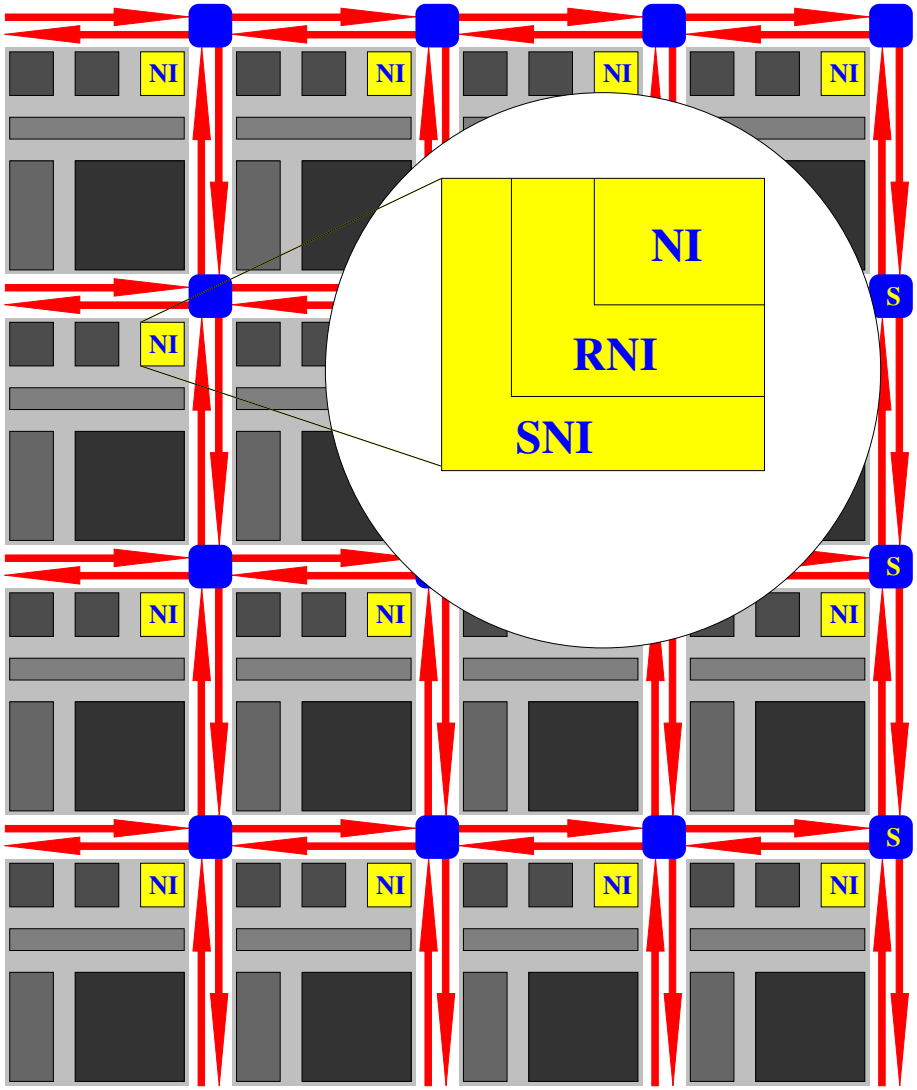
The Node in a Mesh



NI: Network Interface:

- Compulsory
- Hardware
- Implements the network layer protocol

The Node in a Mesh



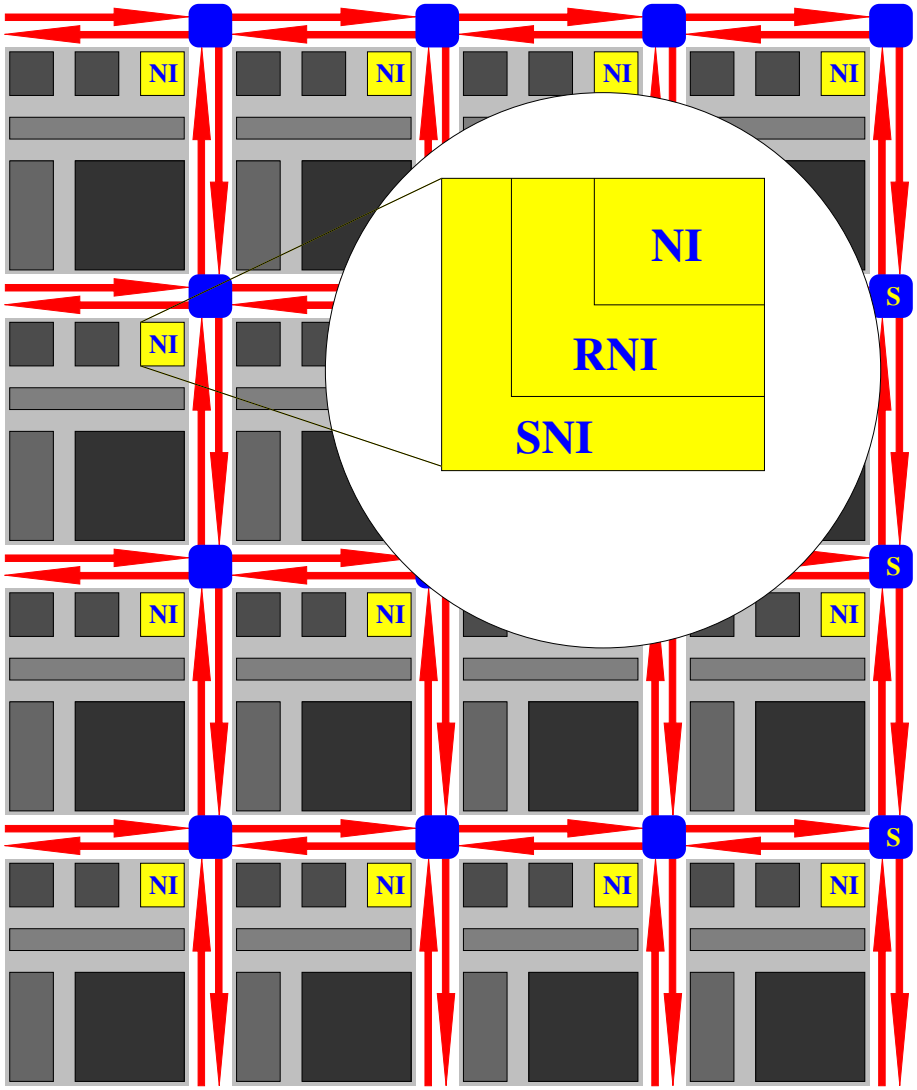
NI: Network Interface:

- Compulsory
- Hardware
- Implements the network layer protocol

RNI: Resource Network Interface:

- Optional
- Hardware and/or Software
- Implements transport layer
- Provides resource specific interfaces

The Node in a Mesh



NI: Network Interface:

- Compulsory
- Hardware
- Implements the network layer protocol

RNI: Resource Network Interface:

- Optional
- Hardware and/or Software
- Implements transport layer
- Provides resource specific interfaces

SLI: Session Layer Interface:

- Optional
- Hardware and/or software
- Implements the session layer protocol

Overview

Topology and Structure

The Network Layer and the Switch

TDM Allocation for Quality of Service

Regulated Flows

Data Protection

Clocking

Dynamic Voltage Scaling

Network Simulator

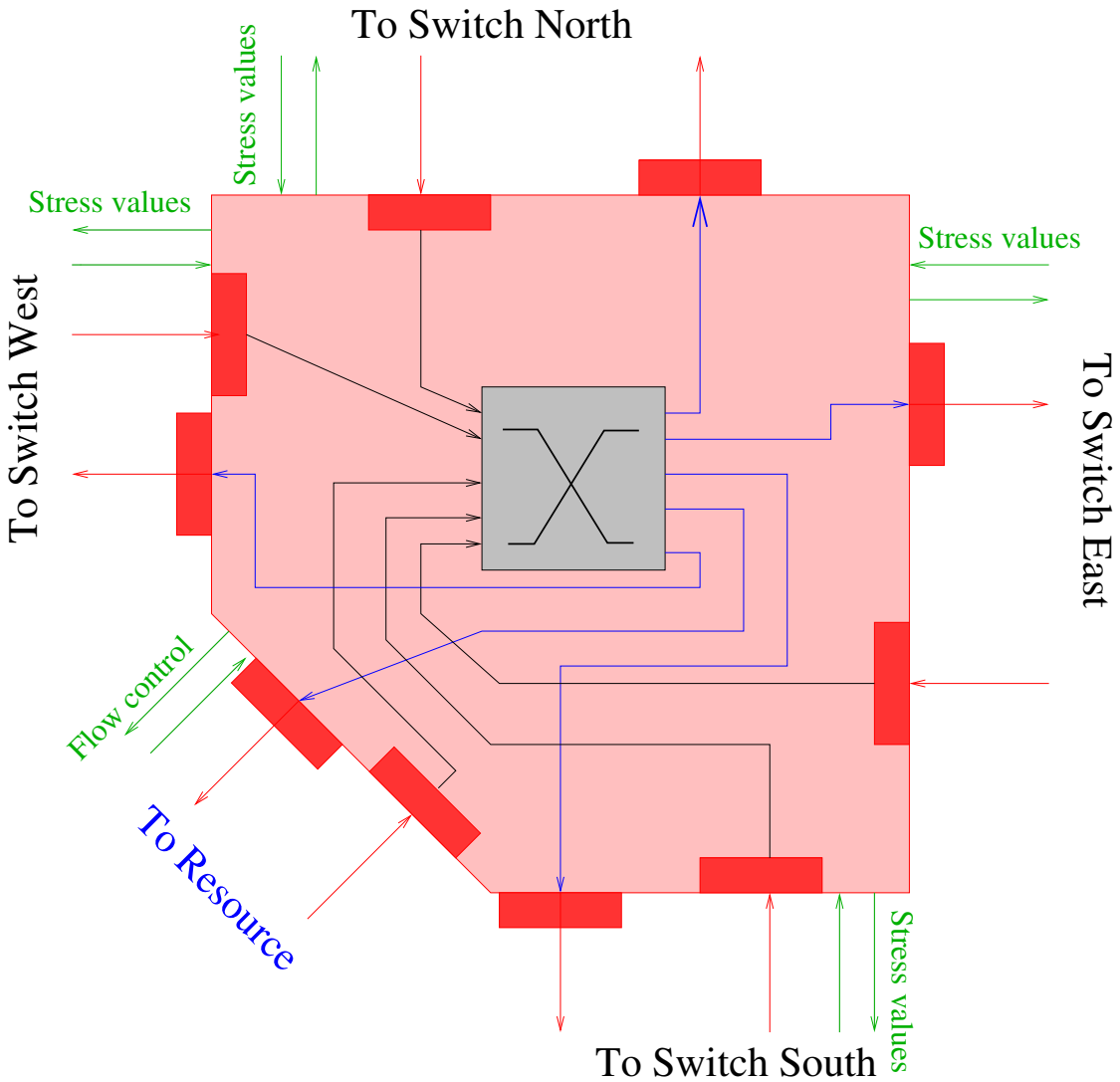
The Network Layer

- Packet switched best effort service
  - ⋆ Packets are guaranteed to arrive
  - ⋆ Packet payload may be protected (4 levels of protection)
  - ⋆ Load dependable delay in the network
  - ⋆ Load dependable delay at the network access point
  - ⋆ Admission policy for best effort traffic:
    - ∗ Network load should be below 60%
    - ∗ Load is measured locally in switch and based on neighboring stress values

The Network Layer

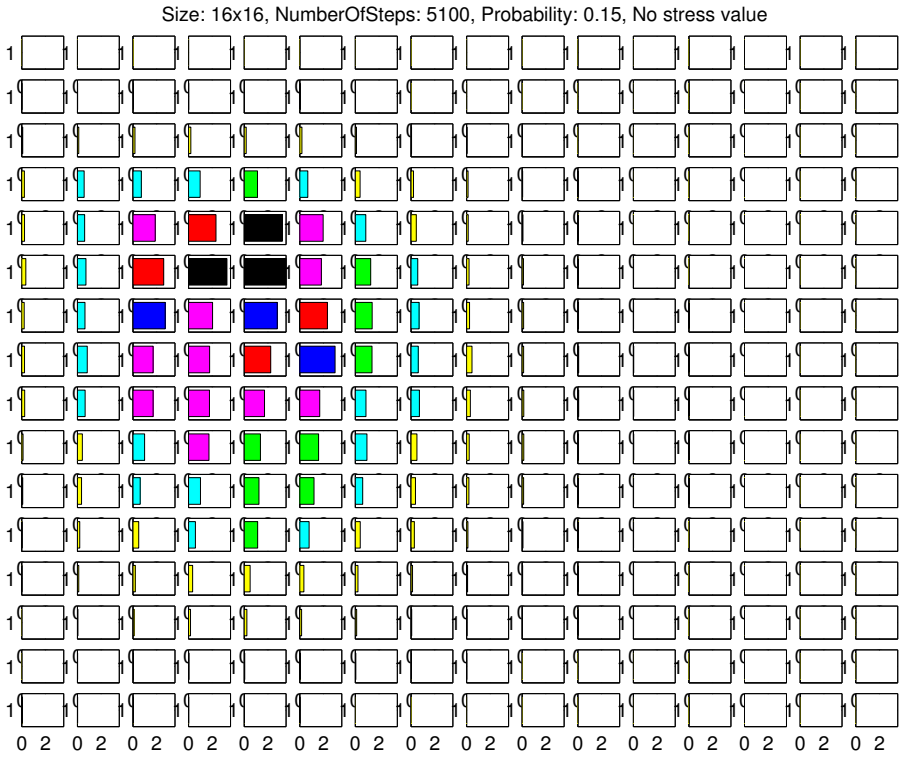
- Packet switched best effort service
  - ⋆ Packets are guaranteed to arrive
  - ⋆ Packet payload may be protected (4 levels of protection)
  - ⋆ Load dependable delay in the network
  - ⋆ Load dependable delay at the network access point
  - ⋆ Admission policy for best effort traffic:
    - ∗ Network load should be below 60%
    - ∗ Load is measured locally in switch and based on neighboring stress values
- Virtual circuit service
  - ⋆ Guaranteed bandwidth
  - ⋆ Guaranteed maximum delay
  - ⋆ Multicast circuits
  - ⋆ Static and semi-static virtual circuits
  - ⋆ Based on packet switching service

The Bufferless Switch



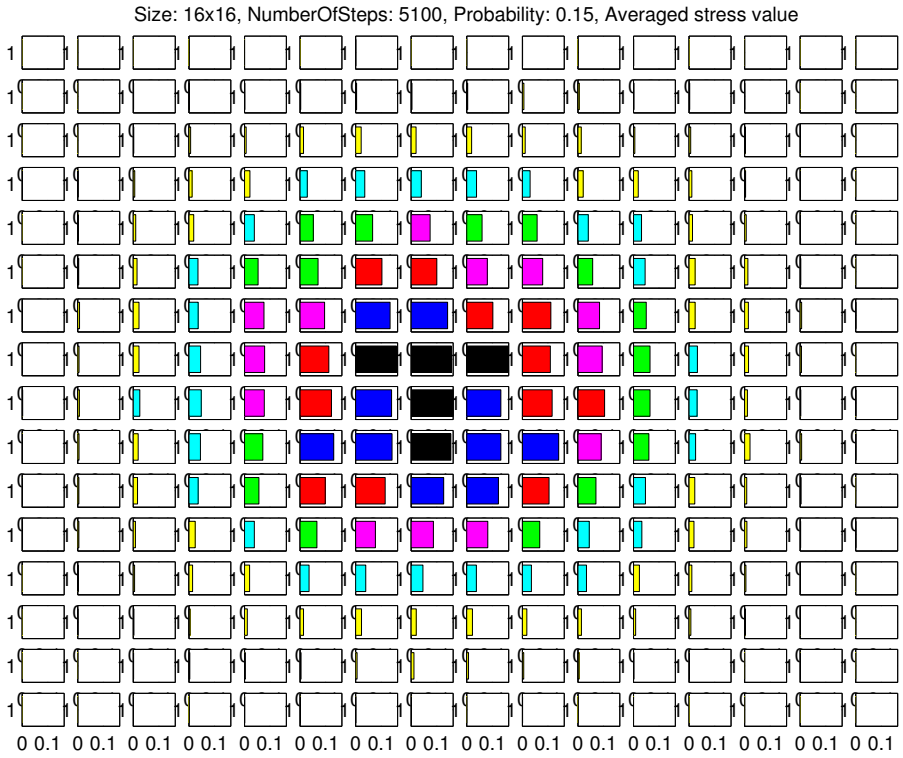
- \+ No buffers
- \+ No routing table
- \+ Small area
- \+ Short delay
- \+ Low power consumption
- – Non-shortest path
- – Header overhead due to destination address

Stress Value Effect on Buffer Sizes and Delays



No stress value control

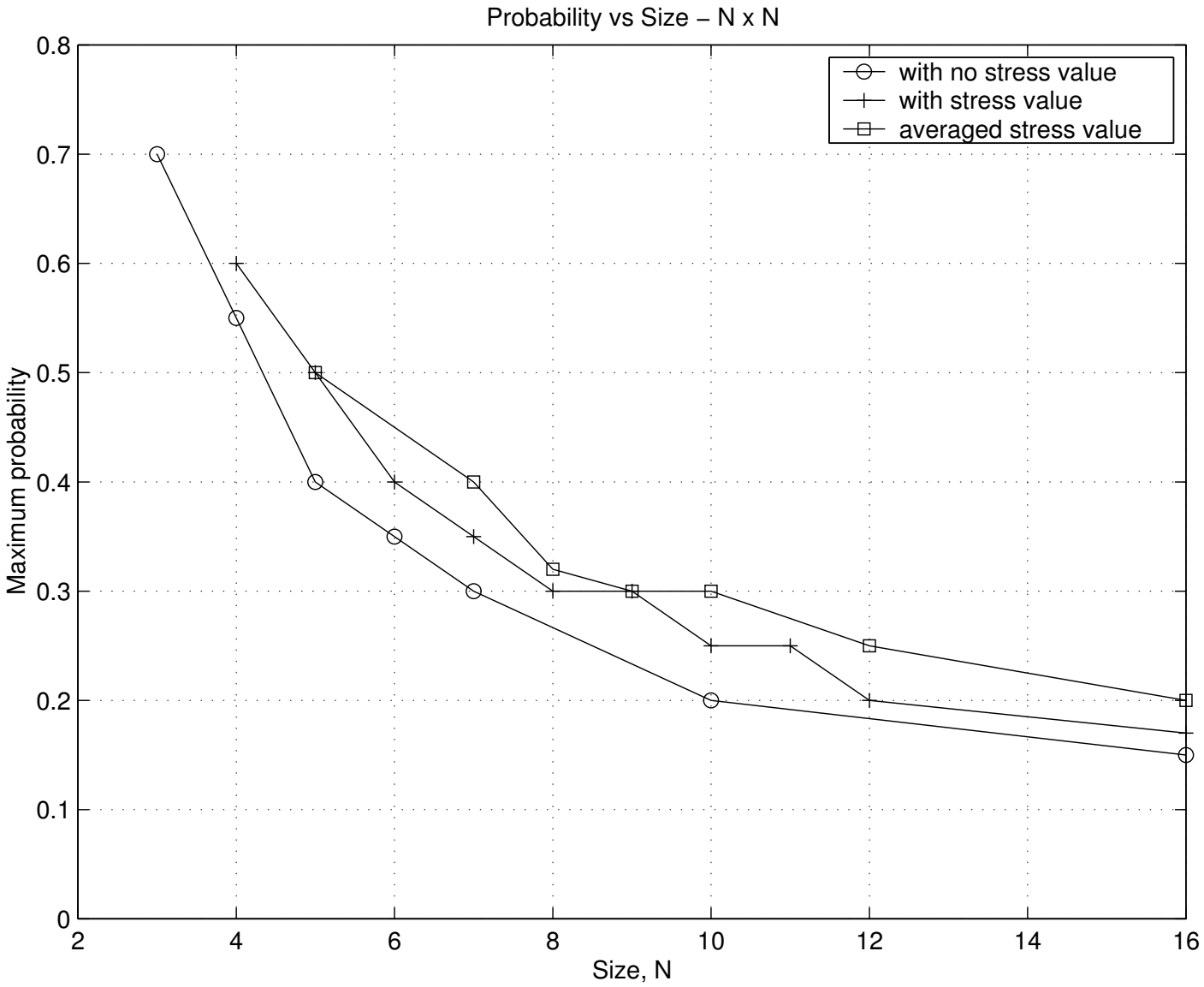
Largest average buffer size: 3.2 (black)



Averaged stress value control

Largest average buffer size: 0.2 (black)

Stress Value Effect on Maximum Load



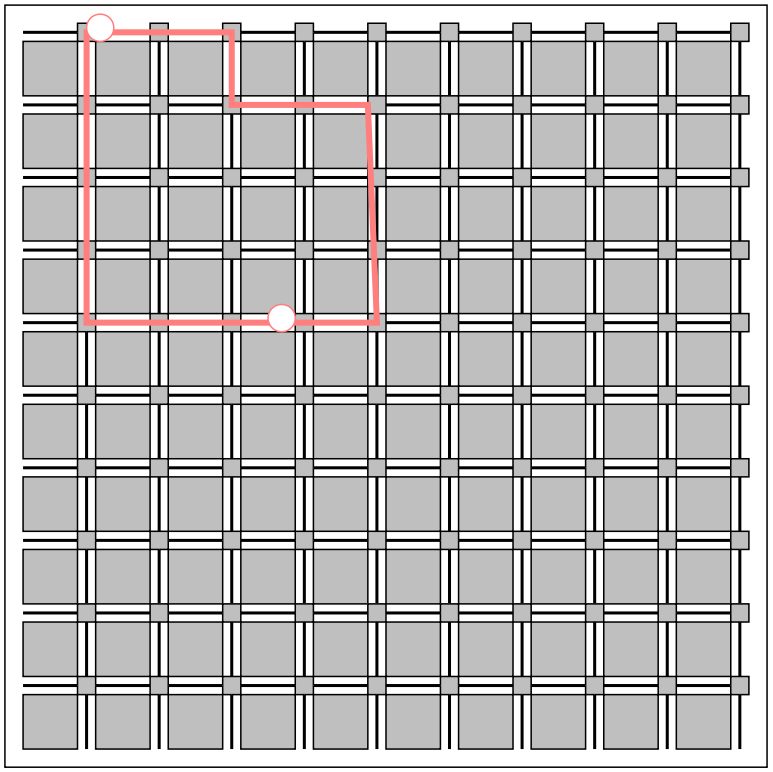
Looped Container based Virtual Circuit

- A container packet loops between two or more end points
- The looping container establishes a closed virtual circuit
- The virtual circuit allows multicast and bus protocol emulation
- Possible bandwidth allocation:

$$2^{j-d}B$$

where $B =$ link bandwidth, $d =$ length of the container loop, $1 \leq j \leq d$

- Examples:
  $d = 2$: possible allocations: 100% and 50%
  $d = 4$: possible allocations: 100%, 50%, 25%, 12.5%



Implementation of Static Virtual Circuits

- Bandwidth allocation and circuit setup at design time
- Implementation alternatives:
  - ⋆ Channel containers have higher priority
  - ⋆ Look-up tables in switches
- Semi-static circuits:
  - ⋆ Active circuits: Circulating containers
  - ⋆ Inactive circuits: Containers removed
  - ⋆ Activation of circuits subject to traffic load dependent delay
  - ⋆ NI can increase stress value to activate virtual circuits

Overview

Topology and Structure

The Network Layer and the Switch

TDM Allocation for Quality of Service

Regulated Flows

Data Protection

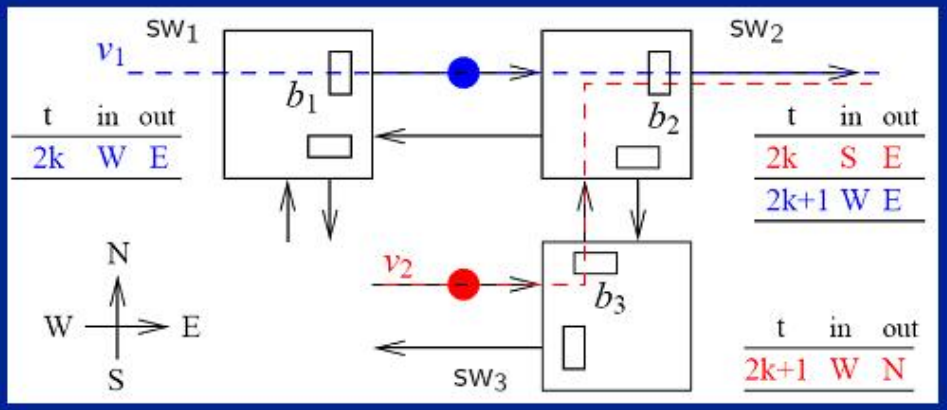
Clocking

Dynamic Voltage Scaling

Network Simulator

TDM Virtual Circuit

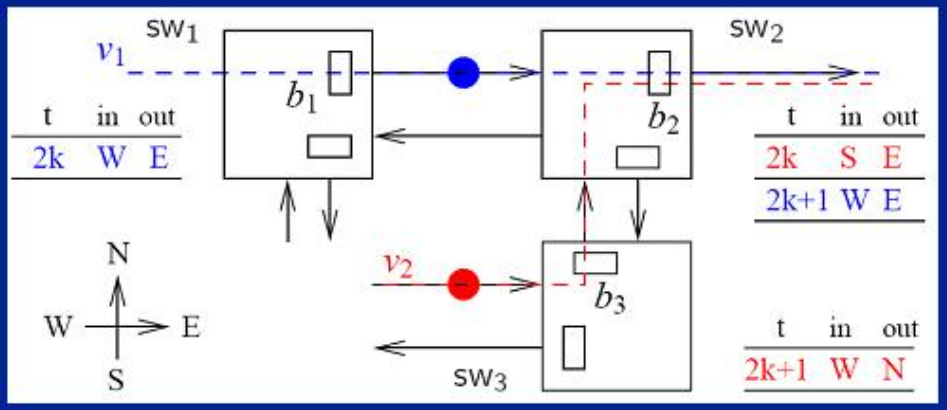
- What is that?
  - A VC is a connection in a packet-switched network.
  - A TDM VC means multiple connections use shared buffers and links in a time-division fashion.
- Example



- Why do we need it?
  - Contention-less, offering guaranteed latency and BW.

TDM Virtual Circuit

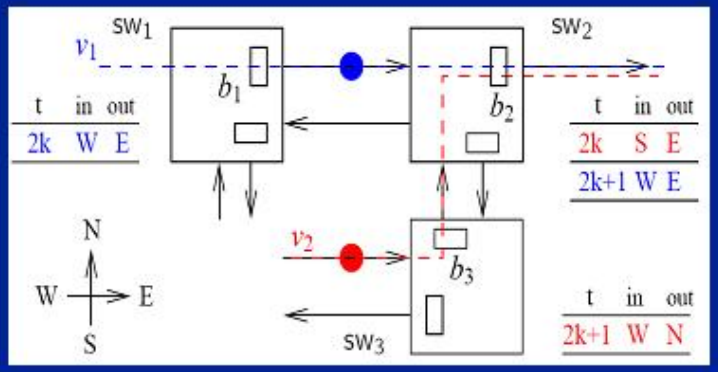
- What is that?
  - A VC is a connection in a packet-switched network.
  - A TDM VC means multiple connections use shared buffers and links in a time-division fashion.
- Example

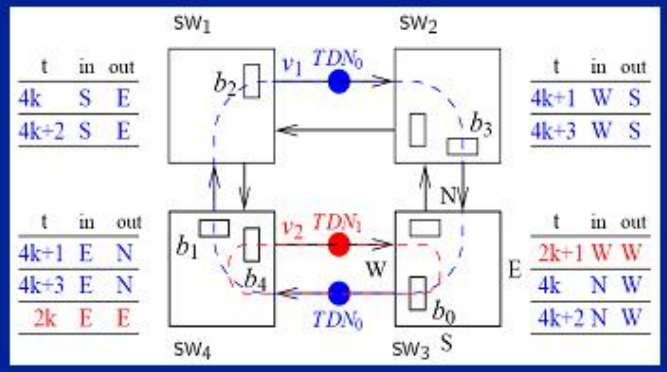


- Why do we need it?
  - Contention-less, offering guaranteed latency and BW.

Two variants of TDM VC

- Open-ended
  - A VC path is not a loop
  - for buffered flow control networks, e.g. wormhole, VCT
- Closed-loop
  - A VC is a loop
  - for buffer-less flow control networks, e.g. deflection

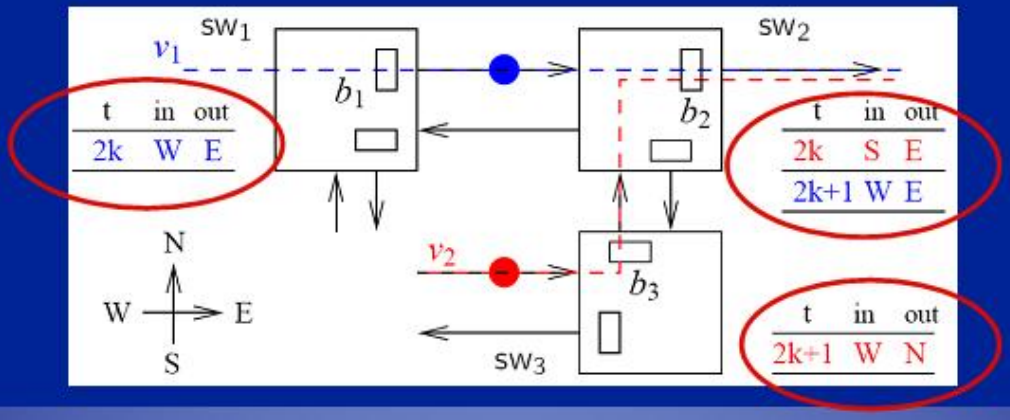




Containers are looped on VCs to carry data packets.

How to configure TDM VCs?

- Properties of a TDM VC
  - A deterministic path
  - Use dedicated time slots to pass buffers, freeing from contention
- Problems of TDM VC configuration
  - Path selection: explore network path diversity
  - Slot allocation: determine when (time slots) VC packets use buffers to be contention free and satisfy BW



Path Selection

- Multi node VC Configuration
- Node visiting order: Hamiltonian Path / Traveling Salesman problem
- Path selection + Slot Allocation by a Depth first search with backtracking
  - Ordering of VCs into a list
  - Finding shortest tour for multi-node comunication
  - Path selection
  - Slot allocation

On the slot allocation problem

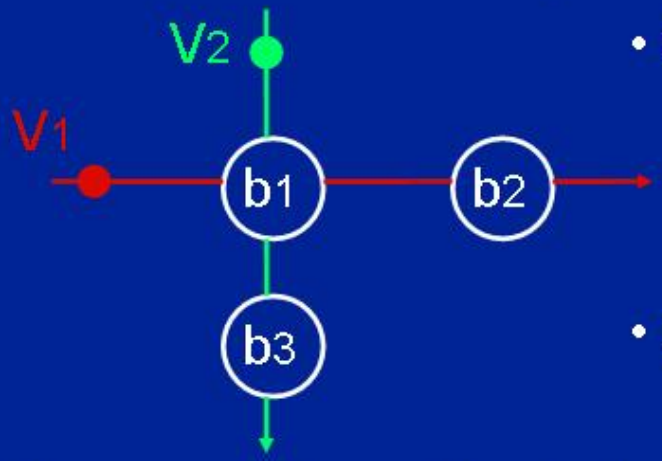
- How to avoid contention?
  - Use exclusive slots
  - Globally synchronize slot tables such that no simultaneous use of a buffer or link is possible
- How to guarantee bandwidth
  - In the first place, contention free
  - In the second place, allocate enough slots

Our approach and contributions

- Propose the concept of Logical Network (LN)
- Formulate a contention-less theory with necessary and sufficient conditions to assign VCs to LNs
- Develop a LN-based slot allocation algorithm

A running example

- To explain the concepts and method intuitively



• Given:

V1=<b1, b2>, BW1=1/2

V2=<b1, b3>, BW2=1/4

• Determine:

Packet admission pattern for v1 and v2 such that

(1) there is no contention and

(2) both BW requirements are met

Admission pattern: N packets are admitted over W slots

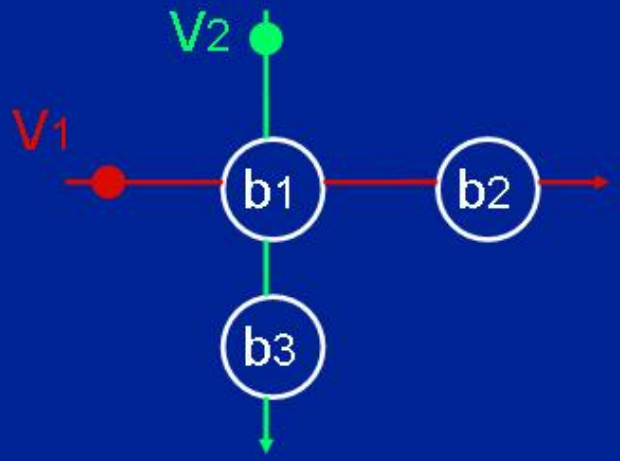
VC traffic flow: repeating this admission pattern

Avoid contention

- Two steps
  1. Slot partitioning with respect to a shared buffer in time domain
  2. Slot mapping along a VC path in space domain
- Consequence
  - Birth of LN, associated *(time slot, buffer)* pairs.
  - Eventually, we can precisely define traffic flow on VCs.

Avoid contention with the example

1. Slot partitioning
   - $V_1 \cap V_2 = \{b_1\}$
   - Even { (2k, b1)} and odd { (2k+1, b1) } slot sets
2. Slot mapping
   - Map b1's odd slot set on V_1, $LN_1^2(v_1,b_1) = \{ (2k+1, b_1), (2k,b_2) \}$
   - Map b1's even slot set on V_2, $LN_0^2(v_2,b_1) = \{ (2k, b_1), (2k+1,b_3) \}$

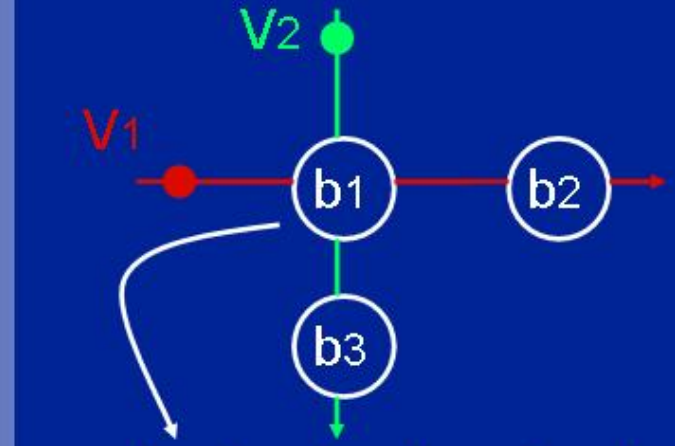


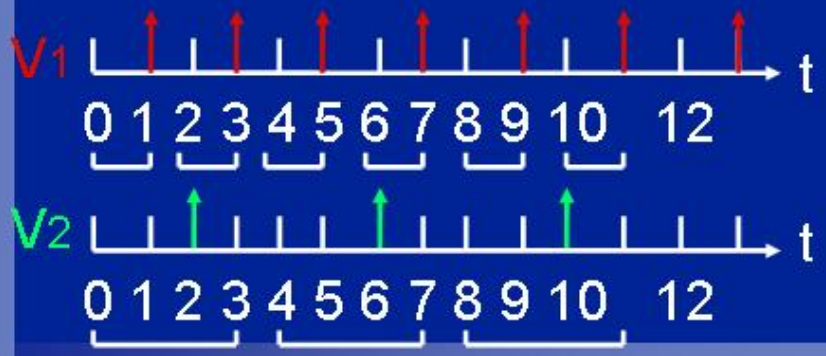
Satisfy bandwidth

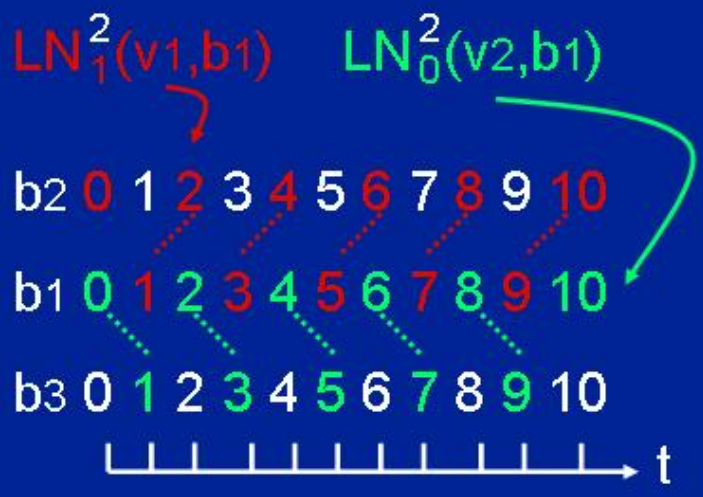
- A LN owns dedicated slots, thus BW.
- For each VC with its LNs, check supported_BW ≥ demanded_BW ?
  - If supported_BW > demanded_BW, do slot refinement, i.e., allocate/consume slots not more than necessary
  - If supported_BW = demanded_BW, consume slots
  - If supported_BW < demanded_BW, LNs are not sufficient to satisfy BW requirement

Satisfy bandwidth with the example

- V1: BW($LN_1^2(v1,b1)$)=1/2 =BW1
- V2: BW($LN_0^2(v2,b1)$)=1/2 >BW2







$BW(LN_0^2(v2,b1))=1/2 > 1/4$ Refine slots

Slot allocation summary

- The properties of a LN
  - Owns dedicated slots in buffers (*(slot, buffer)* pairs)
  - Function of VC and a reference buffer
  - One LN owns $1/N_{LN}$ bandwidth
- Slot allocation becomes VC-to-LN assignment:
  1. Slot partitioning to create LNs referring to a shared buffer
  2. Slot mapping to assign VCs to different LNs
  3. Slot refinement to allocate enough BW to LNs

How to generalize the results?

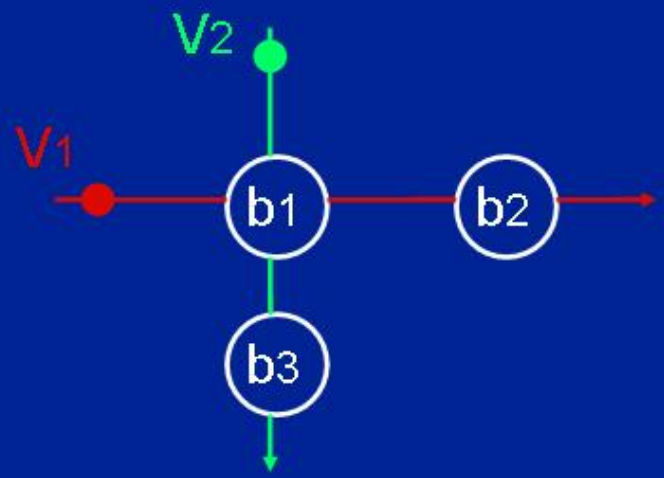
Essential issues

- How many LNs exist when VCs overlap?
  - How to partition slots?
- Is allocating VCs to different LNs sufficient and necessary?
- How to select a reference buffer when VCs have multiple shared buffers?
- Does the result change if a different reference buffer is selected?

Addressed by the contention-free theory

The number of LNs

- Two VCs, V_1 and V_2, with admission window W_1 and W_2, the max. number of LNs, $N_{LN}(V_1, V_2)=GCD(W_1, W_2)$
  - W_1 and W_2 are derivable from BW requirement and subject to application constraints
  - More than one solution, reflecting design space
- Example



V_1=<b1, b2>, BW_1=1/2

V_2=<b1, b3>, BW_2=1/4

⇩

W_1=2, W_2=4

$N_{LN}(V_1, V_2)=GCD(2, 4)=2$

⇩

b1 0 1 2 3 4 5 6 7 8 9 10

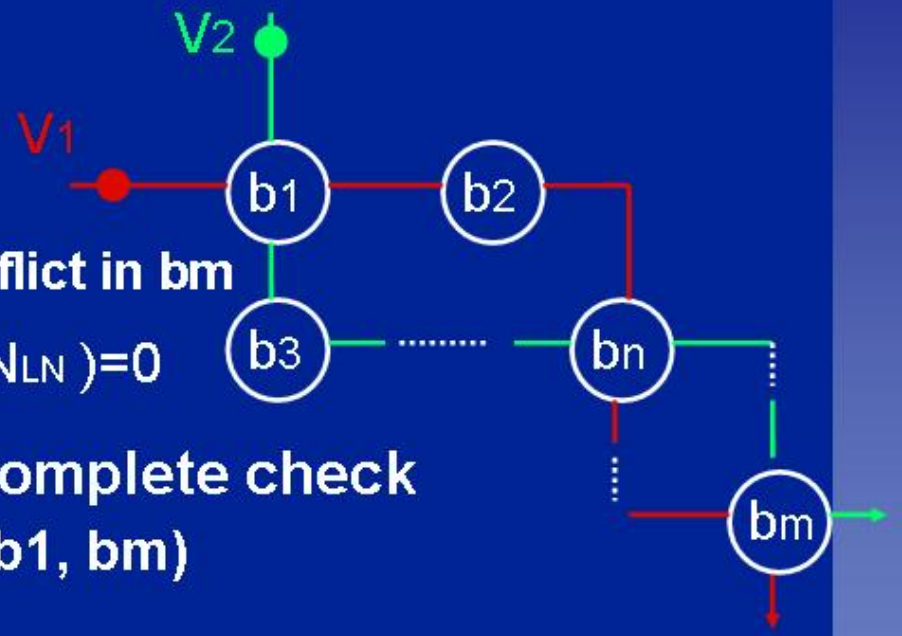
Sufficient and necessary condition

- VC-to-LN assignment steps:
  1. Slot partitioning to create LNs with respect to a reference buffer
  2. Slot mapping to assign VCs to different LNs
  3. Slot refinement
- Assigning VCs to different LNs is sufficient and necessary to promise contention-free.

Multiple shared buffers

- If two VCs have multiple shared buffers, how to select the reference buffer?
- Example
  - $V1 \cap V2 = \{b1, bn, bm\}$
- Consistency check
  - **No conflict in bn => No conflict in bm**

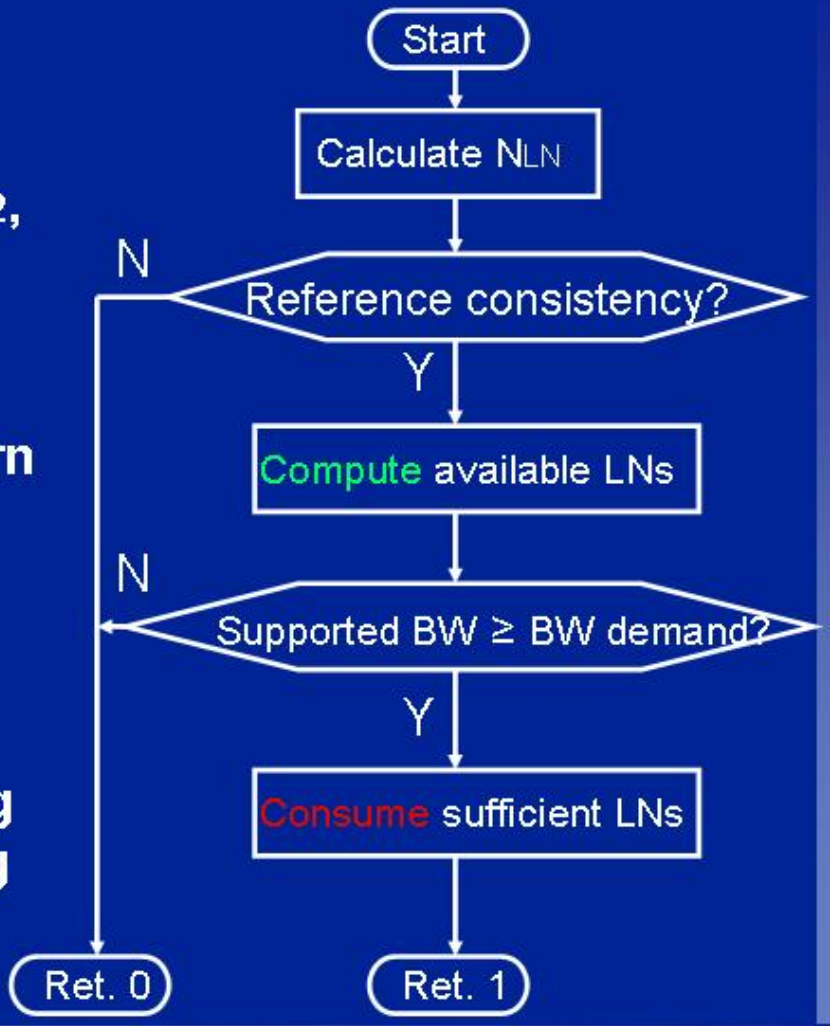
$$mod(d_{bnbm}(V1) - d_{bnbm}(V2), N_{LN})=0$$



- Linear check instead of complete check
  - **(b1,bn) and (bn, bm) => (b1, bm)**

LN-oriented slot allocation

- Input
  - VC spec. set, V_1, V_2, … Vn, with BW and path known
  - Admission windows, W_1, W_2, …, Wn, respectively
- Output
  - Fail or succeed
  - If succeed, admission pattern for each VC
- Slot allocation procedure
  - Pair-wise (v_i, v_j) and incremental
  - Compute LNs: slot partitioning and LN mapping
  - Consume LNs: slot mapping and refinement



VC configuration program

- The LN-based slot allocation method has been implemented in our VC configuration program
- The VC configuration program
  - supports both open-ended and closed-loop VCs
  - explores the network path diversity via back-tracking

Overview

Topology and Structure

The Network Layer and the Switch

TDM Allocation for Quality of Service

Contracts based on Regulated Flows

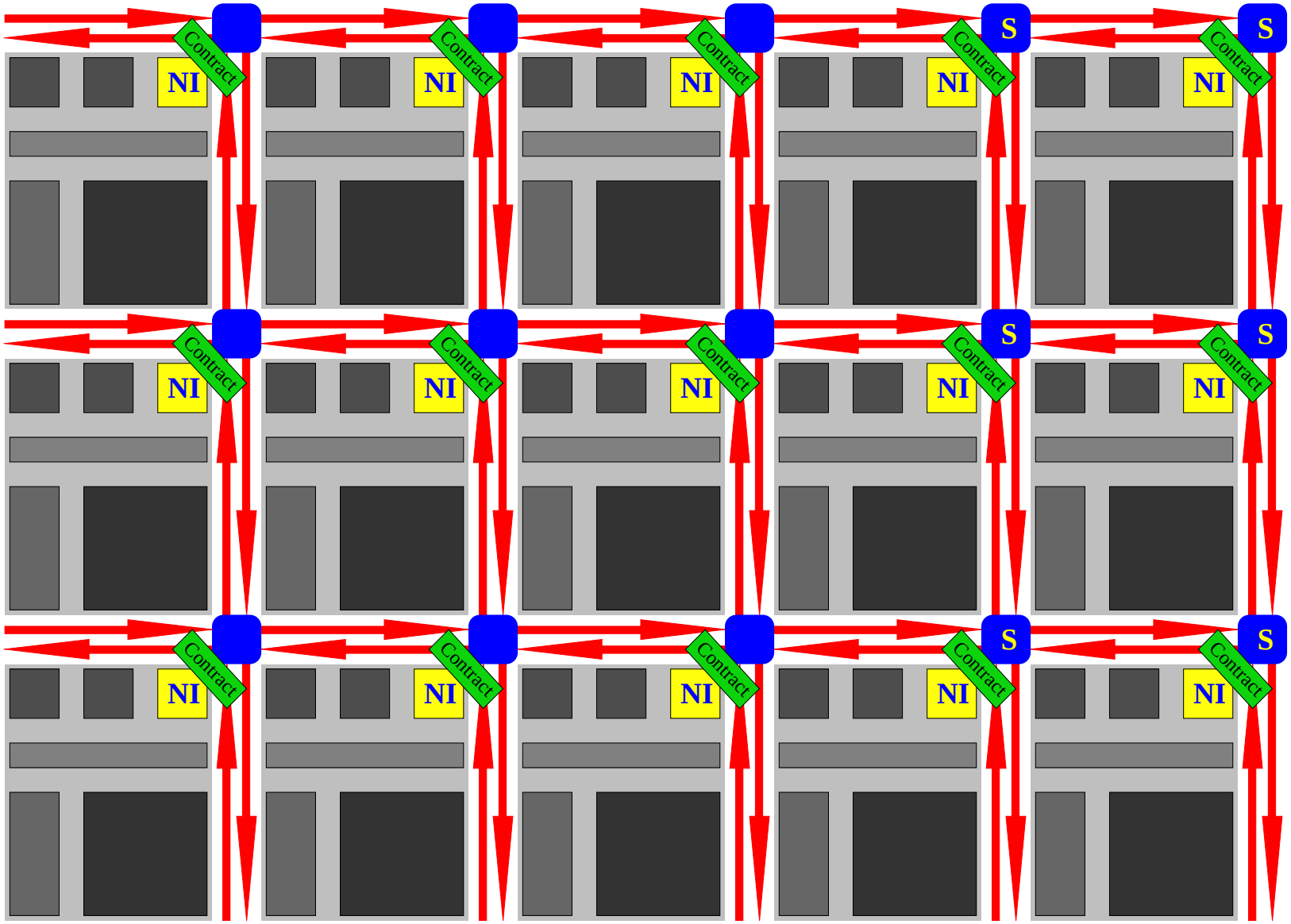
Data Protection

Clocking

Dynamic Voltage Scaling

Network Simulator

Contract based Flows



Regulated Flows

A Flow F is (σ, ρ) regulated if

$$F(b) - F(a) \leq \sigma + \rho(b - a)$$

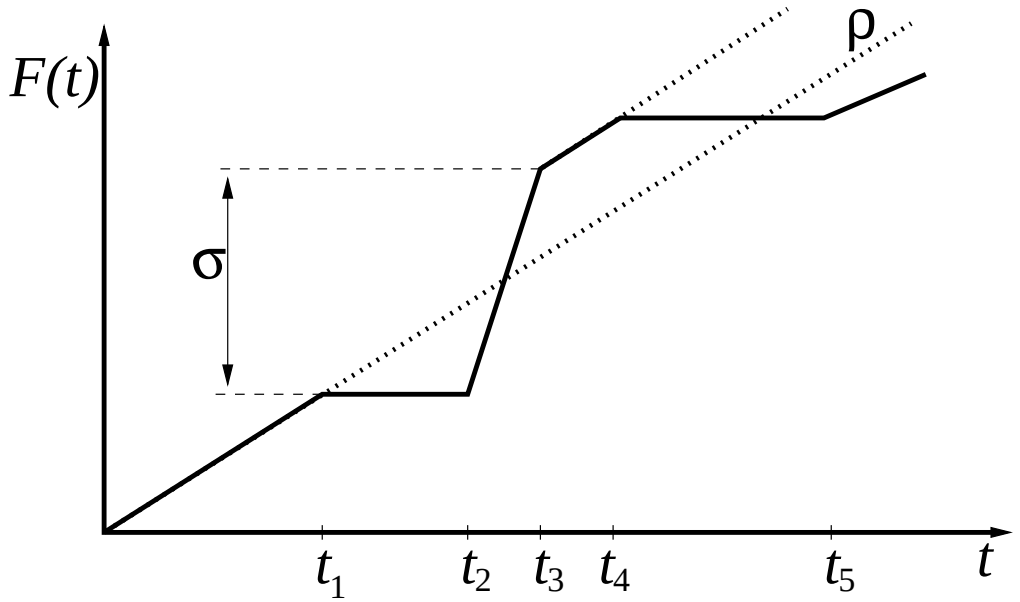
for all time intervals $[a, b], 0 \leq a \leq b$ and where
$F(t) \cdots$ the cumulative amount of traffic between 0 and $t \geq 0$.
$\sigma \geq 0$ is the burstiness constraint;
$\rho \geq 0$ is the maximum average rate;

Regulated Flows

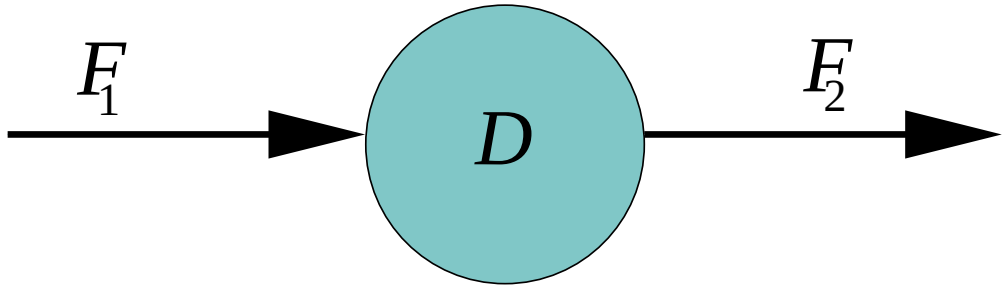
A Flow F is (σ, ρ) regulated if

$$F(b) - F(a) \leq \sigma + \rho(b - a)$$

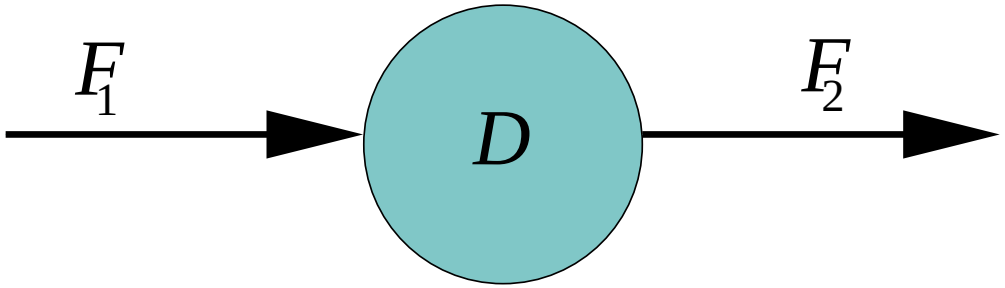
for all time intervals $[a, b], 0 \leq a \leq b$ and where
$F(t) \cdots$ the cumulative amount of traffic between 0 and $t \geq 0$.
$\sigma \geq 0$ is the burstiness constraint;
$\rho \geq 0$ is the maximum average rate;



Regulated Flows - Delay Element



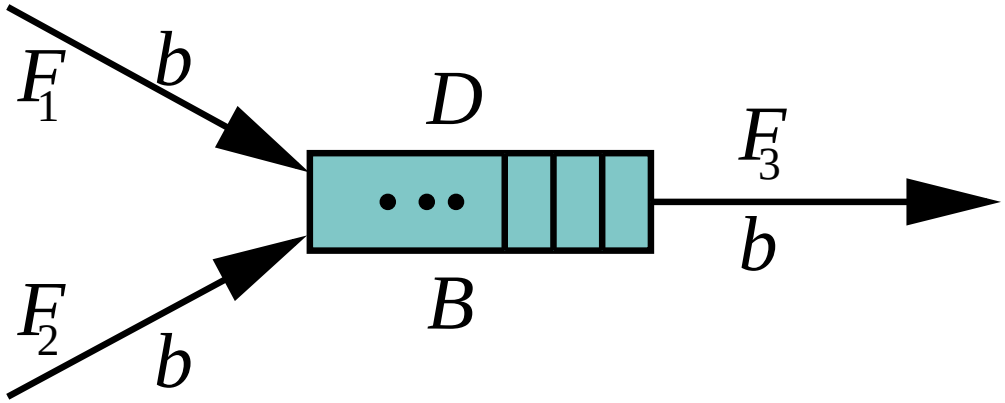
Regulated Flows - Delay Element



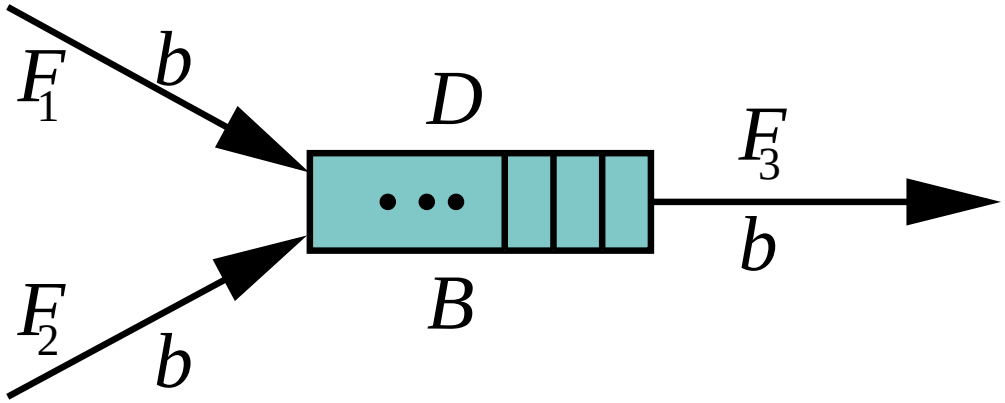
$$F_1 \sim (\sigma, \rho)$$

$$F_2 \sim (\sigma + \rho D, \rho)$$

Regulated Flows - Work Conserving Multiplexer

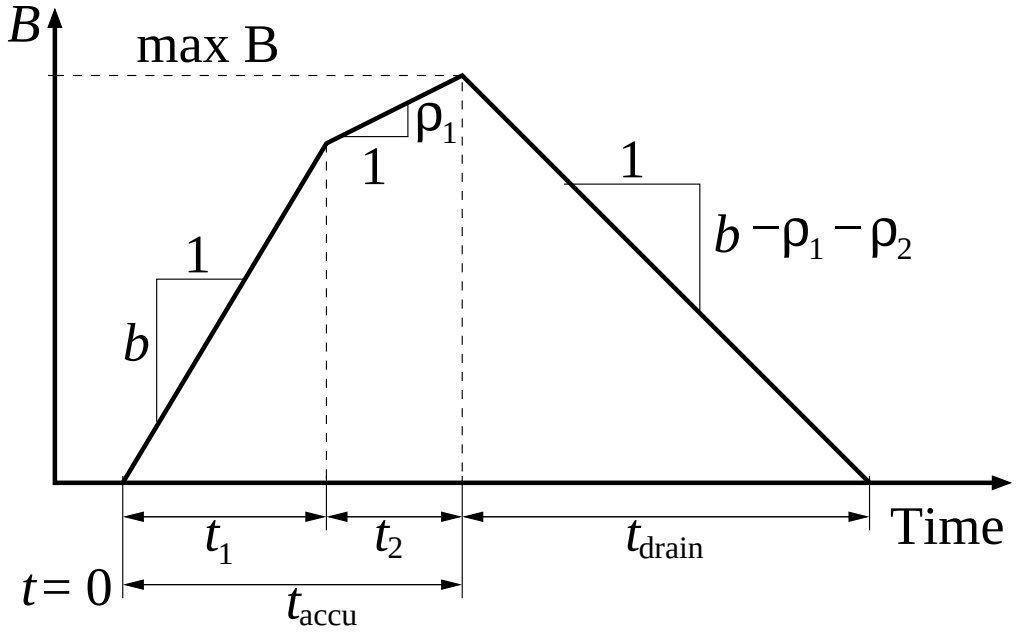


Regulated Flows - Work Conserving Multiplexer

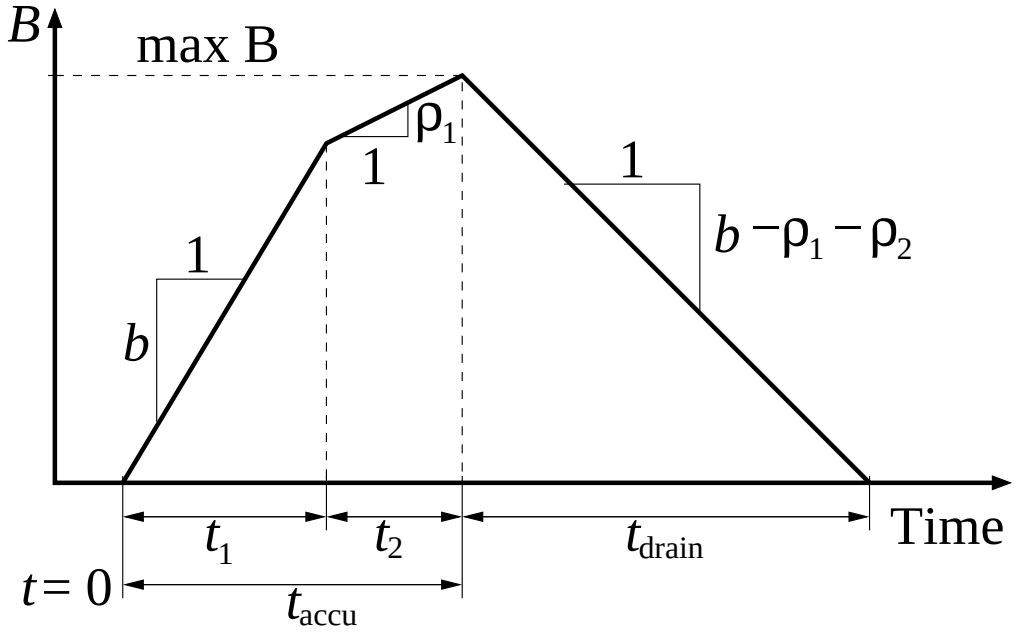


$$
\begin{aligned}
F_1 &\sim (\sigma_1, \rho_1) \\
F_2 &\sim (\sigma_2, \rho_2) \\
\text{link bandwidth } b &< \rho_1 + \rho_2 \\
F_3 &\sim ? \\
\text{maximum delay } D &= ? \\
\text{maximum backlog } B &= ?
\end{aligned}
$$

Work Conserving Multiplexer - 1

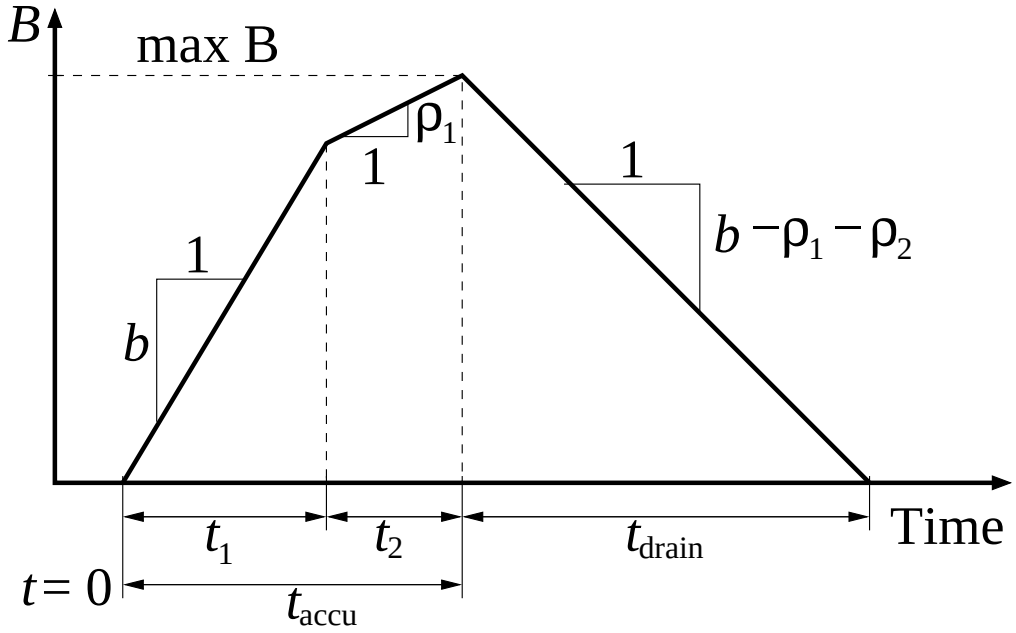


Work Conserving Multiplexer - 1



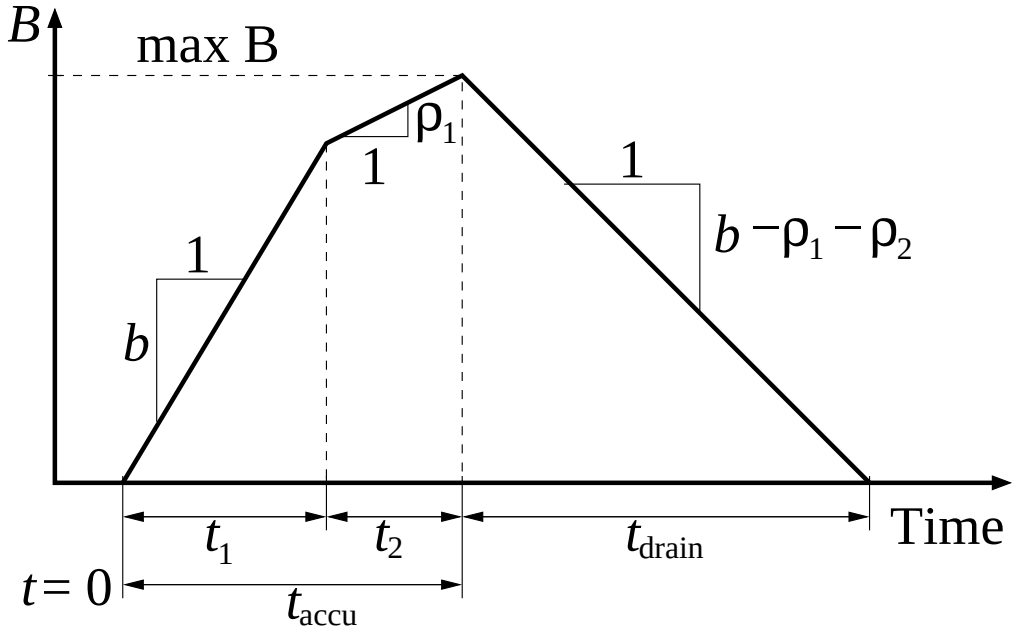
Phase 1 (t_1): F_1 and F_2 transmit at full speed;

Work Conserving Multiplexer - 1



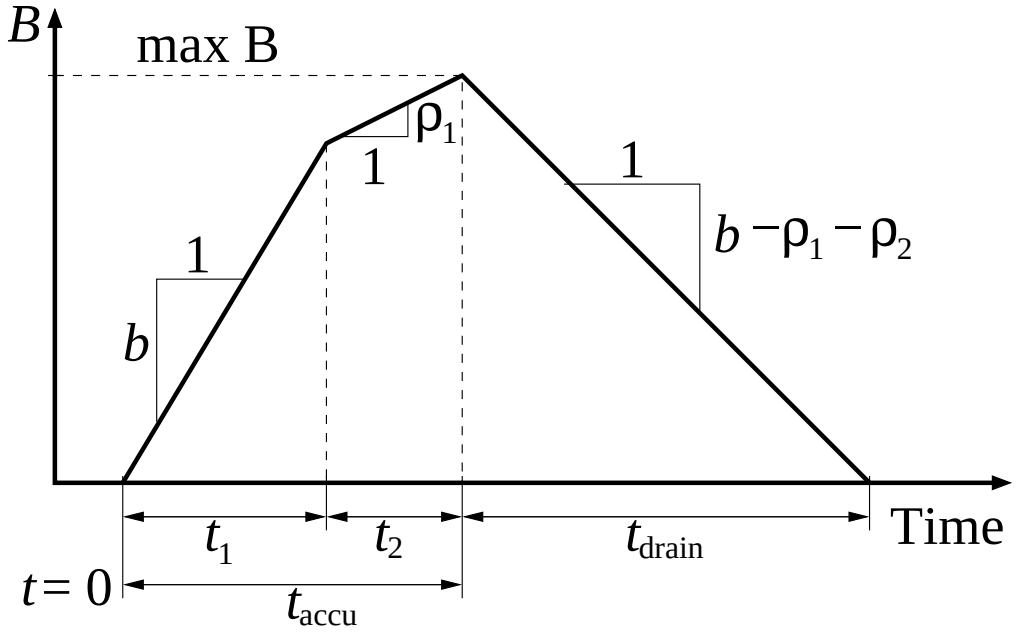
Phase 1 (t_1): F_1 and F_2 transmit at full speed;
Assume: At $t = 0$ the queue is empty; $\sigma_1 \leq \sigma_2$

Work Conserving Multiplexer - 1



Phase 1 (t_1): F_1 and F_2 transmit at full speed;
Assume: At $t = 0$ the queue is empty; $\sigma_1 \leq \sigma_2$
Injection rate: $2b$; Drain rate: b

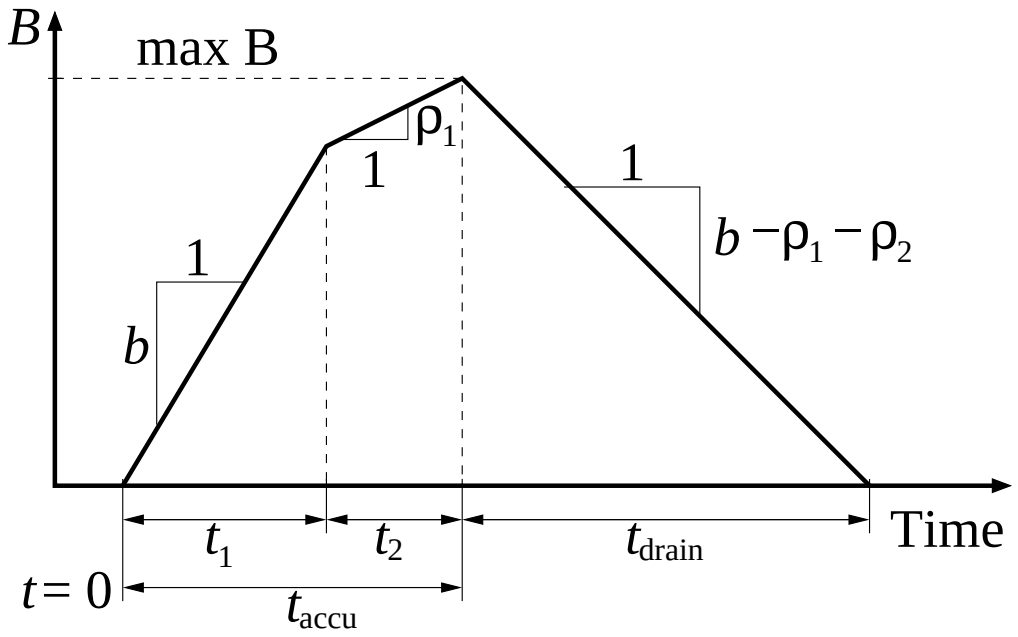
Work Conserving Multiplexer - 1



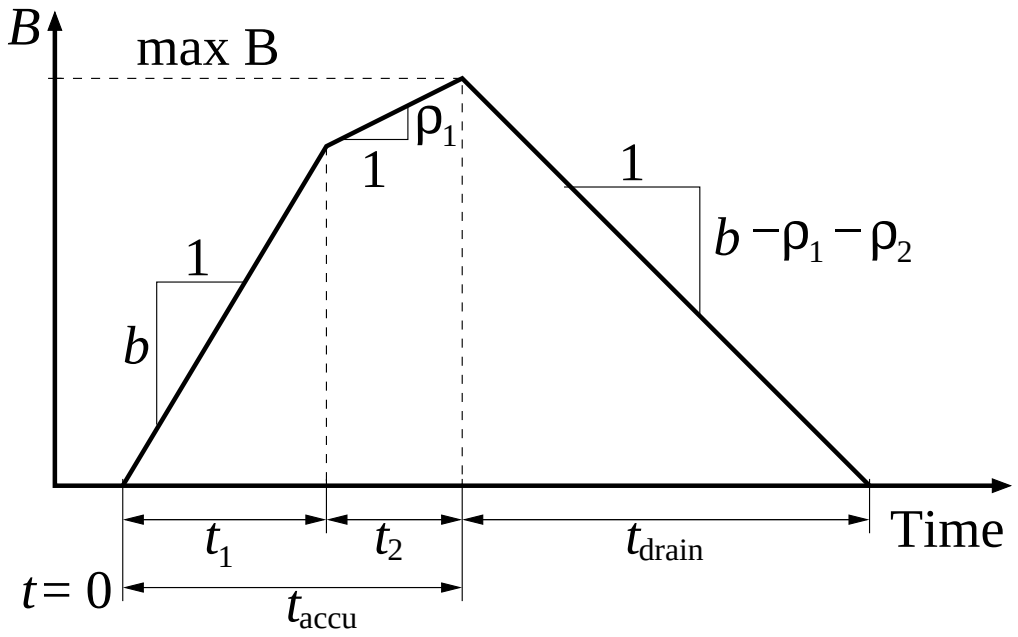
Phase 1 (t_1): F_1 and F_2 transmit at full speed;
Assume: At $t = 0$ the queue is empty; $\sigma_1 \leq \sigma_2$
Injection rate: $2b$; Drain rate: b

$$
\begin{aligned}
bt_1 &= \sigma_1 + \rho_1 t_1 \\
t_1 &= \frac{\sigma_1}{b - \rho_1}
\end{aligned}
$$

Work Conserving Multiplexer - 2

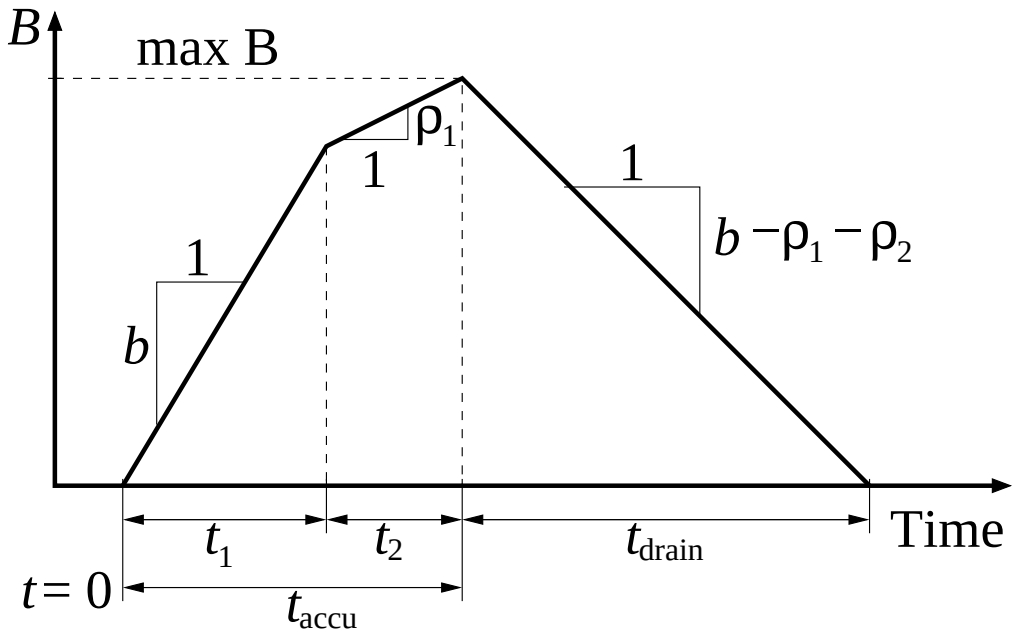


Work Conserving Multiplexer - 2



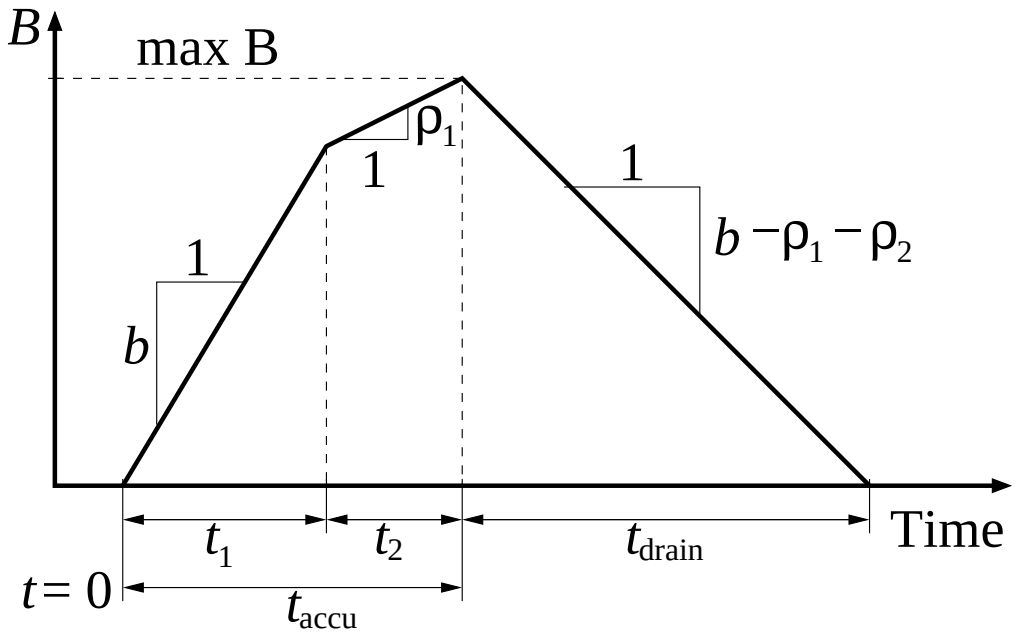
Phase 2 (t_2): F_1 transmits at rate ρ_1, F_2 transmits at full speed;

Work Conserving Multiplexer - 2



Phase 2 (t_2): F_1 transmits at rate ρ_1, F_2 transmits at full speed;
Injection rate: $b + \rho_1$; Drain rate: b

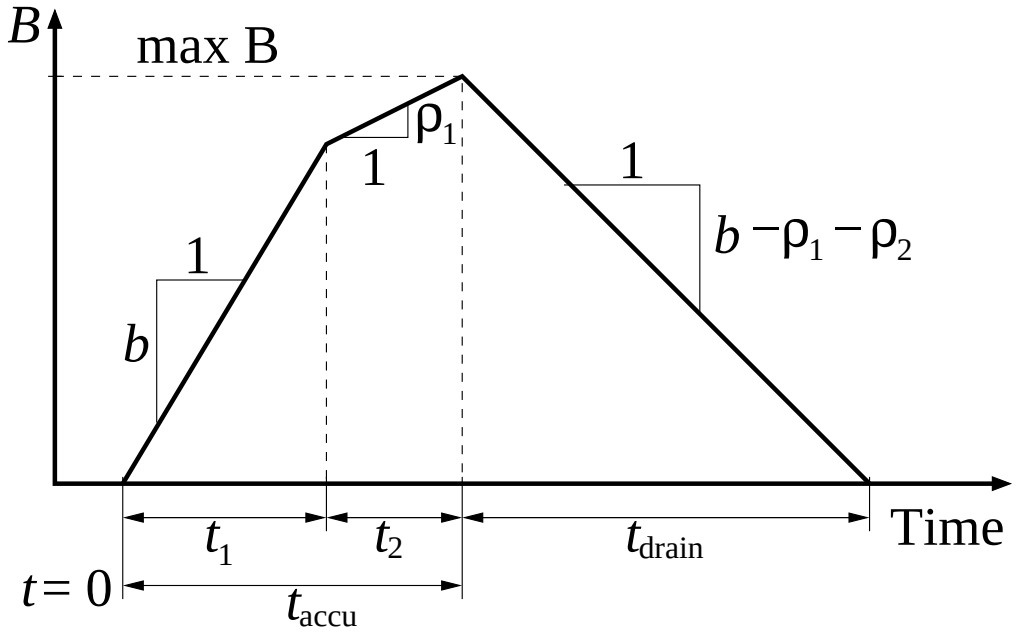
Work Conserving Multiplexer - 2



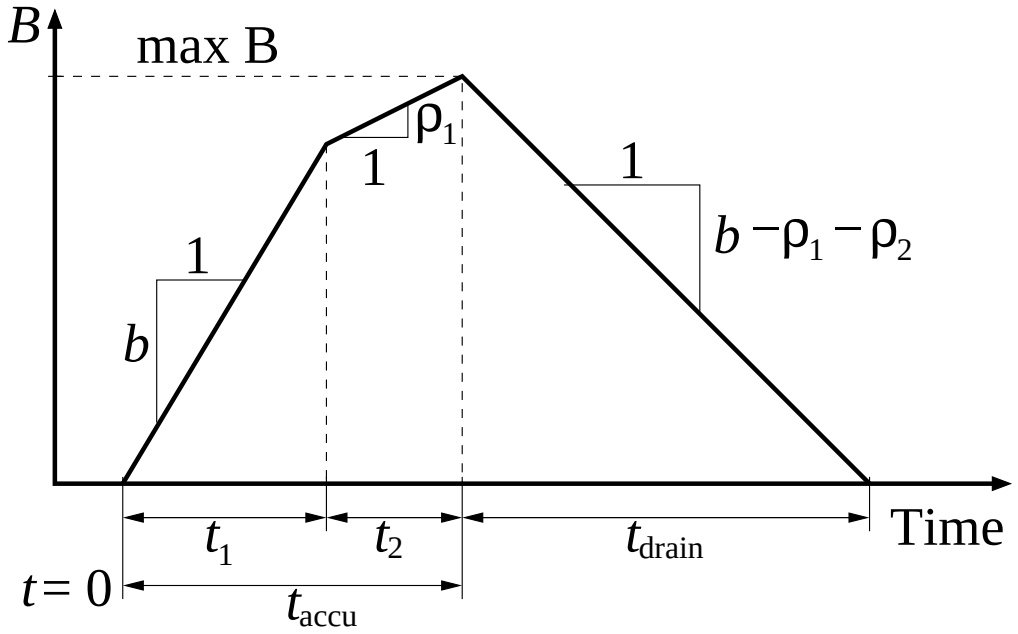
Phase 2 (t_2): F_1 transmits at rate ρ_1, F_2 transmits at full speed; Injection rate: $b + \rho_1$; Drain rate: b

$$
\begin{aligned}
b t_{\text{accu}} &= \sigma_2 + \rho_2 t_{\text{accu}} \\
t_{\text{accu}} &= \frac{\sigma_2}{b - \rho_2}
\end{aligned}
$$

Work Conserving Multiplexer - 3

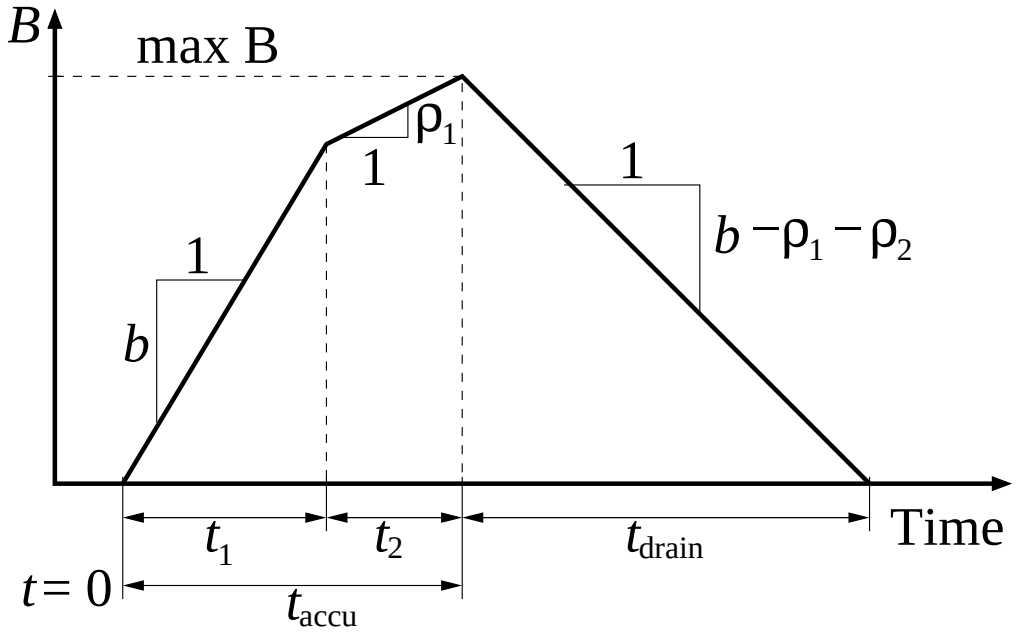


Work Conserving Multiplexer - 3



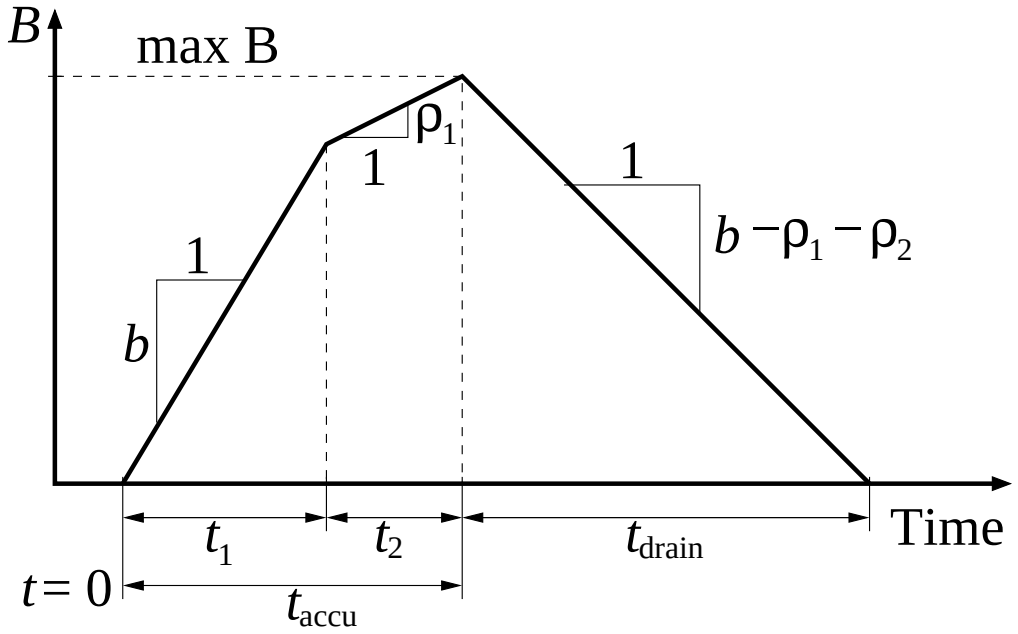
Phase 3 (t_{drain}): F_1 transmits at rate ρ_1, F_2 transmits at rate ρ_2;

Work Conserving Multiplexer - 3



Phase 3 (t_{drain}): F_1 transmits at rate ρ_1, F_2 transmits at rate ρ_2; Injection rate: $\rho_1 + \rho_2$; Drain rate: b

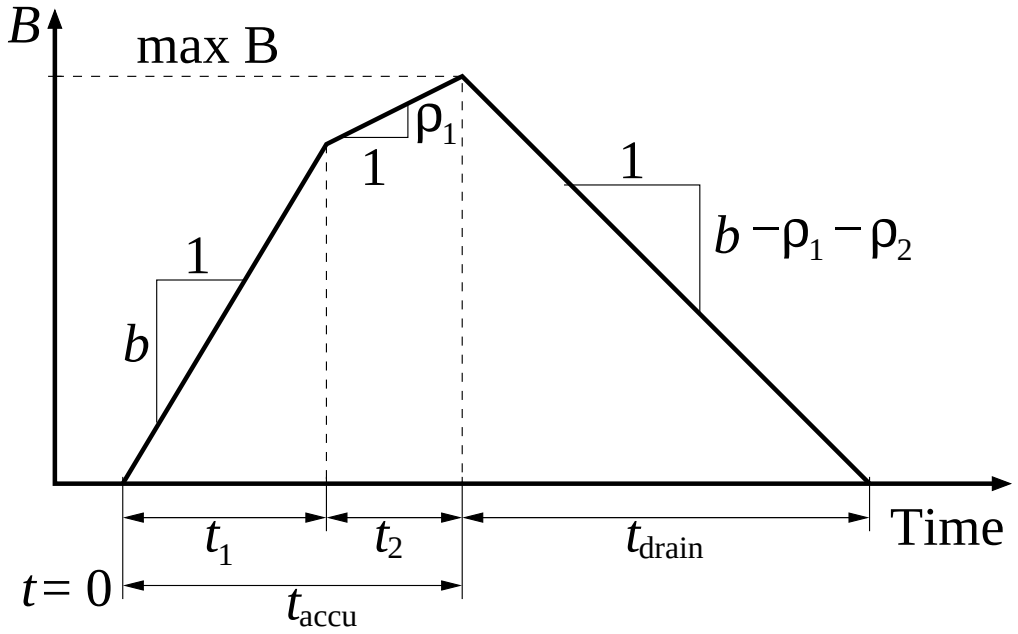
Work Conserving Multiplexer - 3



Phase 3 (t_{drain}): F_1 transmits at rate ρ_1, F_2 transmits at rate ρ_2; Injection rate: $\rho_1 + \rho_2$; Drain rate: b

$$t_{\mathrm{drain}} = \frac{B_{\max}}{b - \rho_1 - \rho_2}$$

Work Conserving Multiplexer - 3

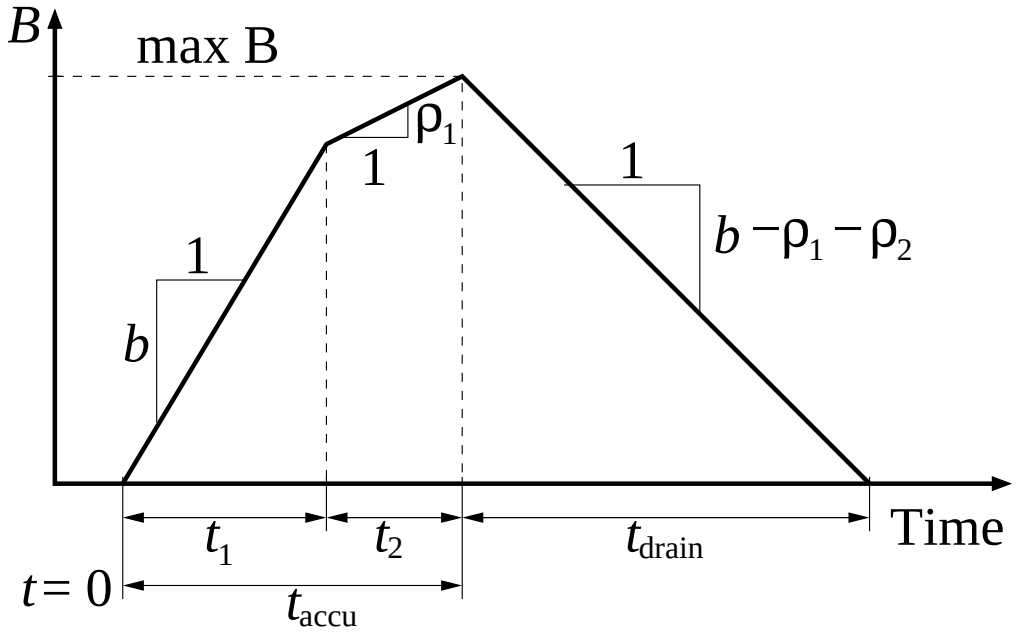


Phase 3 (t_{drain}): F_1 transmits at rate ρ_1, F_2 transmits at rate ρ_2; Injection rate: $\rho_1 + \rho_2$; Drain rate: b

$$t_{\text{drain}} = \frac{B_{\max}}{b - \rho_1 - \rho_2}$$

$$B_{\max} = bt_1 + \rho_1 t_2$$

Work Conserving Multiplexer - 3

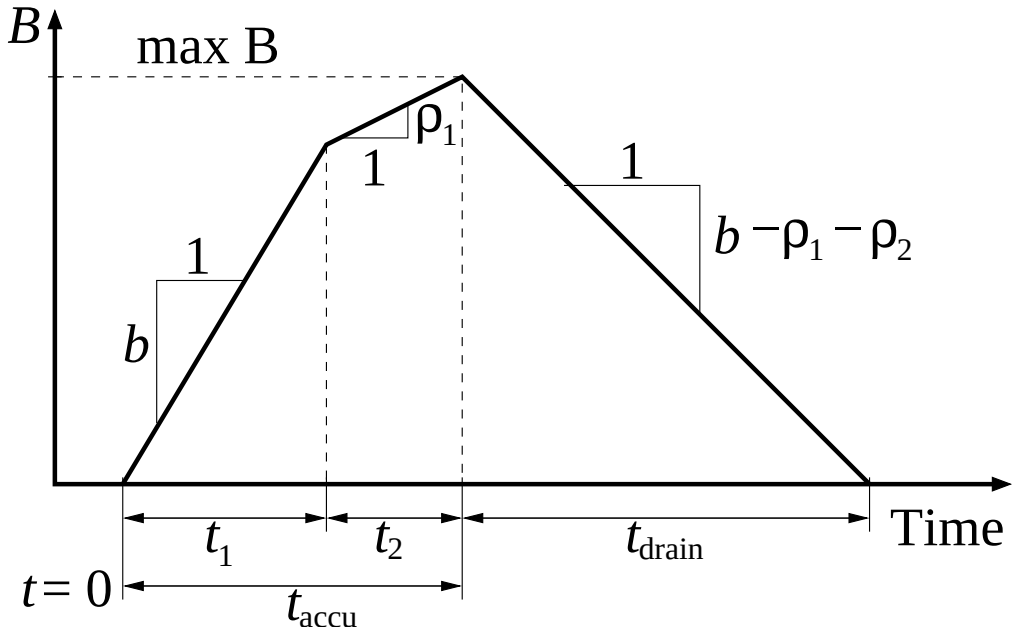


Phase 3 (t_{drain}): F_1 transmits at rate ρ_1, F_2 transmits at rate ρ_2; Injection rate: $\rho_1 + \rho_2$; Drain rate: b

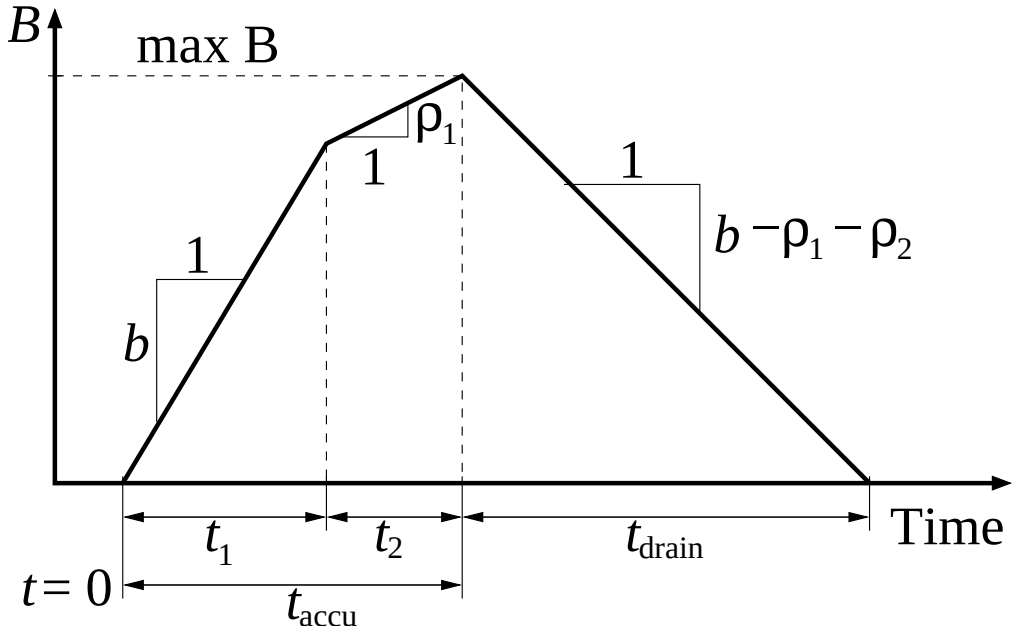
$$t_{\text{drain}} = \frac{B_{\max}}{b - \rho_1 - \rho_2}$$

$$B_{\max} = bt_1 + \rho_1 t_2 = \sigma_1 + \frac{\rho_1 \sigma_2}{b - \rho_2}$$

Work Conserving Multiplexer - Summary

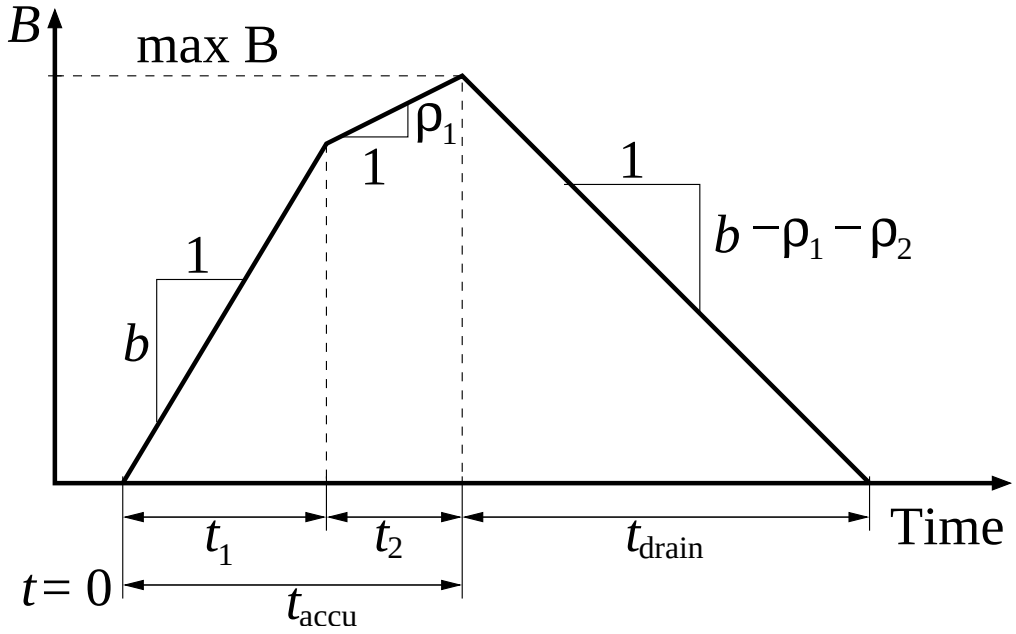


Work Conserving Multiplexer - Summary



$$B_{\max} = \sigma_1 + \frac{\rho_1 \sigma_2}{b - \rho_2}$$

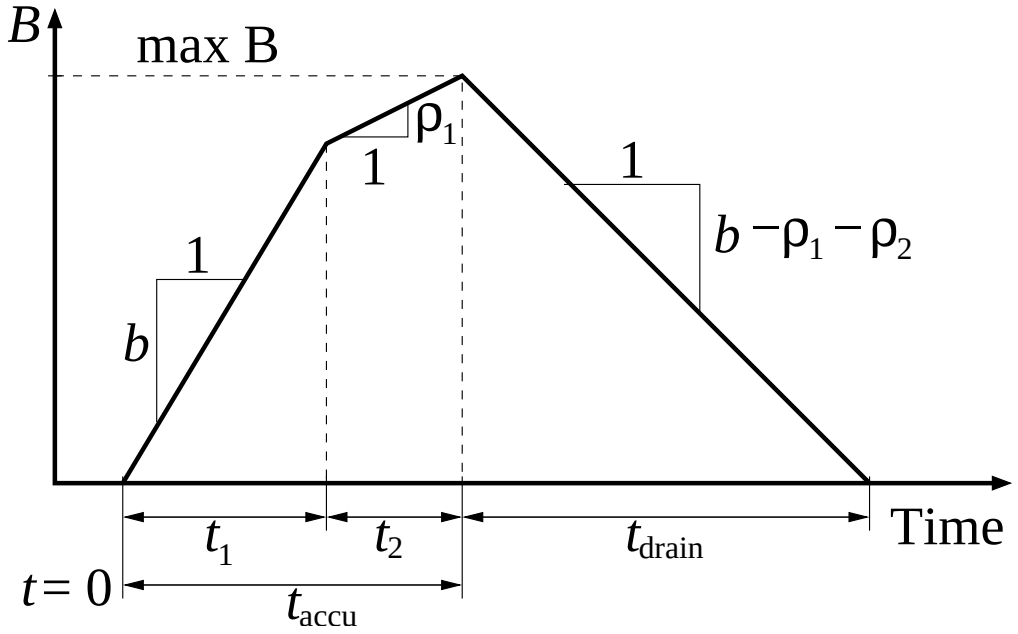
Work Conserving Multiplexer - Summary



$$B_{\max} = \sigma_1 + \frac{\rho_1 \sigma_2}{b - \rho_2}$$

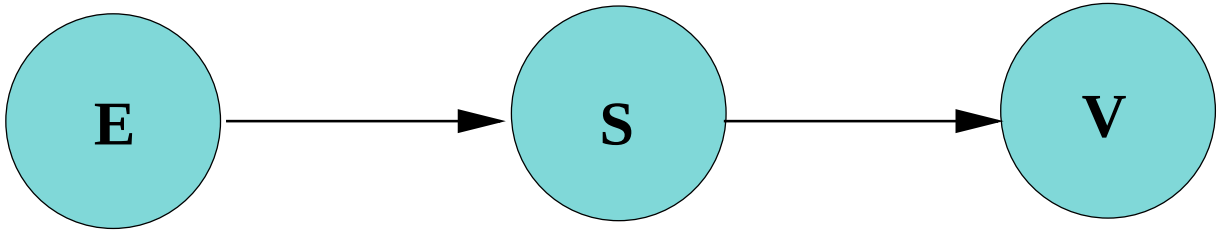
$$D_{\max} = t_{\text{accu}} + t_{\text{drain}} = \frac{\sigma_1 + \sigma_2}{b - \rho_1 - \rho_2}$$

Work Conserving Multiplexer - Summary

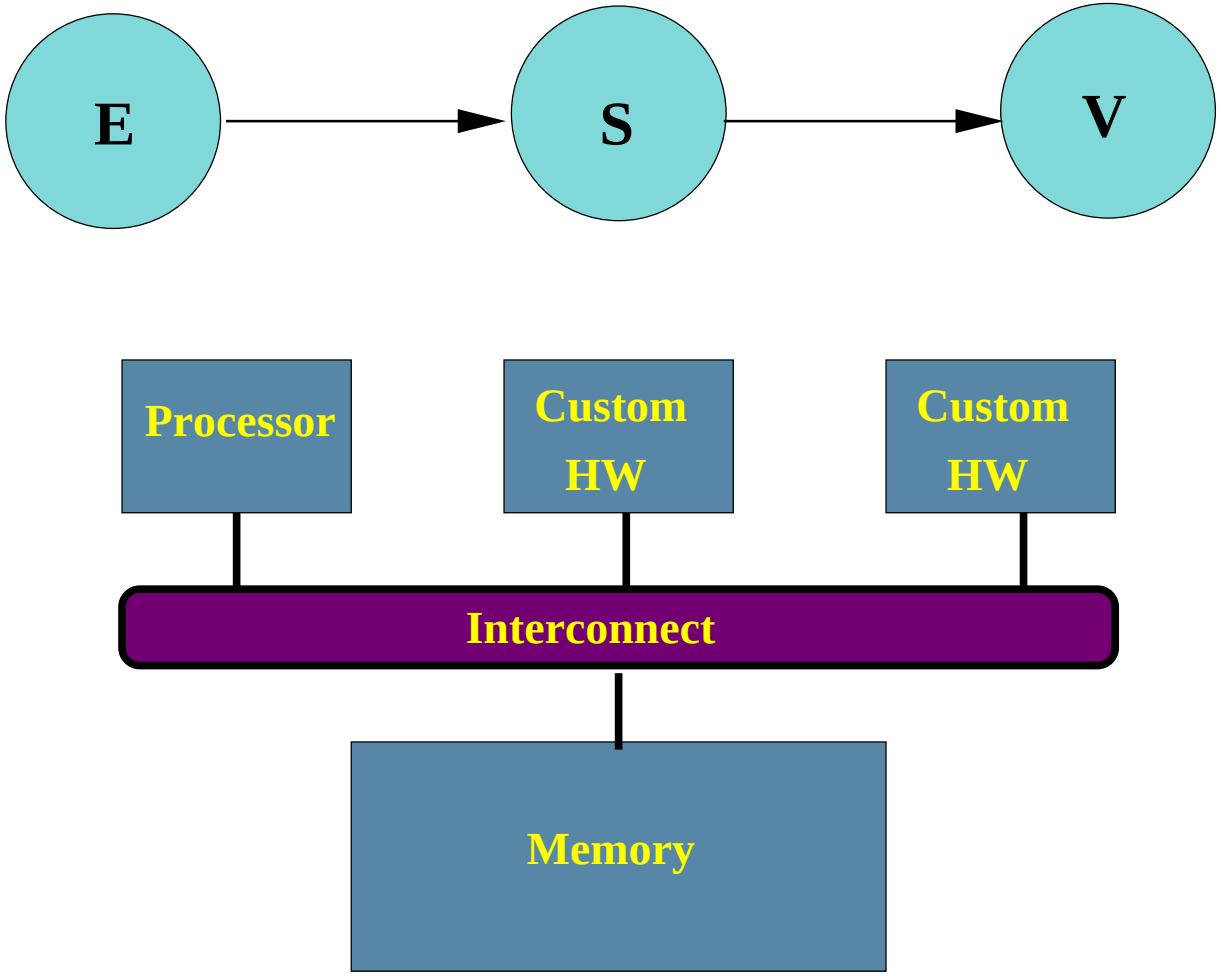


$$
\begin{aligned}
B_{\max} &= \sigma_1 + \frac{\rho_1 \sigma_2}{b - \rho_2} \\
D_{\max} &= t_{\text{accu}} + t_{\text{drain}} = \frac{\sigma_1 + \sigma_2}{b - \rho_1 - \rho_2} \\
F_3 &\sim (\sigma_1 + \sigma_2, \rho_1 + \rho_2)
\end{aligned}
$$

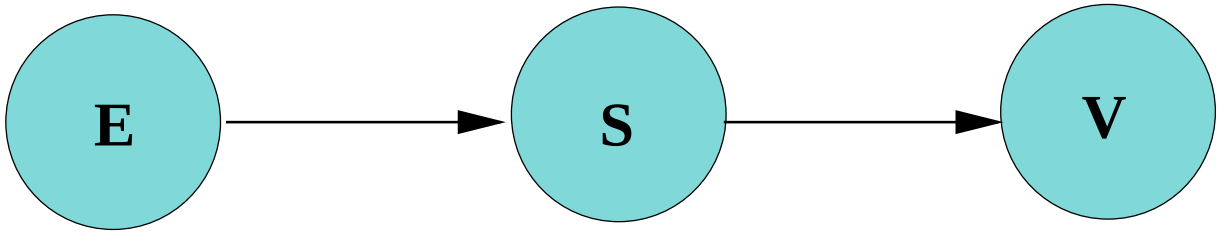
MPEG Encoding Case Study

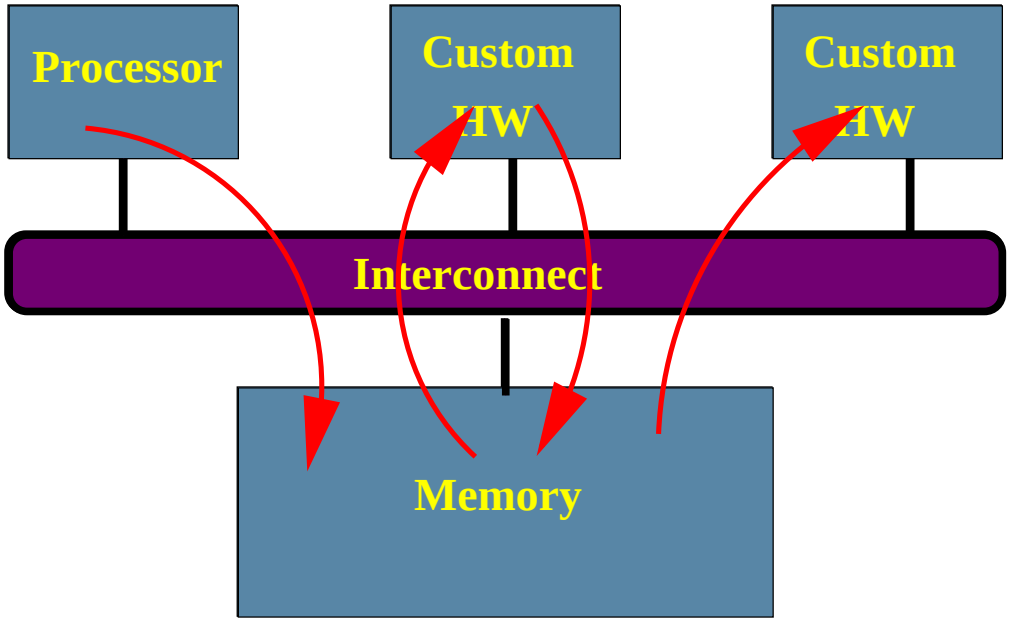


MPEG Encoding Case Study

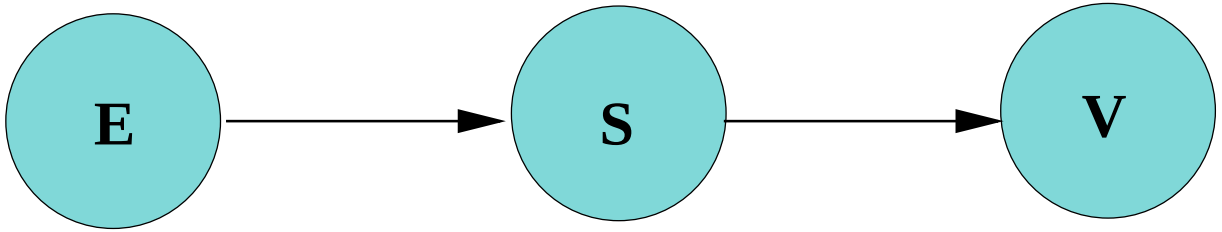


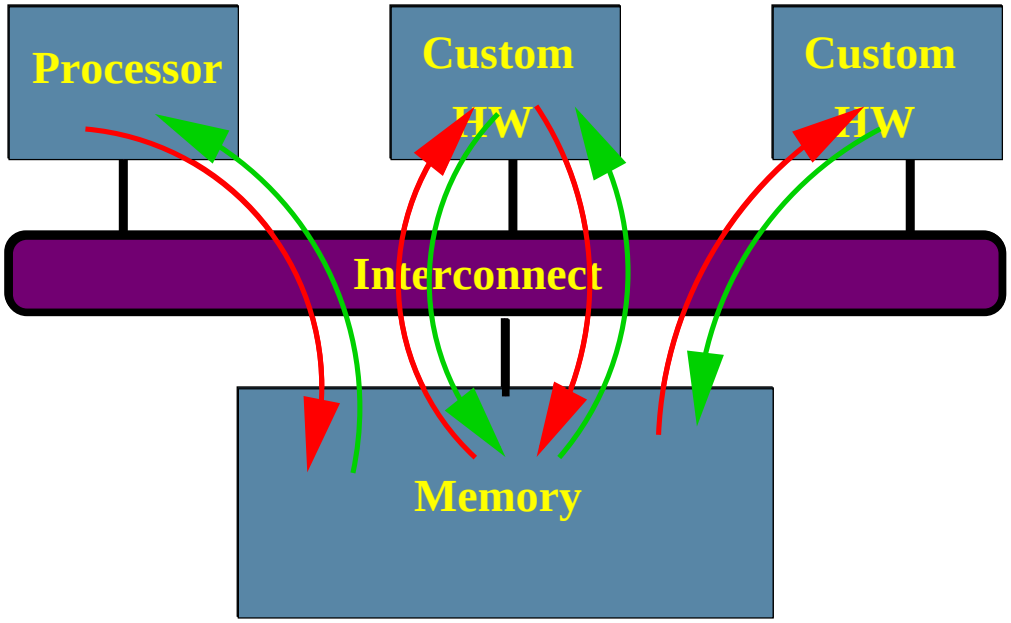
MPEG Encoding Case Study



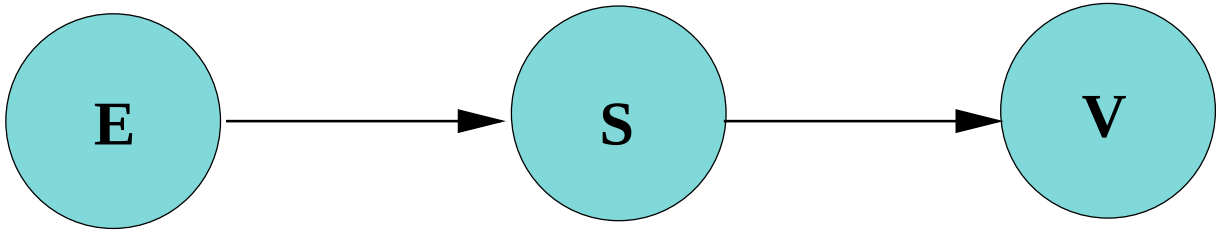


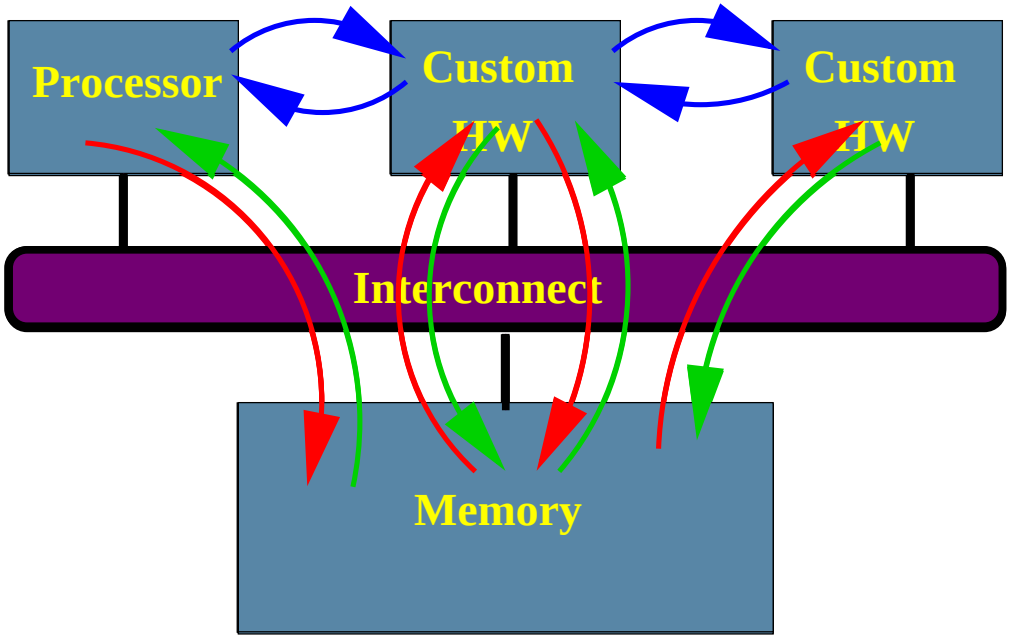
MPEG Encoding Case Study



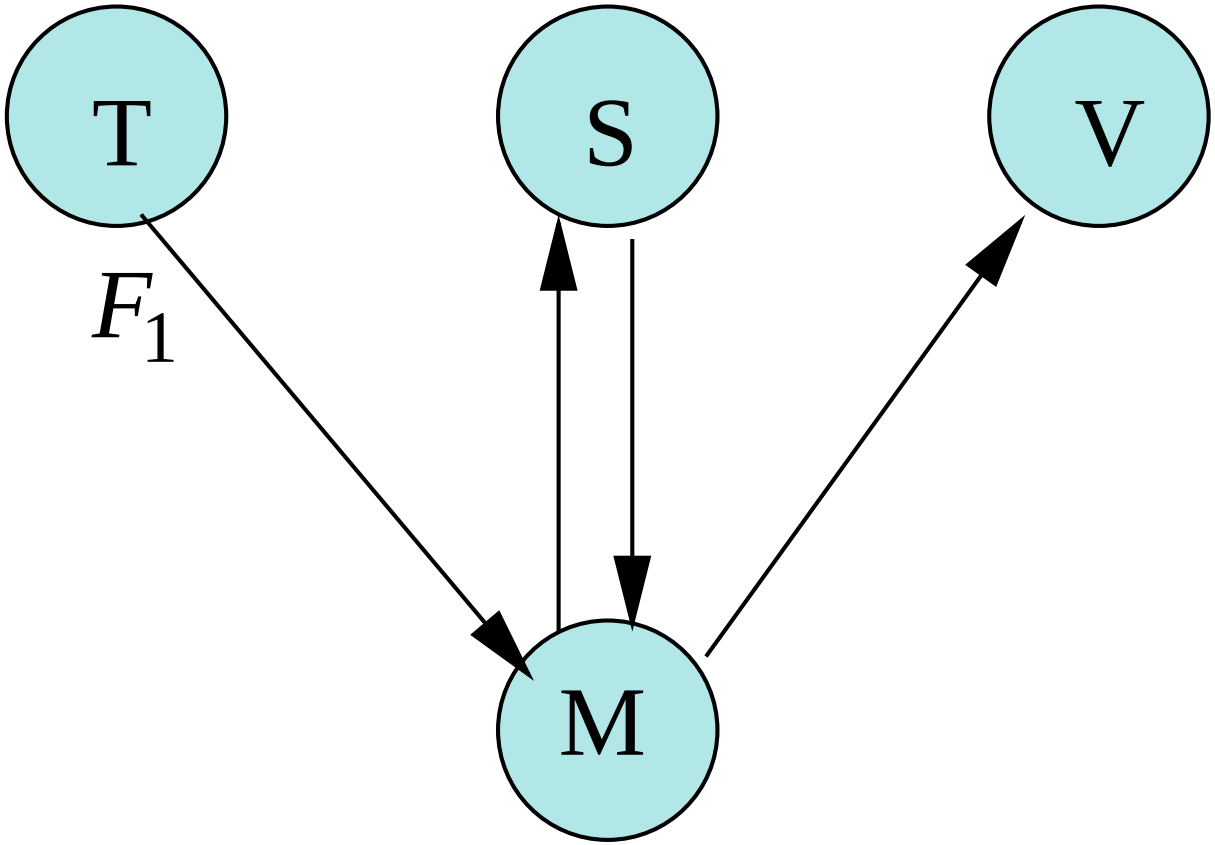


MPEG Encoding Case Study



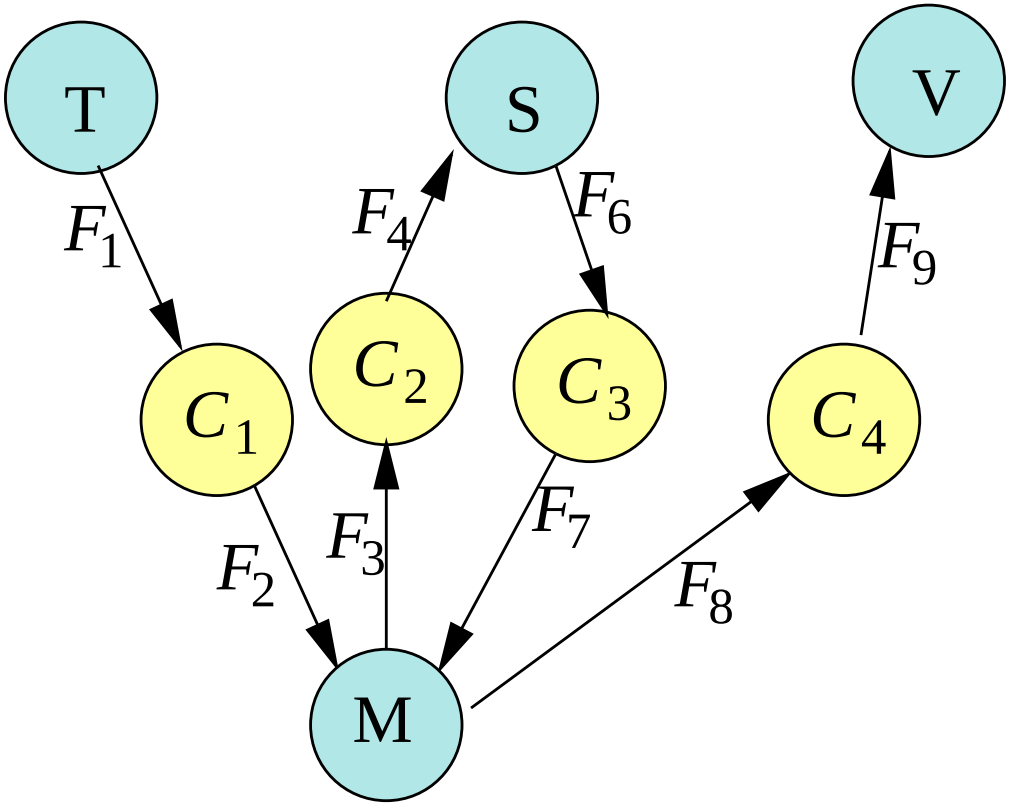


MPEG Encoding Case Study - cont'd



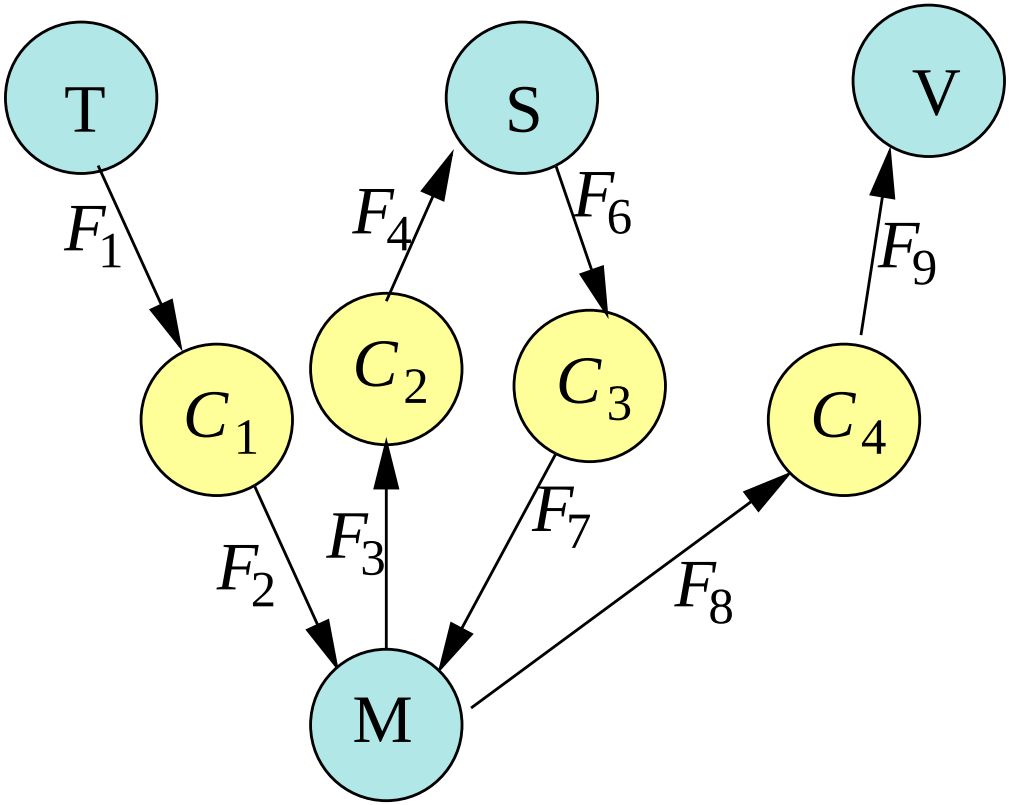
$$F_1 \sim (0, \rho_t)$$

MPEG Encoding Case Study - cont'd



$F_1 \sim (0, \rho_t)$

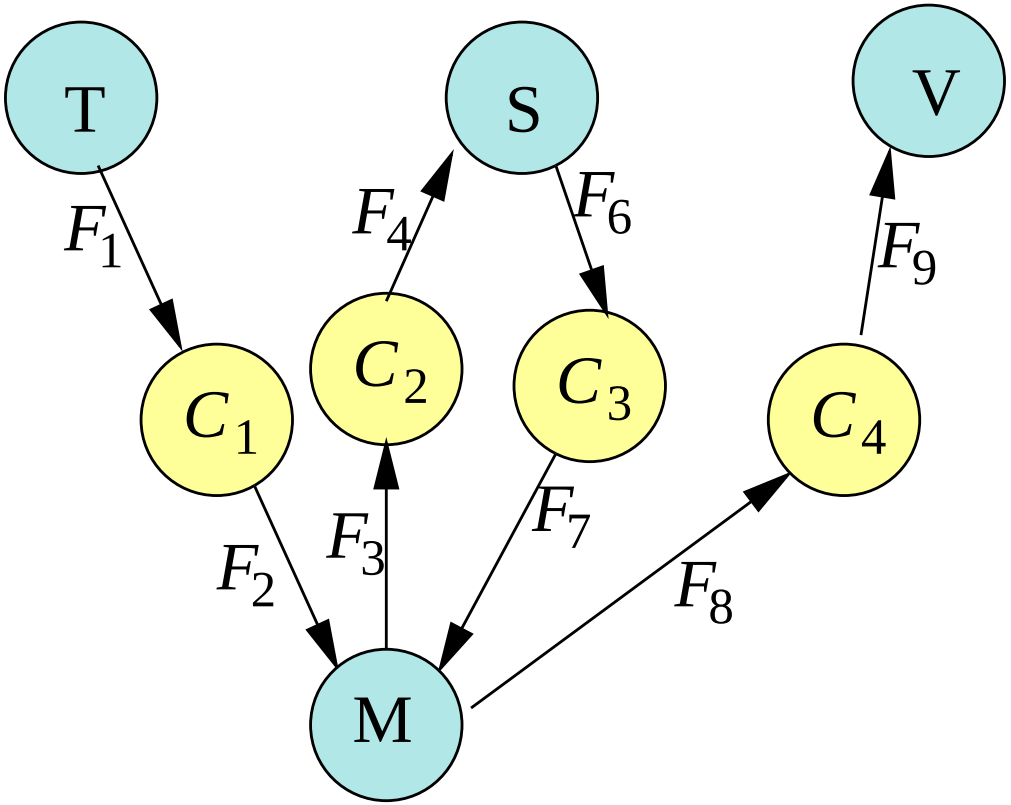
MPEG Encoding Case Study - cont'd



$$F_1 \sim (0, \rho_t)$$

$$C_1 : (\rho_t, D_1)$$
$$C_2 : (\rho_t, D_2)$$
$$C_1 : (\rho_t, D_3)$$
$$C_4 : (\rho_t, D_4)$$

MPEG Encoding Case Study - cont'd

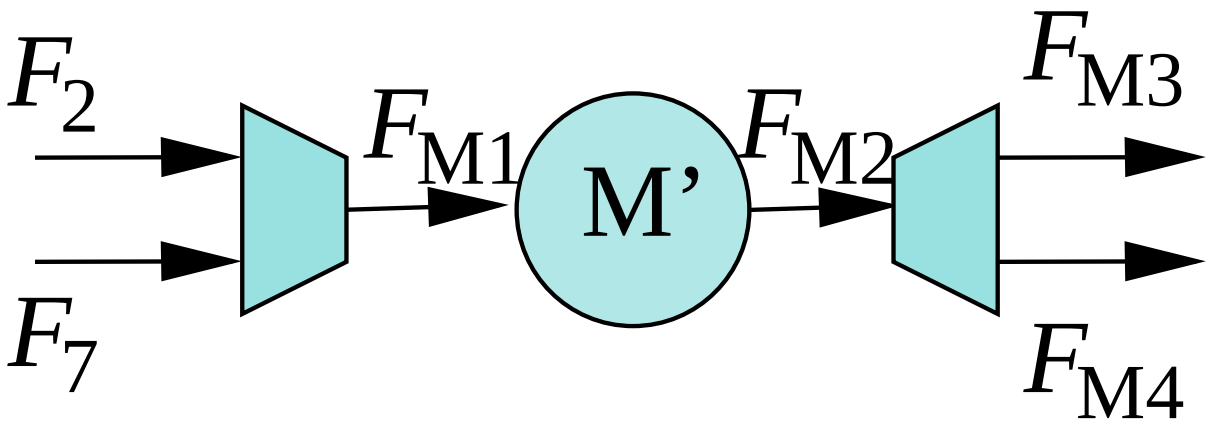


$$F_1 \sim (0, \rho_t)$$

$$C_1 : (\rho_t, D_1)$$
$$C_2 : (\rho_t, D_2)$$
$$C_1 : (\rho_t, D_3)$$
$$C_4 : (\rho_t, D_4)$$

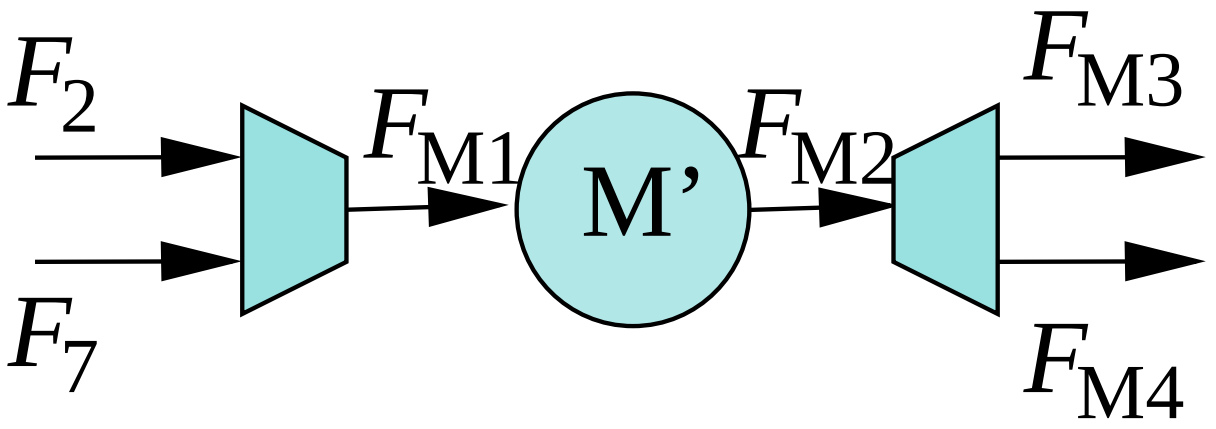
$$F_2 \sim (\rho_t D_1, \rho_t)$$

MPEG Encoding Case Study - Memory



$$M' \quad : \quad (2\rho_t, D_{M'})$$

MPEG Encoding Case Study - Memory



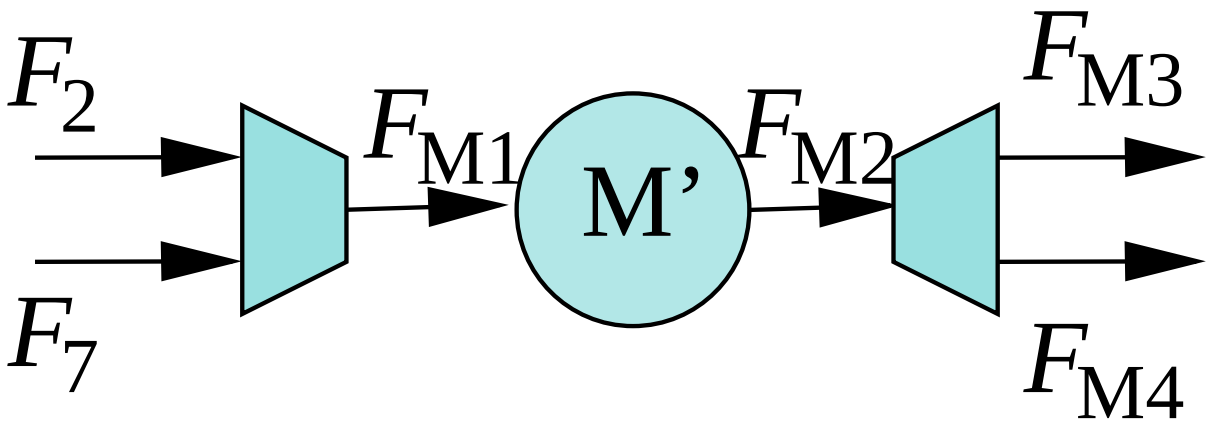
$$M' \quad : \quad (2\rho_t, D_{M'})$$

For a general multiplexer we have:

$$D_{\text{mux}} \quad = \quad \frac{\sigma_1 + \sigma_2}{C_{\text{out}} - \rho_1 - \rho_2}$$

$$F_{\text{muxout}} \quad \sim \quad (\sigma_1 + \sigma_2, \rho_1 + \rho_2)$$

MPEG Encoding Case Study - Memory



$$M' \quad : \quad (2\rho_t, D_{M'})$$

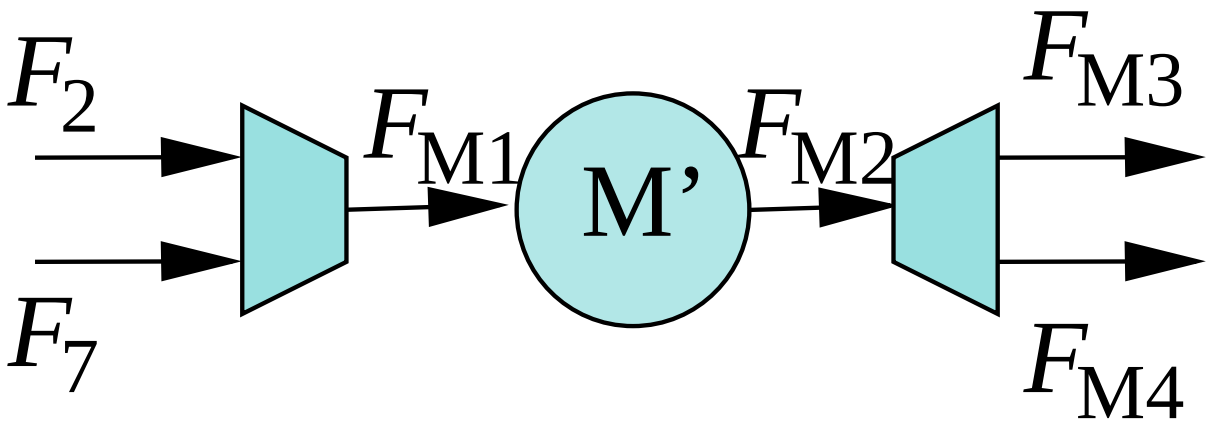
For a general multiplexer we have:

$$D_{\text{mux}} = \frac{\sigma_1 + \sigma_2}{C_{\text{out}} - \rho_1 - \rho_2}$$

$$F_{\text{muxout}} \sim (\sigma_1 + \sigma_2, \rho_1 + \rho_2)$$

$$F_{M3} \sim (\rho_t(D_1 + D_{\text{mux}} + D_{M'}), \rho_t)$$

MPEG Encoding Case Study - Memory



$$M' \quad : \quad (2\rho_t, D_{M'})$$

For a general multiplexer we have:

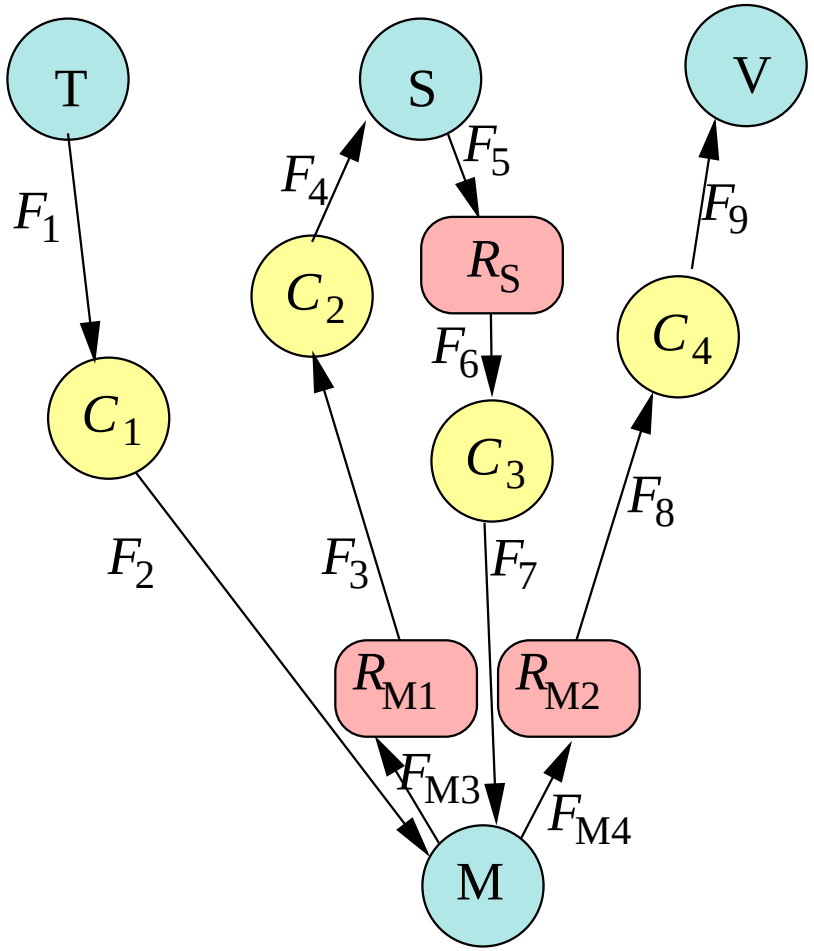
$$D_{\text{mux}} = \frac{\sigma_1 + \sigma_2}{C_{\text{out}} - \rho_1 - \rho_2}$$

$$F_{\text{muxout}} \sim (\sigma_1 + \sigma_2, \rho_1 + \rho_2)$$

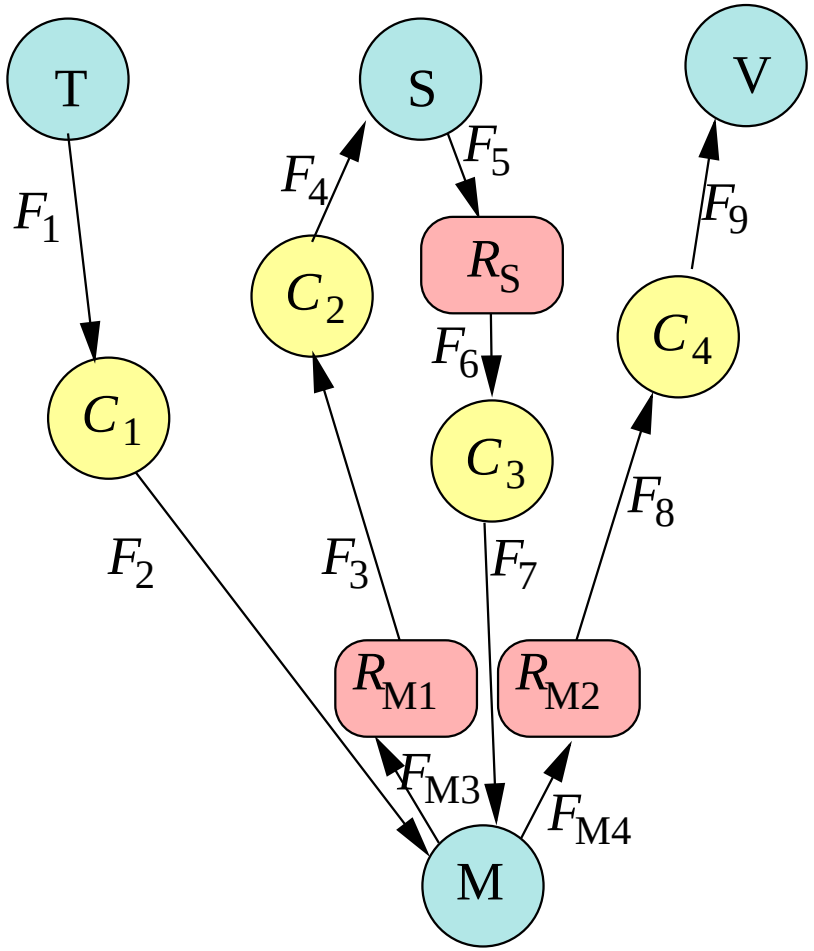
$$F_{M3} \sim (\rho_t(D_1 + D_{\text{mux}} + D_{M'}), \rho_t)$$

$$F_{M4} \sim ?$$

MPEG Encoding Case Study - cont'd



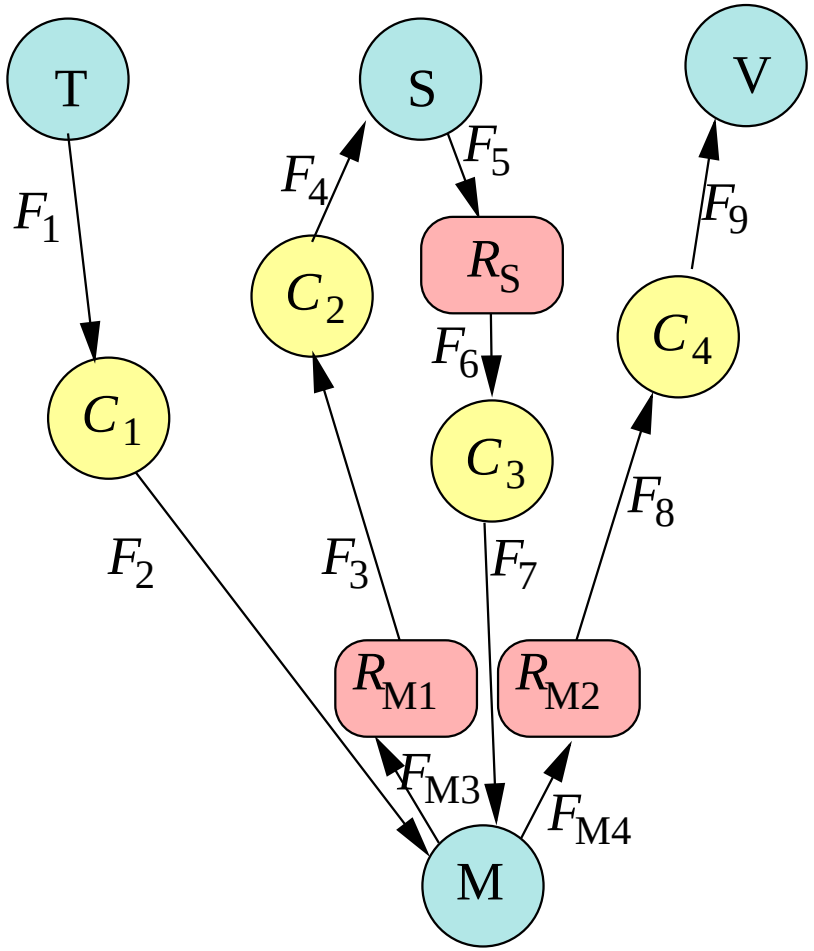
MPEG Encoding Case Study - cont'd



$$R_{M1} \sim (S_{\text{buffer}}, \rho_t);$$
$$R_{M2} \sim (S_{\text{buffer}}, \rho_t);$$
$$R_S \sim (S_{\text{buffer}}, \rho_t);$$

S_{buffer} is the size of the input buffer in S.

MPEG Encoding Case Study - cont'd

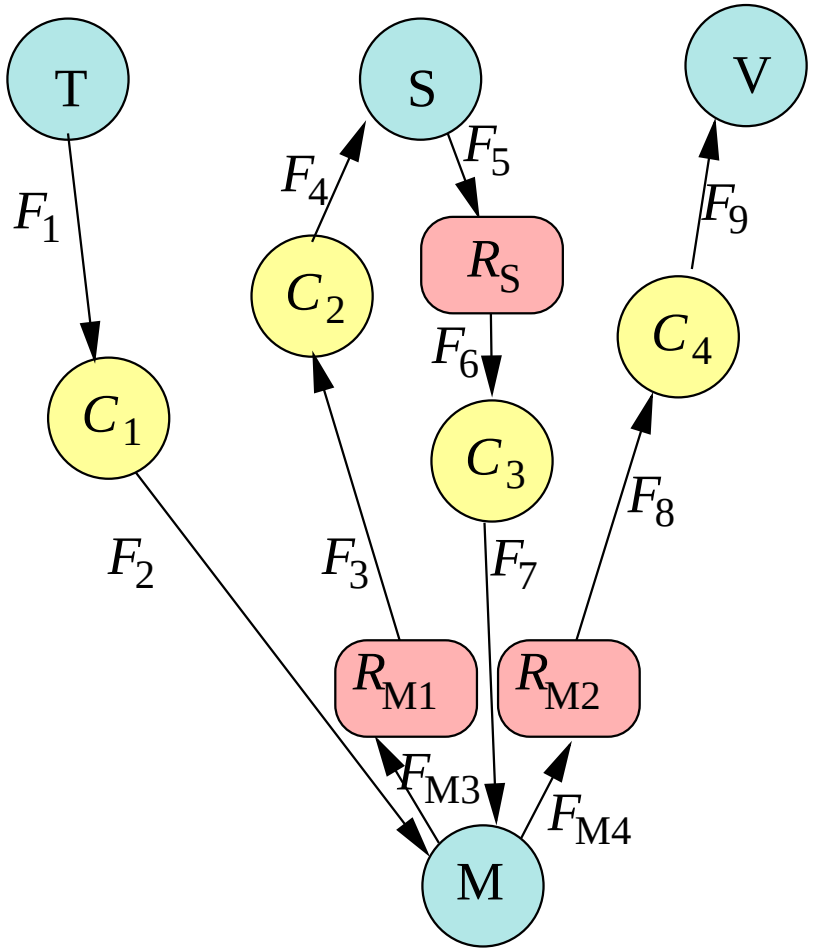


$$R_{M1} \sim (S_{\text{buffer}}, \rho_t);$$
$$R_{M2} \sim (S_{\text{buffer}}, \rho_t);$$
$$R_S \sim (S_{\text{buffer}}, \rho_t);$$

S_{buffer} is the size of the input buffer in S.

$$D_{(\sigma,\,\rho)\text{-regulator}} = \frac{\max(0, \sigma' - \sigma)}{\rho}$$
$$B_{(\sigma,\,\rho)\text{-regulator}} = \max(0, \sigma' - \sigma)$$

MPEG Encoding Case Study - cont'd

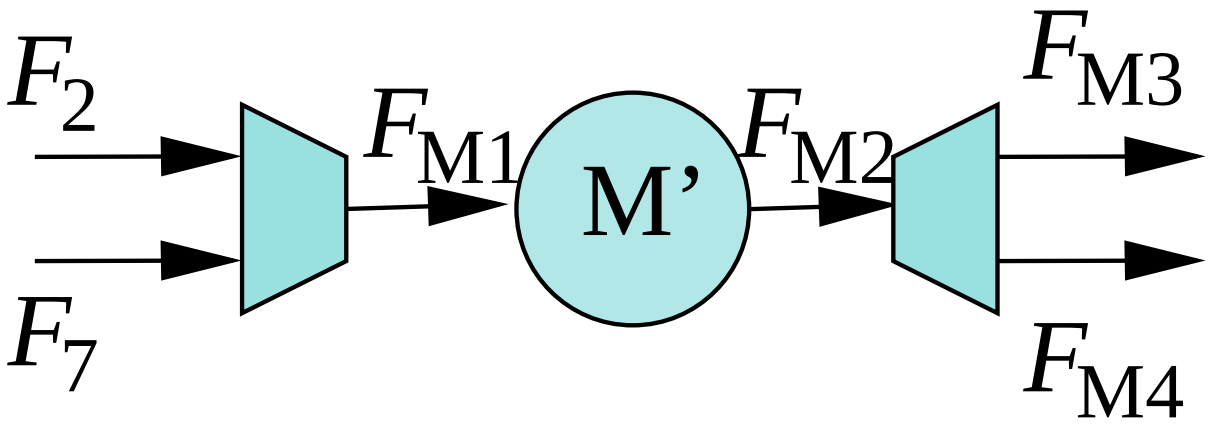


$$R_{M1} \sim (S_{\text{buffer}}, \rho_t);$$
$$R_{M2} \sim (S_{\text{buffer}}, \rho_t);$$
$$R_S \sim (S_{\text{buffer}}, \rho_t);$$

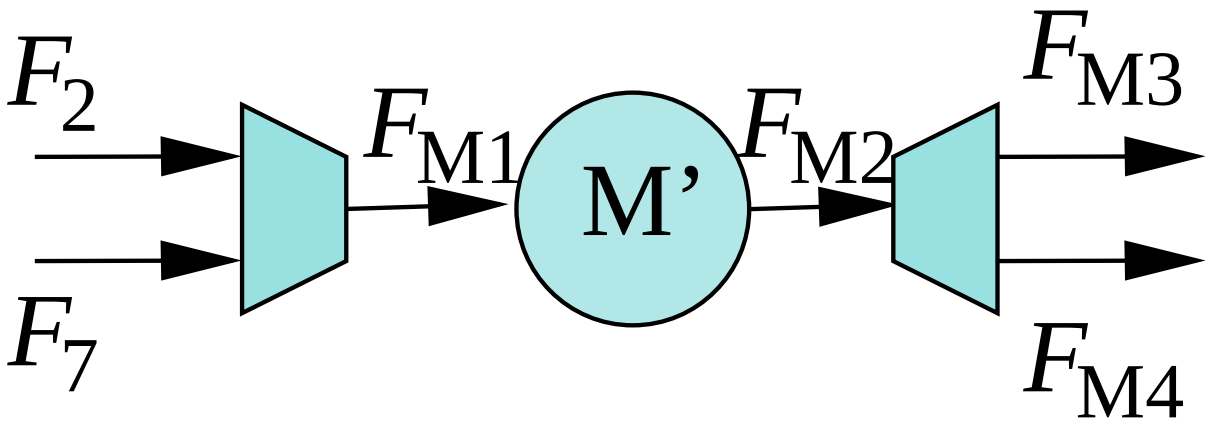
S_{buffer} is the size of the input buffer in S.

$$\begin{aligned} F_6 &\sim (S_{\text{buffer}}, \rho_t) \\ C_3 &: (\rho_t, D_3) \\ F_7 &\sim (S_{\text{buffer}} + \rho_t D_3, \rho_t) \end{aligned}$$

MPEG Encoding Case Study - Memory

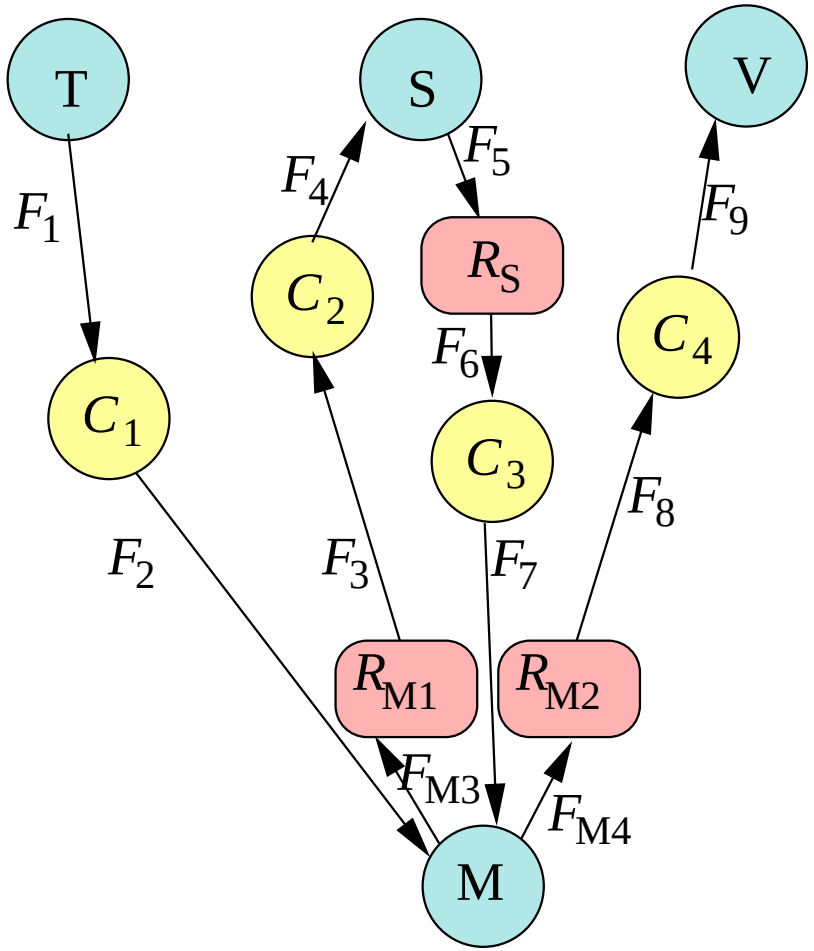


MPEG Encoding Case Study - Memory

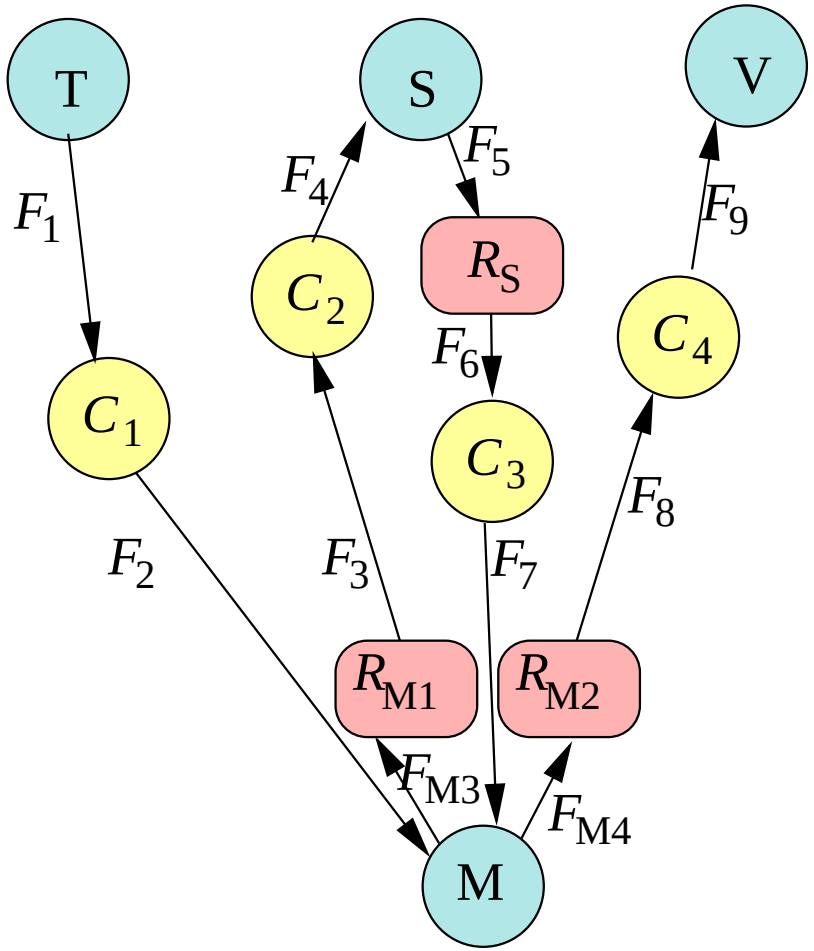


$$
\begin{aligned}
M' &: (2\rho_t, D_{M'}) \\
D_{\text{mux}} &= \frac{\sigma_1 + \sigma_2}{C_{\text{out}} - \rho_1 - \rho_2} = \frac{S_{\text{buffer}} + \rho_t(D_1 + D_3)}{C_{\text{out}} - 2\rho_t} \\
F_{M1} &\sim (S_{\text{buffer}} + \rho_t(D_1 + D_3), 2\rho_t) \\
F_{M2} &\sim (S_{\text{buffer}} + \rho_t(D_1 + D_3 + 2D_{M'}), 2\rho_t) \\
F_{M3} &\sim (\rho_t(D_1 + D_{mux} + D_{M'}), \rho_t) \\
F_{M4} &\sim (S_{\text{buffer}} + \rho_t(D_3 + D_{mux} + D_{M'}), \rho_t)
\end{aligned}
$$

MPEG Encoding Case Study - cont'd



MPEG Encoding Case Study - cont'd



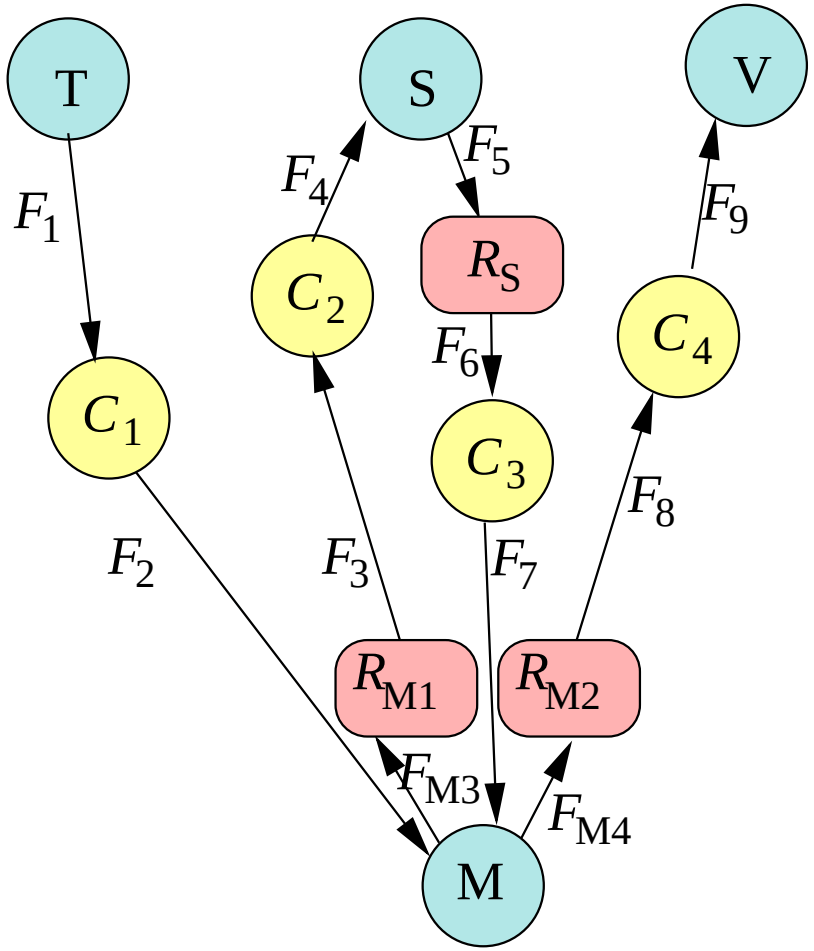
Backlog of the regulators:

$$
\begin{aligned}
B_{RM1} &= \max(0, \rho_t(D_1 + D_{\text{mux}} + D_{M'}) \\
&\quad -S_{\text{buffer}}) \\
B_{RM2} &= \max(0, 128B + \rho_t(D_3 + D_{\text{mux}} \\
&\quad +D_{M'}) - S_{\text{buffer}})
\end{aligned}
$$

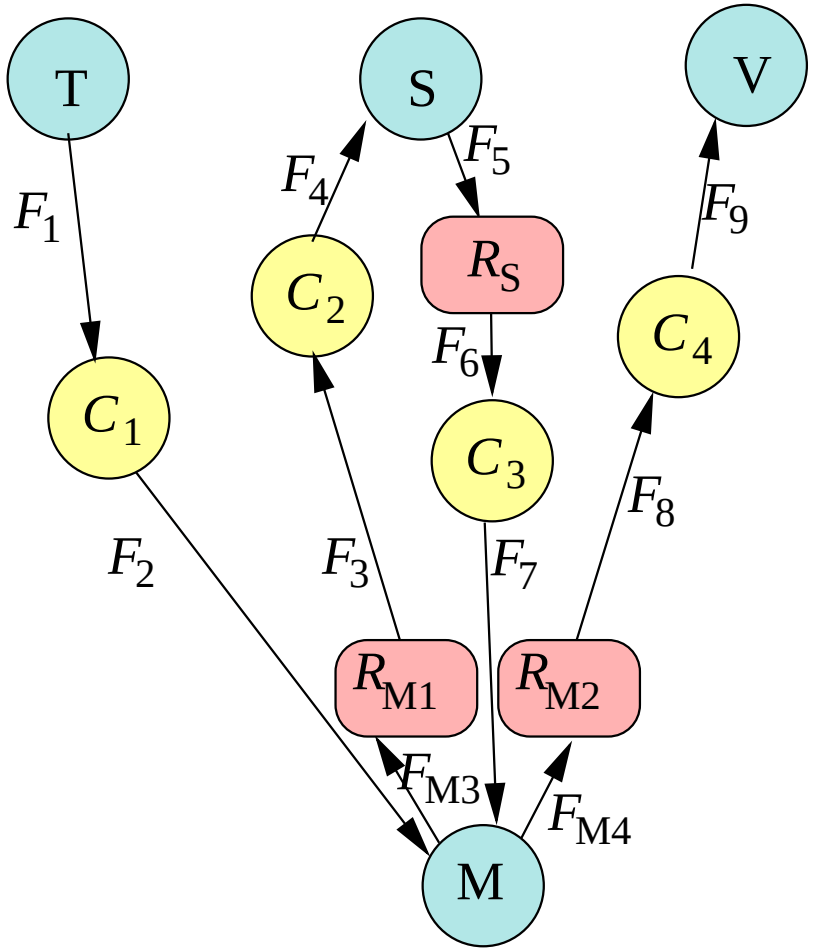
Delay of the regulators:

$$
\begin{aligned}
D_{RM1} &= \frac{B_{RM1}}{\rho_t} \\
D_{RM2} &= \frac{B_{RM2}}{\rho_t}
\end{aligned}
$$

MPEG Encoding Case Study - cont'd



MPEG Encoding Case Study - cont'd



The flow from the memory to S:

$$\begin{aligned} F_3 &\sim (S_{\text{buffer}}, \rho_t) \\ C_2 &: (\rho_t, D_2) \\ F_4 &\sim (S_{\text{buffer}} + \rho D_2, \rho_t), \end{aligned}$$

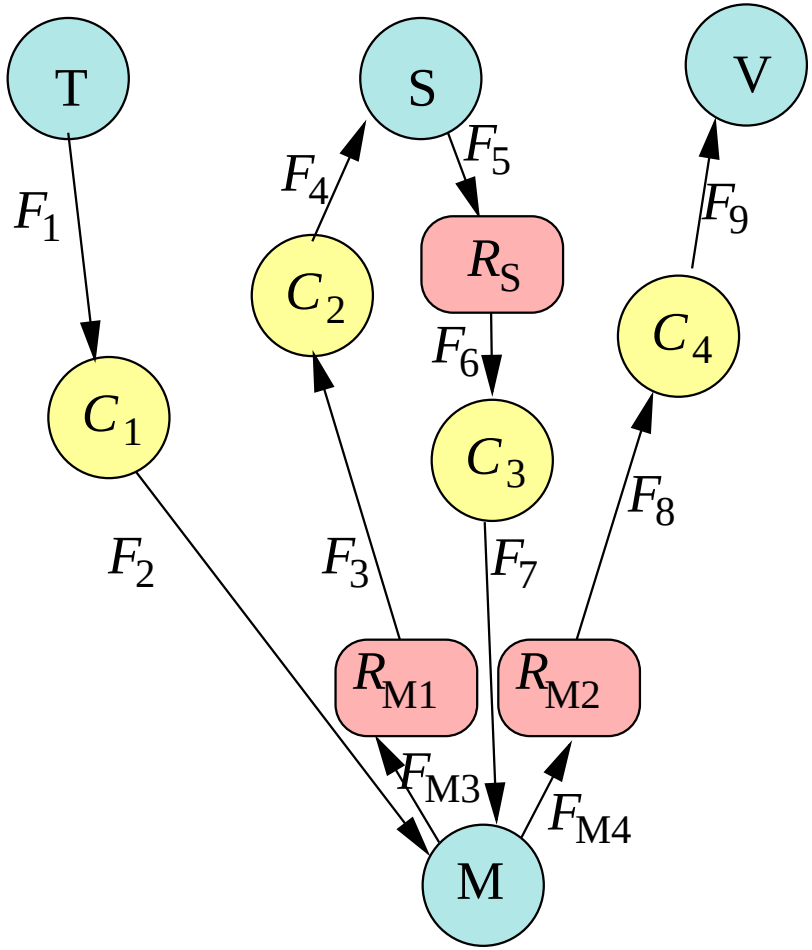
A charatcerization of S and its output:

$$\begin{aligned} S &: (\rho_t, \frac{S_{\text{buffer}}}{\rho_t}) \\ F_5 &\sim (2S_{\text{buffer}} + \rho_t D_2, \rho_t) \end{aligned}$$

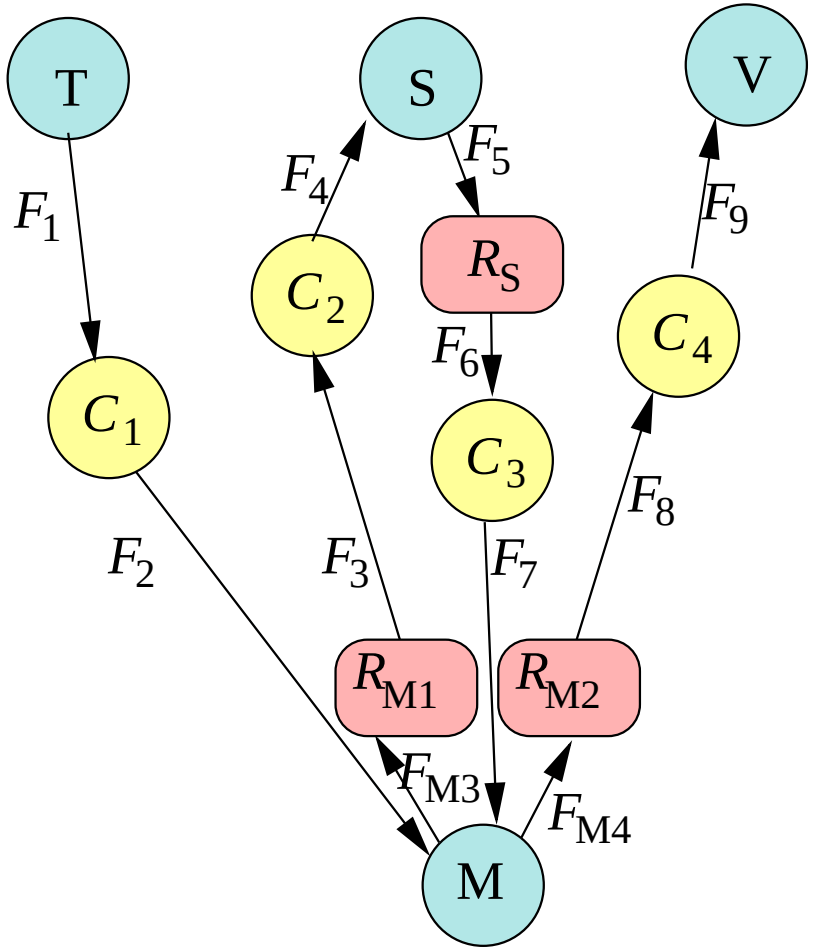
The flows between memory and V:

$$\begin{aligned} F_8 &\sim (S_{\text{buffer}}, \rho_t) \\ C_4 &: (\rho, D_4) \\ F_9 &\sim (S_{\text{buffer}} + \rho_t D_4, \rho_t) \end{aligned}$$

MPEG Encoding Case Study - cont'd



MPEG Encoding Case Study - cont'd



End to end delay:

$$\begin{aligned} D_{\text{total}} = & D_1 + D_{\text{mux}} + D_{M'} \\ & + D_{RM1} + D_2 + D_S + D_{RS} + D_3 \\ & + D_{\text{mux}} + D_{M'} + D_{RM2} + D_4 \end{aligned}$$

The flow at V:

$$F_{T \to V} \sim (0 + \rho_t D_{total}, \rho_t)$$

Modeling with Regulated Flows

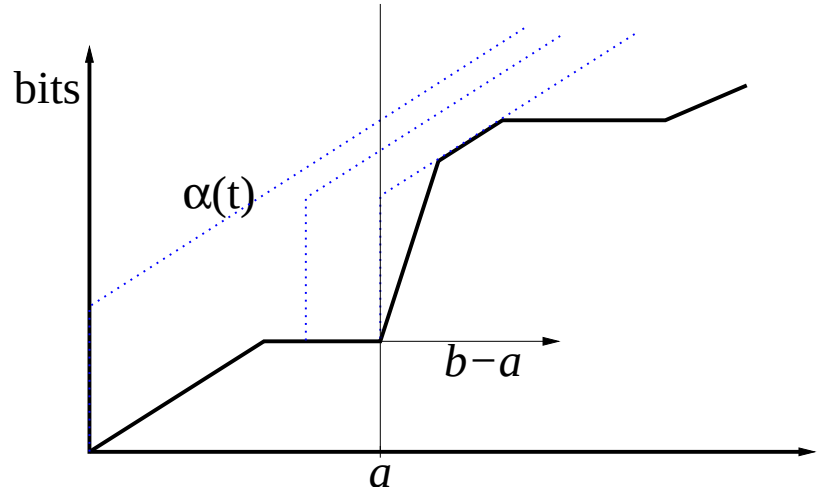
- Interconnect:
  - ⋆ Model each channel by available bandwidth and maximum delay variation;
  - ⋆ Model each node in the interconnect as an arbiter;
- Model read request, write acknowledge as separate flows;
- Model synchronization as separate flows;
- A simple generalization of (σ, ρ) flows is

$$F \sim \min(\sigma_i, \rho_i), i > 0$$

$$F(b) - F(a) \leq \min_i(\sigma_i + \rho_i(b - a))$$

- Good analysis depends on good element models;

Network Calculus - Arrival Curves



Given a monotonically increasing function α, defined for $t \geq 0$, α is an arrival curve for flow F if for all $0 \leq a \leq b$:

$$F(b) - F(a) \leq \alpha(b - a)$$

Network Calculus - Min-Plus Convolusion

Given two monotonically increasing functions f and g. The min-plus convolusion of f and g is the function

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

Network Calculus - Min-Plus Convolusion

Given two monotonically increasing functions f and g. The min-plus convolusion of f and g is the function

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

If α is an arrival curve for F we have:

$$F \leq F \otimes \alpha$$

Network Calculus - Min-Plus Convolusion

Given two monotonically increasing functions f and g. The min-plus convolusion of f and g is the function

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

If α is an arrival curve for F we have:

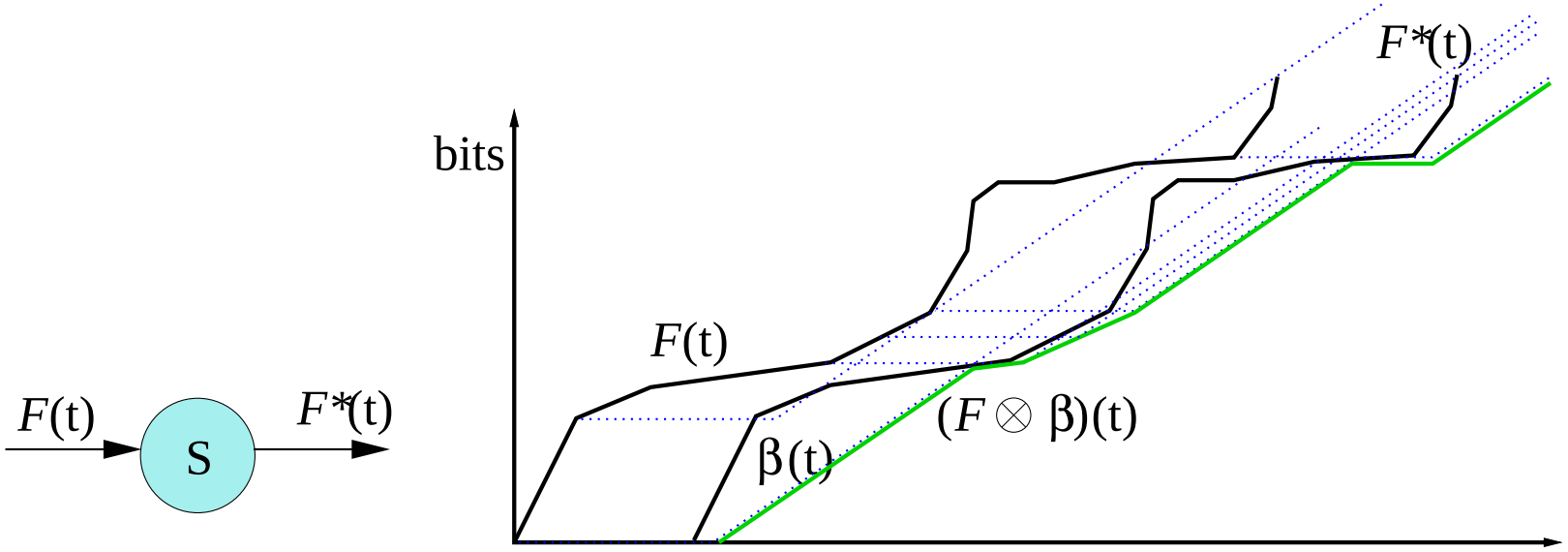
$$F \leq F \otimes \alpha$$

and

$$F \leq \alpha \otimes \alpha$$

with $\alpha \otimes \alpha$ being the best bound that we can find based on information of α.

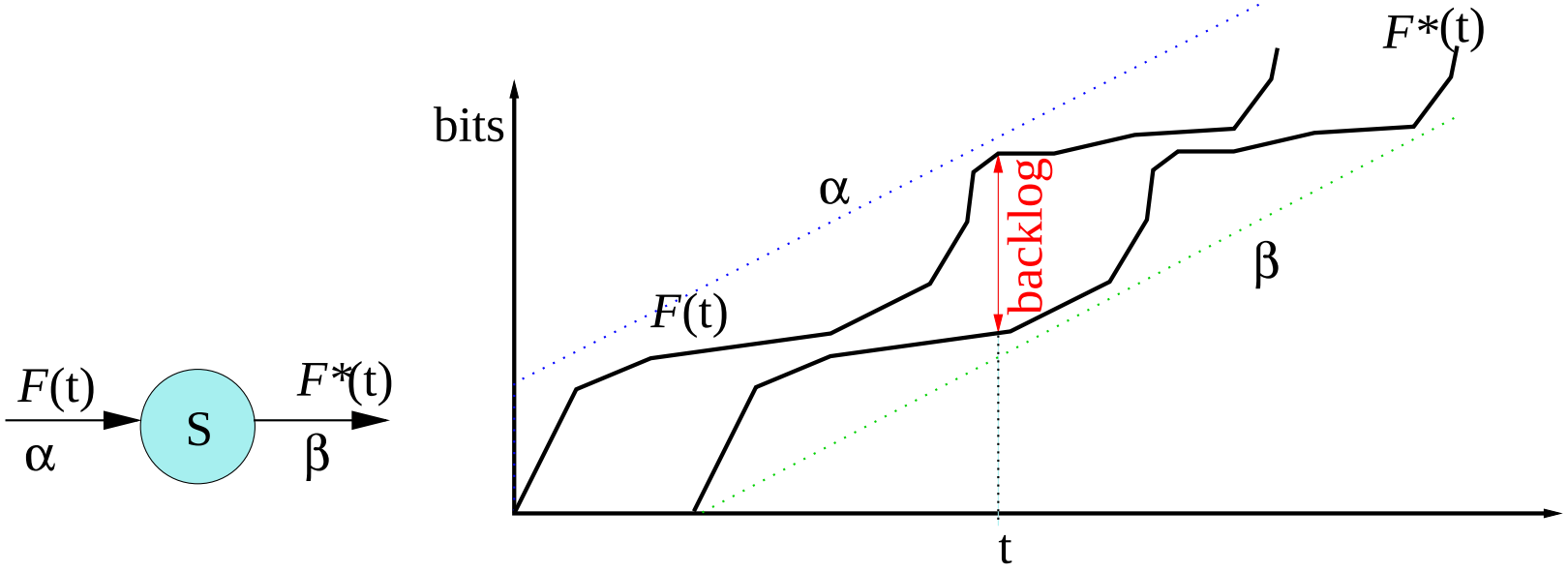
Network Calculus - Service Curves



Given a system S with an input flow F and an output flow F^*. S offers the flow a service curve β if and only if β is a monotonically increasing function and $F^* \geq F \otimes \beta$ which means that

$$F^*(t) \geq \inf_{s \leq t}(F(t) + \beta(t-s))$$

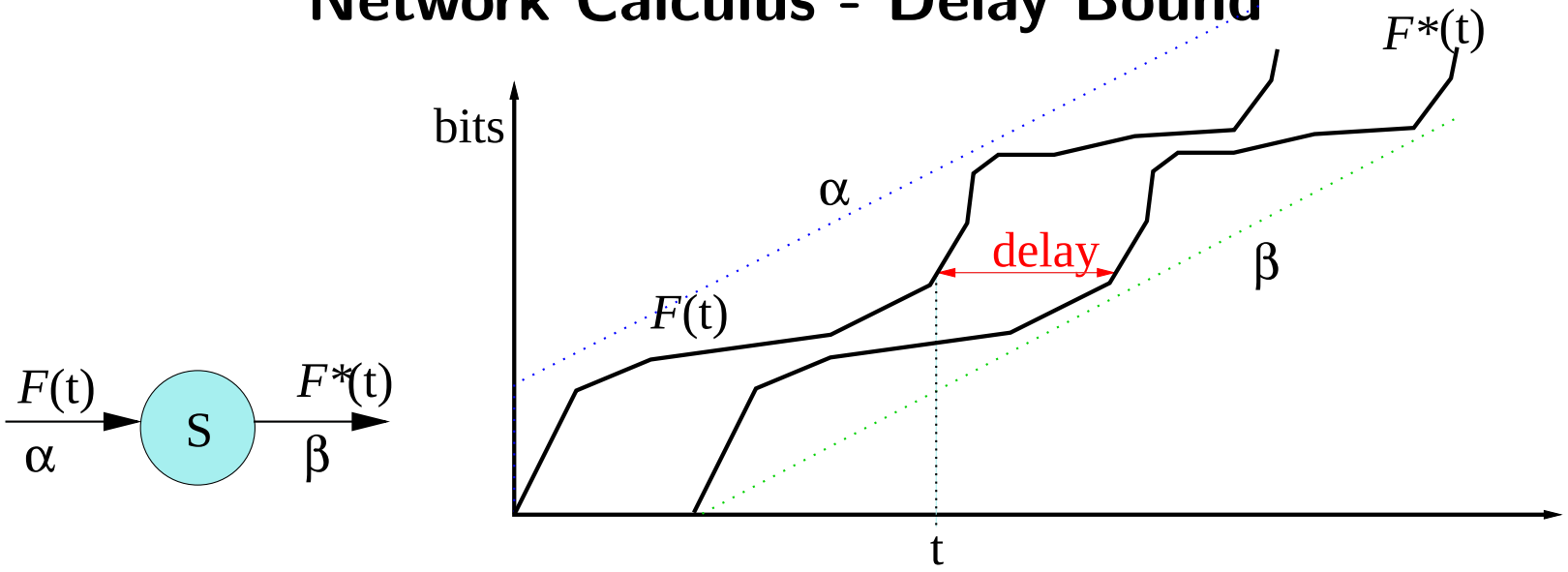
Network Calculus - Backlog Bound



Given a flow F constrained by arrival curve α and a system offering a service curve β, the backlog $F(t) - F^*(t)$ for all t satisfies

$$F(t) - F^*(t) \leq \sup_{s \geq 0}(\alpha(s) - \beta(s))$$

Network Calculus - Delay Bound



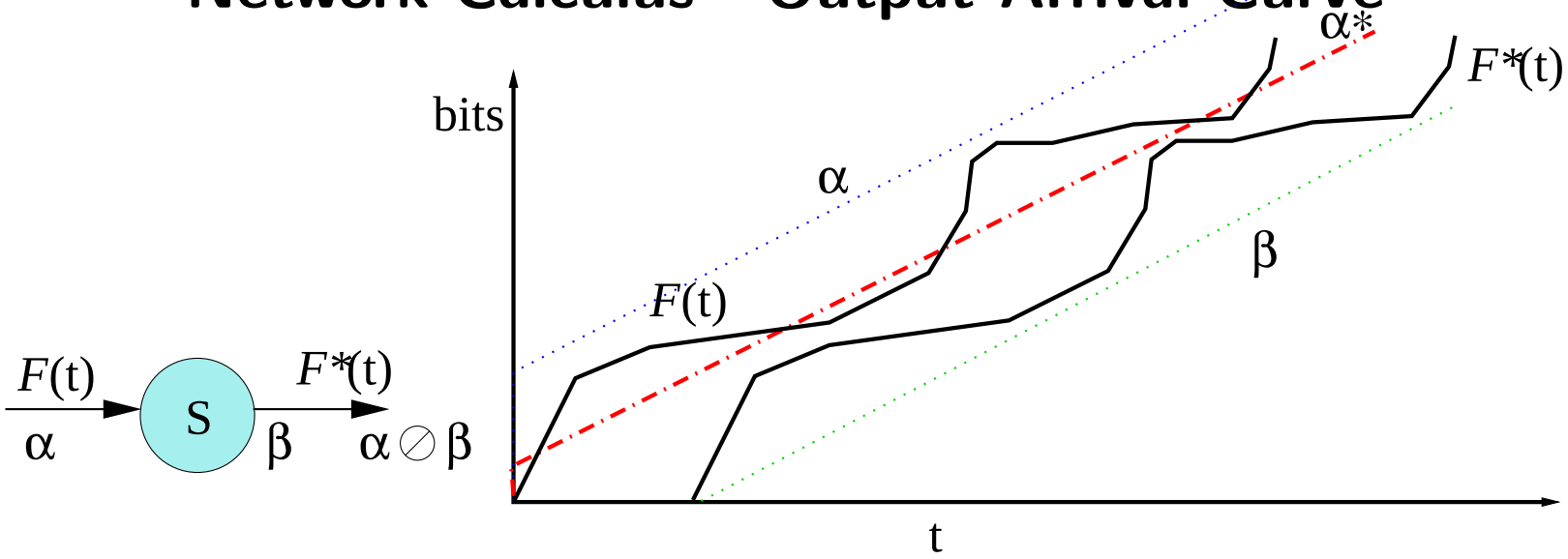
Given a flow F constrained by arrival curve α and a system offering a service curve β, the delay $d(t)$ at time t is

$$d(t) = \inf(\tau \geq 0 : F(t) \leq F^*(t+\tau)).$$

It satisfies

$$d(t) \leq h(\alpha, \beta) = \sup_{t \geq 0}(\inf(\tau \geq 0 : \alpha(t) \leq \beta(t+\tau)))$$

Network Calculus - Output Arrival Curve

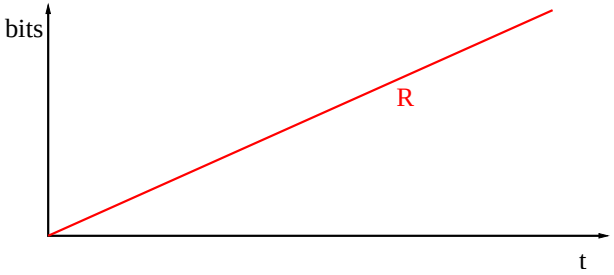


Given a flow F constrained by arrival curve α and a system offering a service curve β, the output flow F^* is constrained by the arrival curve α^*

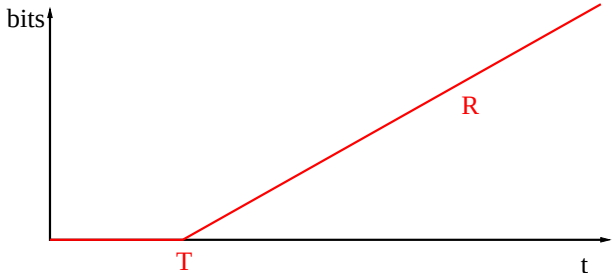
$$\alpha^* = \alpha \oslash \beta.$$

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

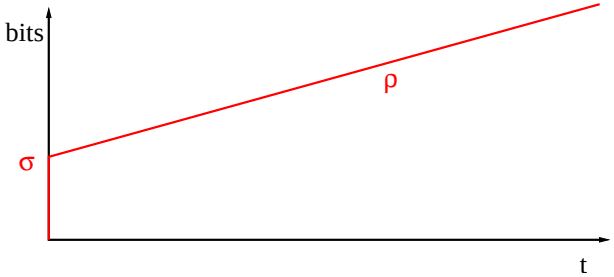
Network Calculus - Useful Functions



Peak rate function:
$\lambda_R(t) = Rt$

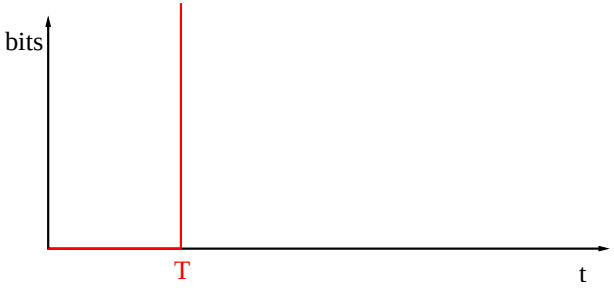


Rate latency function:
$\beta_{R,T}(t) = R[t-T]^+$



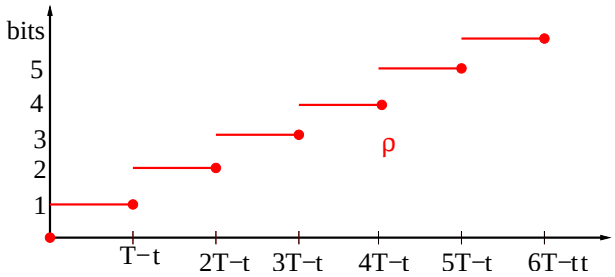
Affine function:

$$\gamma_{\sigma,\rho}(t) = \begin{cases} 0 & \text{for } t = 0 \\ \sigma + \rho t & \text{for } t > 0 \end{cases}$$

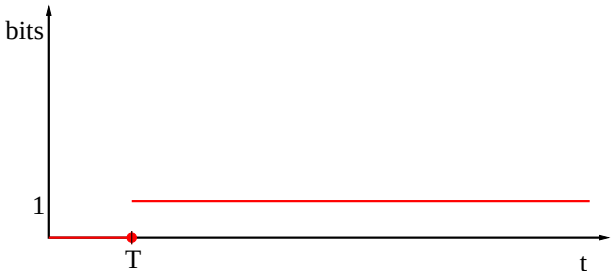


Burst-delay function:

$$\delta_T(t) = \begin{cases} 0 & \text{for } t \leq T \\ \infty & \text{for } t > T \end{cases}$$



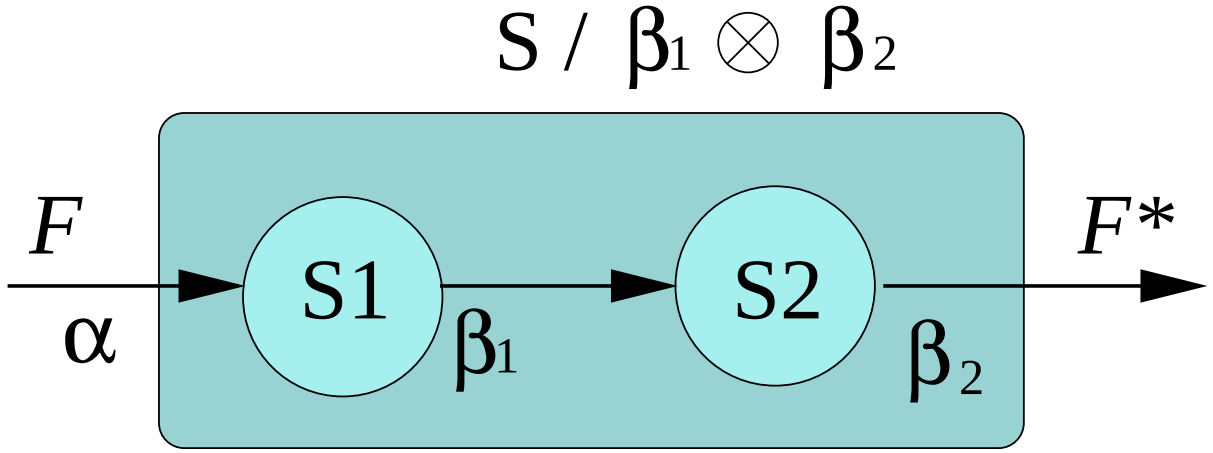
Staircase function:
$v_{T,\tau}(t) = \lceil (t+\tau)/T \rceil$



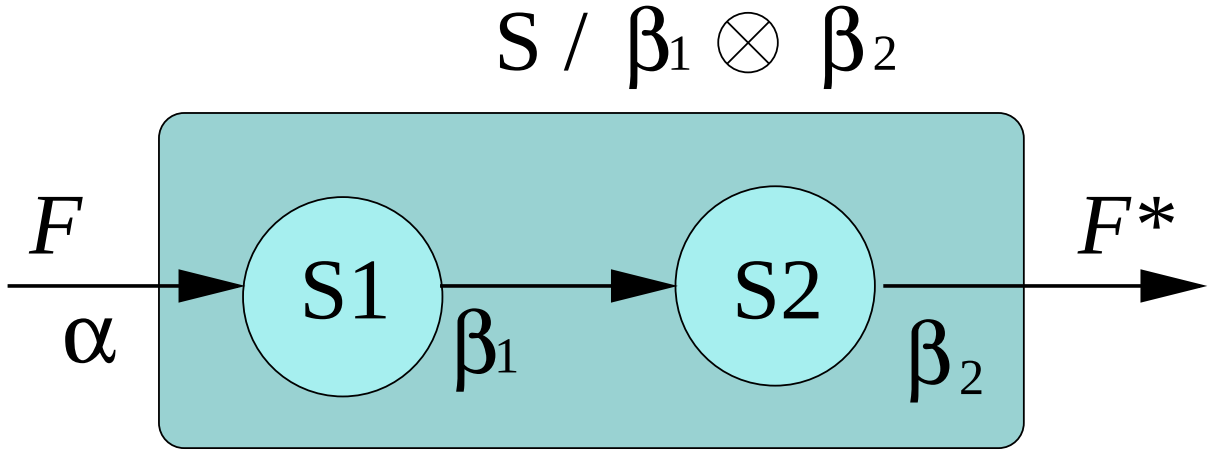
Step function:

$$u_T(t) = \begin{cases} 0 & \text{for } t \leq T \\ 1 & \text{for } t > T \end{cases}$$

Network Calculus - Concatenation of Nodes



Network Calculus - Concatenation of Nodes



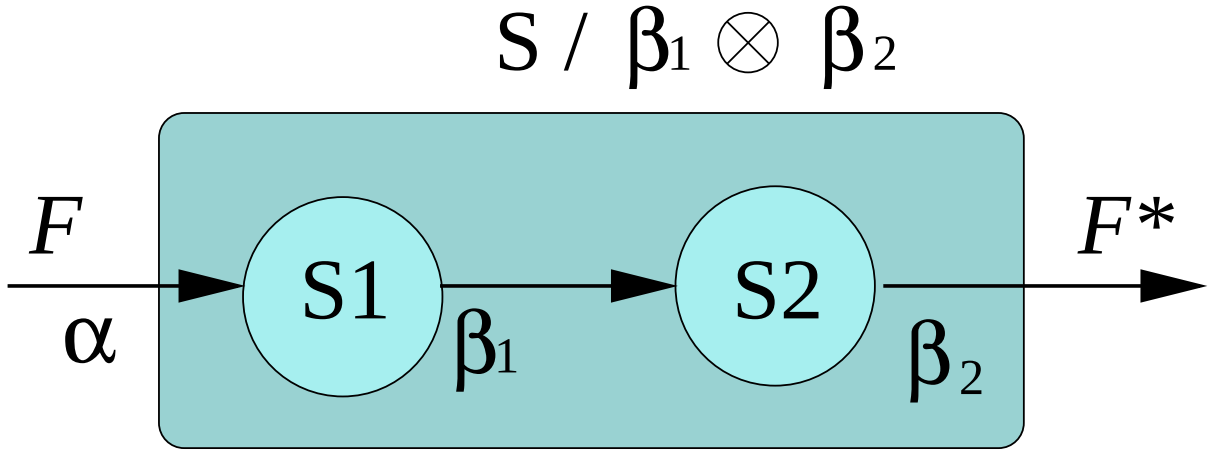
Example:

$$\beta_1 = \beta_{R_1,T_1}$$

$$\beta_2 = \beta_{R_2,T_2}$$

$$\beta_{R_1,T_1} \otimes \beta_{R_2,T_2} = \beta_{\min(R_1,R_2),T_1+T_2}$$

Network Calculus - Concatenation of Nodes



Example:

$$\beta_1 = \beta_{R_1,T_1}$$

$$\beta_2 = \beta_{R_2,T_2}$$

$$\beta_{R_1,T_1} \otimes \beta_{R_2,T_2} = \beta_{\min(R_1,R_2),T_1+T_2}$$

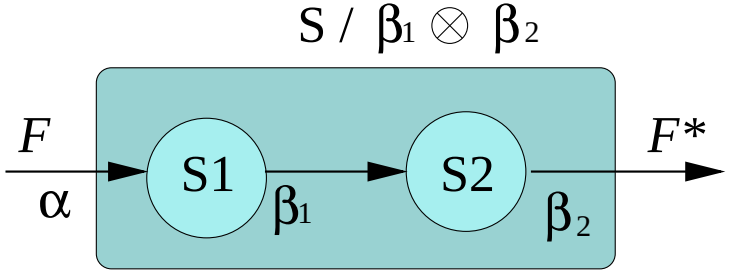
Useful properties:

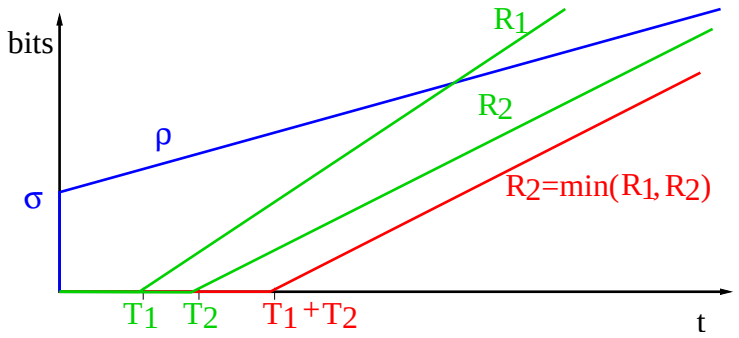
$$f \otimes g = g \otimes f$$

$$(f \otimes g) \otimes h = f \otimes (g \otimes h)$$

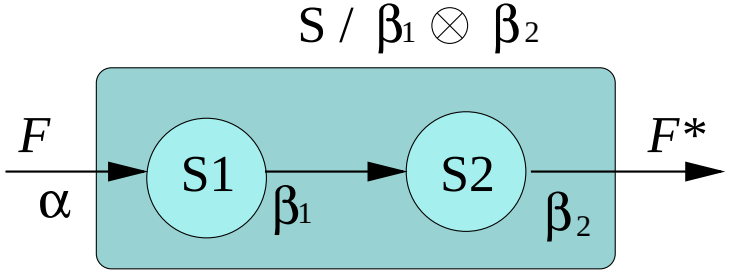
$$(f + c) \otimes g = (f \otimes g) + c \text{ for any constant } c \in \mathbb{R}$$

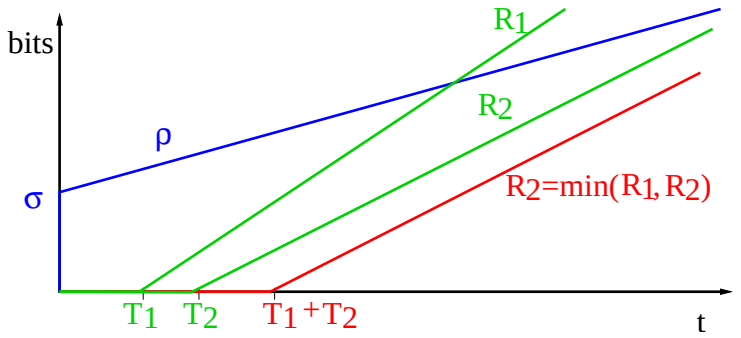
Network Calculus - Pay Bursts Only Once





Network Calculus - Pay Bursts Only Once



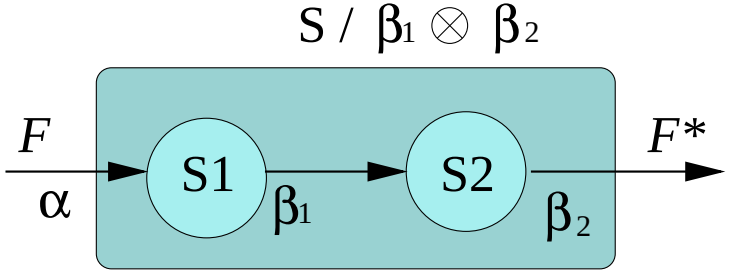


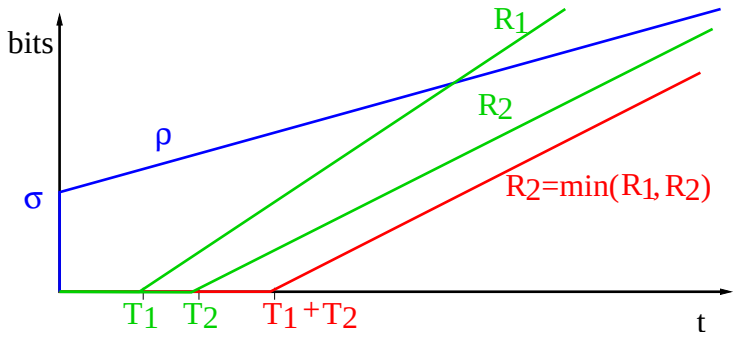
$$\alpha = \gamma_{\rho,\sigma}$$

$$\beta_1 = \beta_{R_1,T_1} = R_1 \max(0, t - T_1)$$

$$\beta_2 = \beta_{R_2,T_2} = R_2 \max(0, t - T_2)$$

Network Calculus - Pay Bursts Only Once





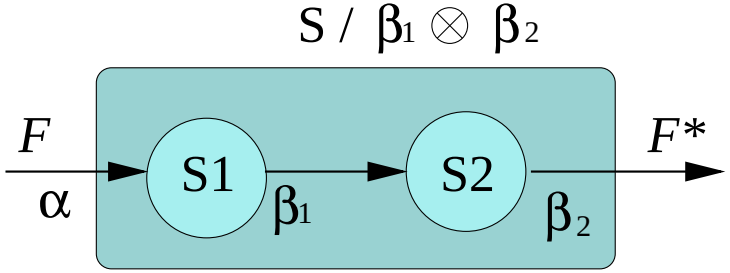
$$\alpha = \gamma_{\rho,\sigma}$$

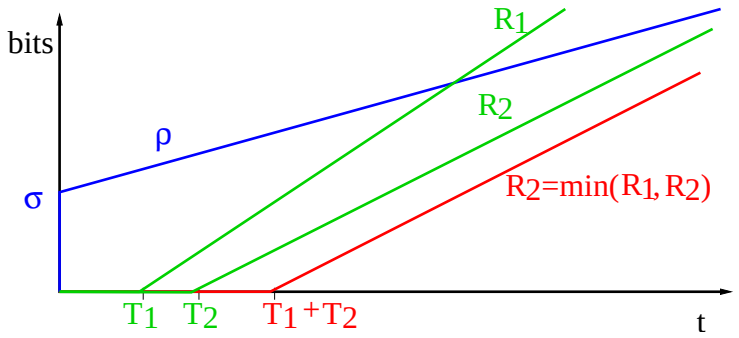
$$\beta_1 = \beta_{R_1,T_1} = R_1 \max(0, t - T_1)$$

$$\beta_2 = \beta_{R_2,T_2} = R_2 \max(0, t - T_2)$$

$$\beta_{R_1,T_1} \otimes \beta_{R_2,T_2} = \beta_{\min(R_1,R_2),T_1+T_2} = \min(R_1, R_2) \max(0, t - (T_1 + T_2))$$

Network Calculus - Pay Bursts Only Once





$$\alpha = \gamma_{\rho,\sigma}$$

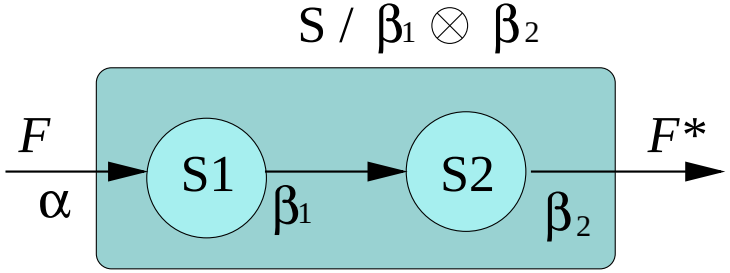
$$\beta_1 = \beta_{R_1,T_1} = R_1 \max(0, t - T_1)$$

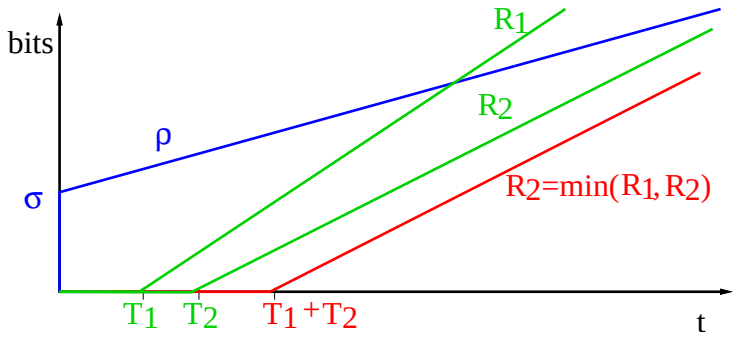
$$\beta_2 = \beta_{R_2,T_2} = R_2 \max(0, t - T_2)$$

$$\beta_{R_1,T_1} \otimes \beta_{R_2,T_2} = \beta_{\min(R_1,R_2),T_1+T_2} = \min(R_1, R_2) \max(0, t - (T_1 + T_2))$$

$$D_1 + D_2 = \frac{\sigma}{R_1} + \frac{\sigma}{R_2} + \frac{\rho T_1}{R_2} + T_1 + T_2$$

Network Calculus - Pay Bursts Only Once





$$\alpha = \gamma_{\rho,\sigma}$$

$$\beta_1 = \beta_{R_1,T_1} = R_1 \max(0, t - T_1)$$

$$\beta_2 = \beta_{R_2,T_2} = R_2 \max(0, t - T_2)$$

$$\beta_{R_1,T_1} \otimes \beta_{R_2,T_2} = \beta_{\min(R_1,R_2),T_1+T_2} = \min(R_1, R_2) \max(0, t - (T_1 + T_2))$$

$$D_1 + D_2 = \frac{\sigma}{R_1} + \frac{\sigma}{R_2} + \frac{\rho T_1}{R_2} + T_1 + T_2$$

$$D_S = \frac{\sigma}{\min(R_1, R_2)} + T_1 + T_2$$

Overview

Topology and Structure

The Network Layer and the Switch

TDM Allocation for Quality of Service

Regulated Flows

Data Protection

Clocking

Dynamic Voltage Scaling

Network Simulator

Data Protection

- Two level protection: Link layer and transport layer
- Data link layer protection:
  - ⋆ SEC-DED header protection (16/26 bits)
  - ⋆ Four levels of payload protection:
    - ∗ Maximum bandwidth - no protection (102/102 bits)
    - ∗ Guaranteed integrity - DED protection (90/102 bits)
    - ∗ Minimum latency - SEC protection (90/102 bits)
    - ∗ High reliability - SEC-DED protection (81/102 bits)
  - ⋆ Parity based codes used (Hamming or Hsiao codes) to allow for low logic depth implementations
- Transport layer:
  - ⋆ Normal mode: Send-and-Forget (SaF) service
  - ⋆ Reliability mode: Acknowledgement-and-Retransmit (AaR) service
    - ∗ window size $N, 1 \leq N \leq 64$
    - ∗ $2N$ packets are buffered in sender and receiver
    - ∗ End-to-end flow control mechanism
- in total 8 modes available

Error Protection for Low Power

Scenario I:

- 8×8 network
- 80 bits payload
- 15 bits header

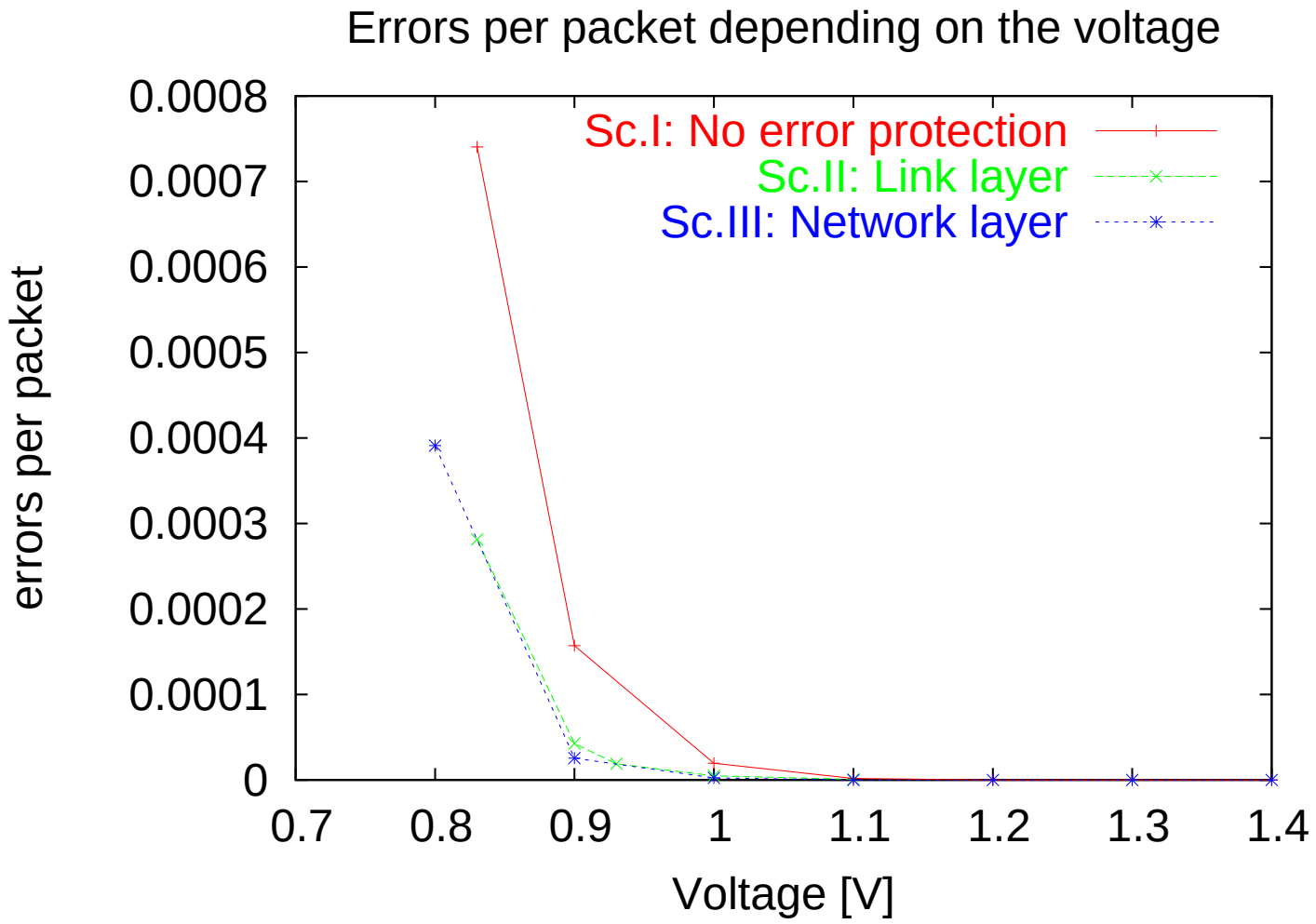
Scenario II: Link layer error protection

- Block code with DED/SEC capability
- 20 payload bits and 5 protection bits per block;
- 80 payload bits
- 15 header bits
- 30 protecting bits
- 125 total bits

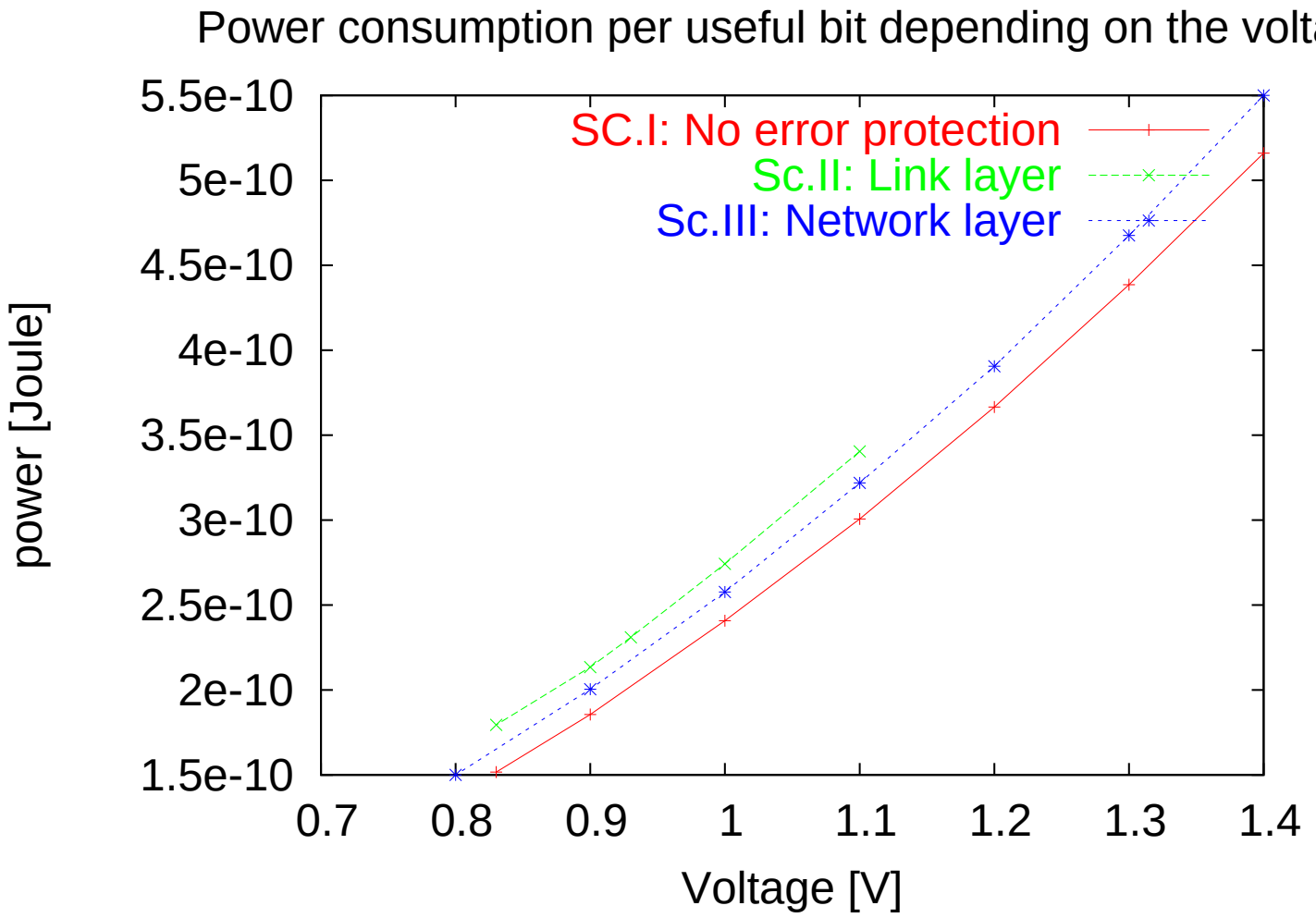
Scenario III: End-to-end protection

- Header is protected at the link layer as in Scenario II
- Payload is protected by a block code with SEC/DED capability
- 80 payload bits
- 15 header bits
- 24 protecting bits
- 119 total bits

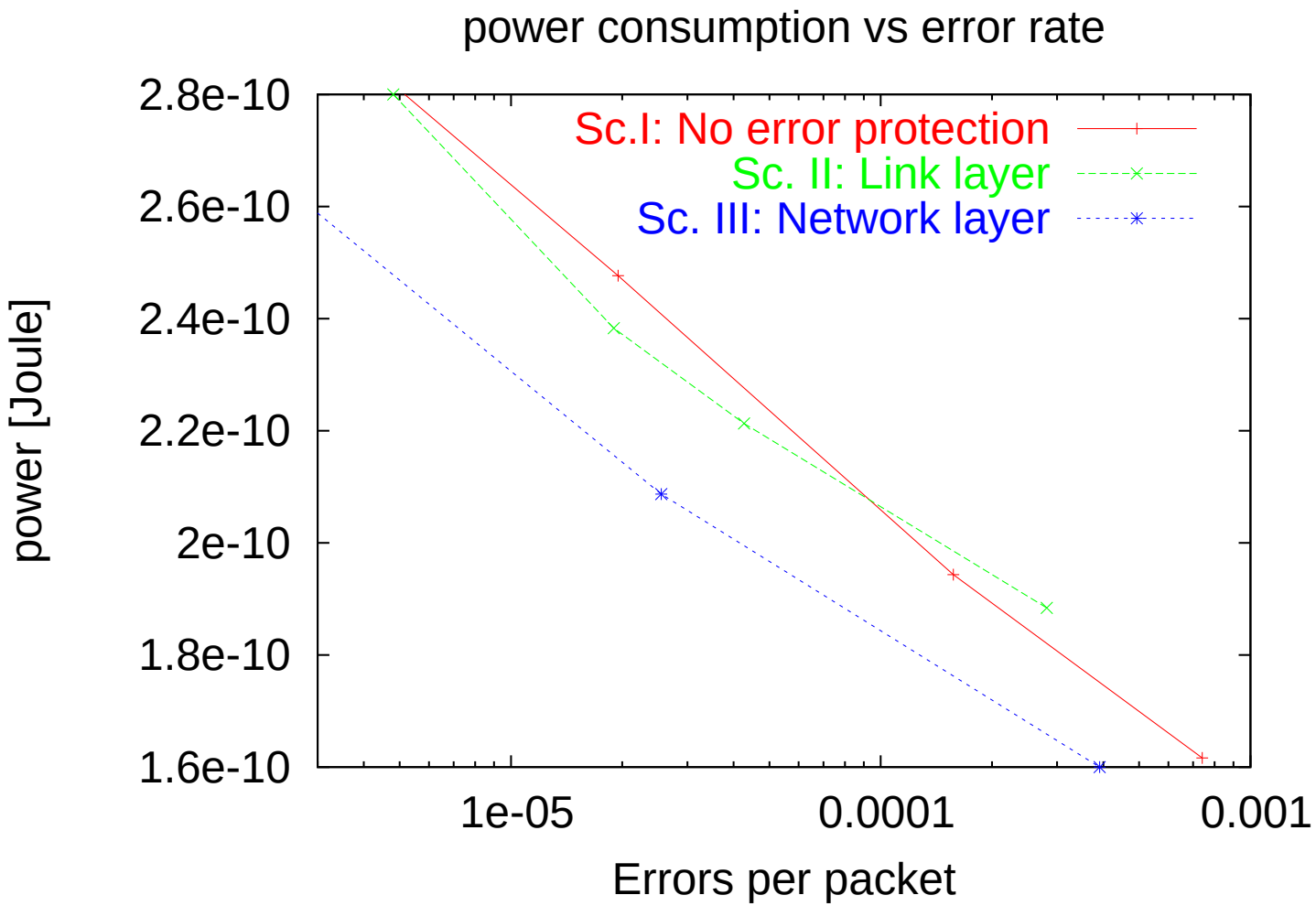
Errors per Packet



Power Consumption per Useful Bit



Power Consumption vs. Error Rate



Low Power encoding - Conclusion

- Low power bus encoding is of limited value and probably increases the overall power consumption.
- Link-level error protection to allow for lower voltage does not give significant improvements.
- End-to-end data protection decreases power consumption for 8×8 networks, with slowly increasing gain for larger networks.

Overview

Topology and Structure

The Network Layer and the Switch

TDM Allocation for Quality of Service

Regulated Flows

Data Protection

Clocking

Dynamic Voltage Scaling

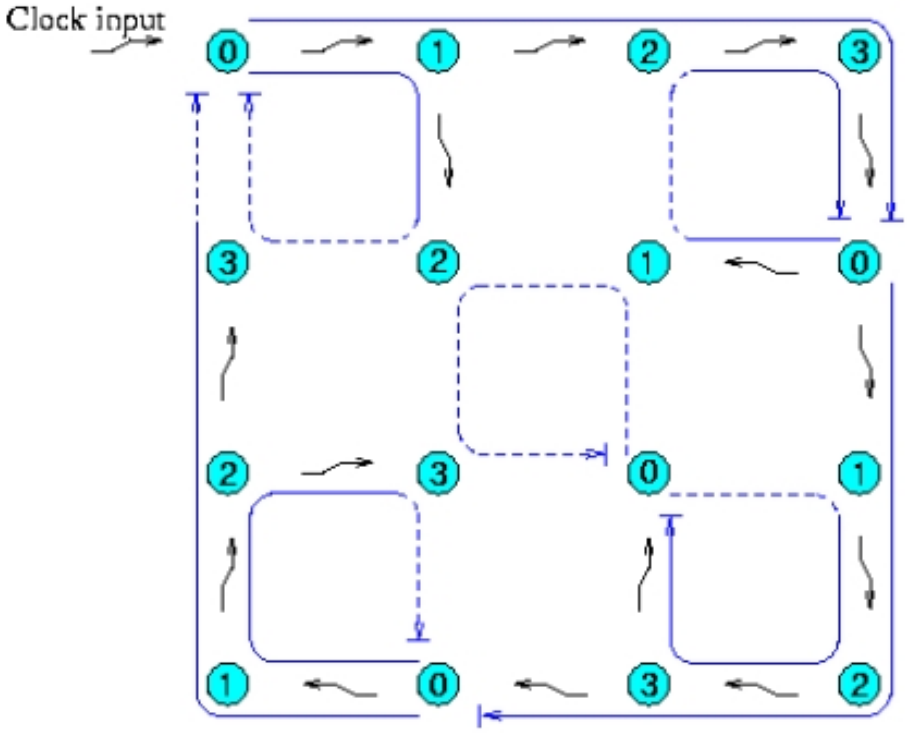
Network Simulator

Globally Pseudosynchronous - Locally Synchronous Clocking

Every switch uses same frequency; phase difference is constant and known.

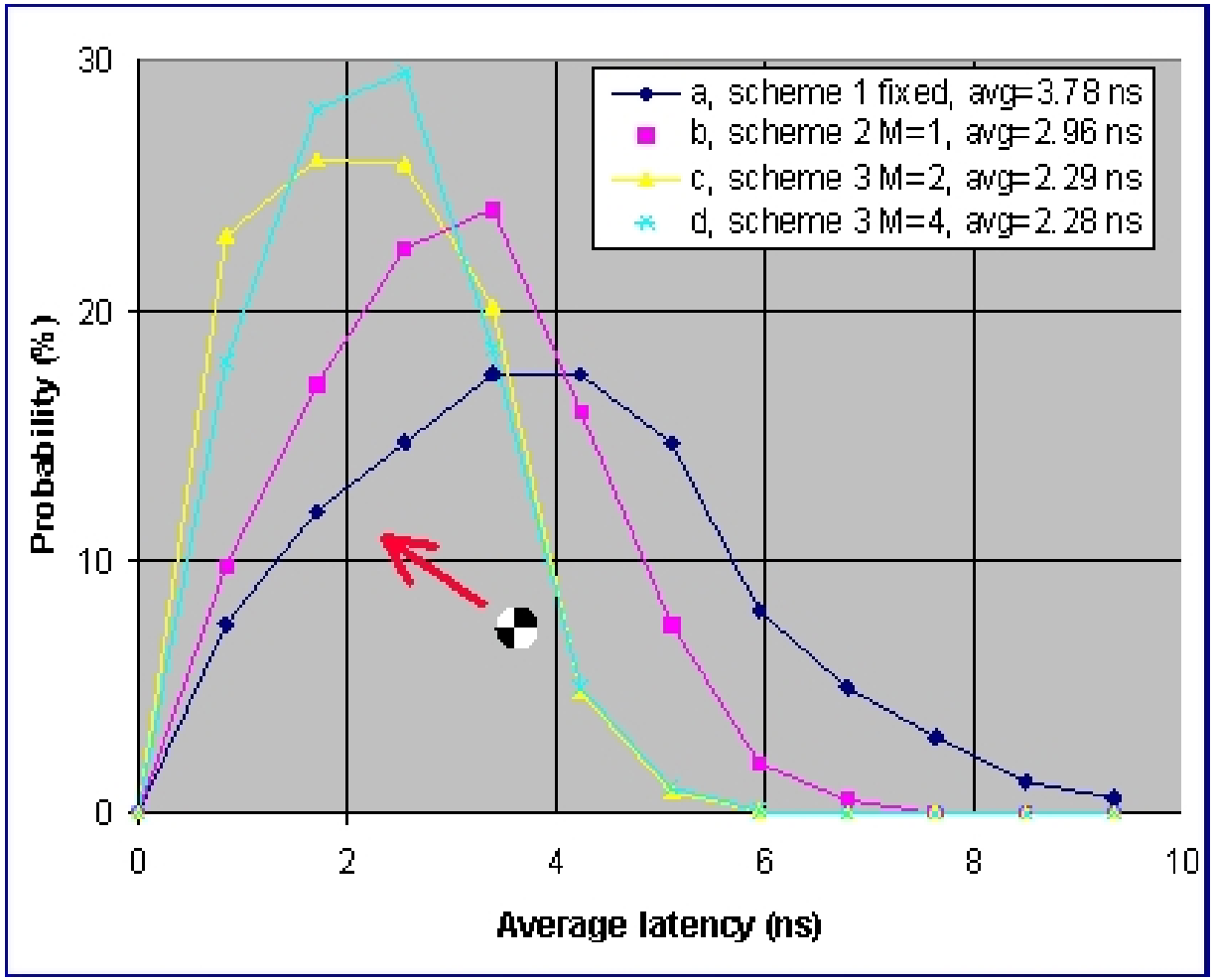
- Latency reduce with 29% at low load; 40% at high load
- Can handle 10% higher load
- More skew tolerant
- Clock skew and jitter is depending only on local constraints
- No global clock distribution with associated power gains
- Reduced peak power with 50% at best
- Jitter reduced significantly

Globally Pseudosynchronous Clocking - cont'd



- Downstream data create low latency paths (Data Motorways)
  - ⋆ Guaranteed data motorways
  - ⋆ Phase related data motorways
- Periphery roundtrip:
  - ⋆ 14 cycles downstream
  - ⋆ 21 cycles upstream
  - ⋆ 24 cycles synchronous

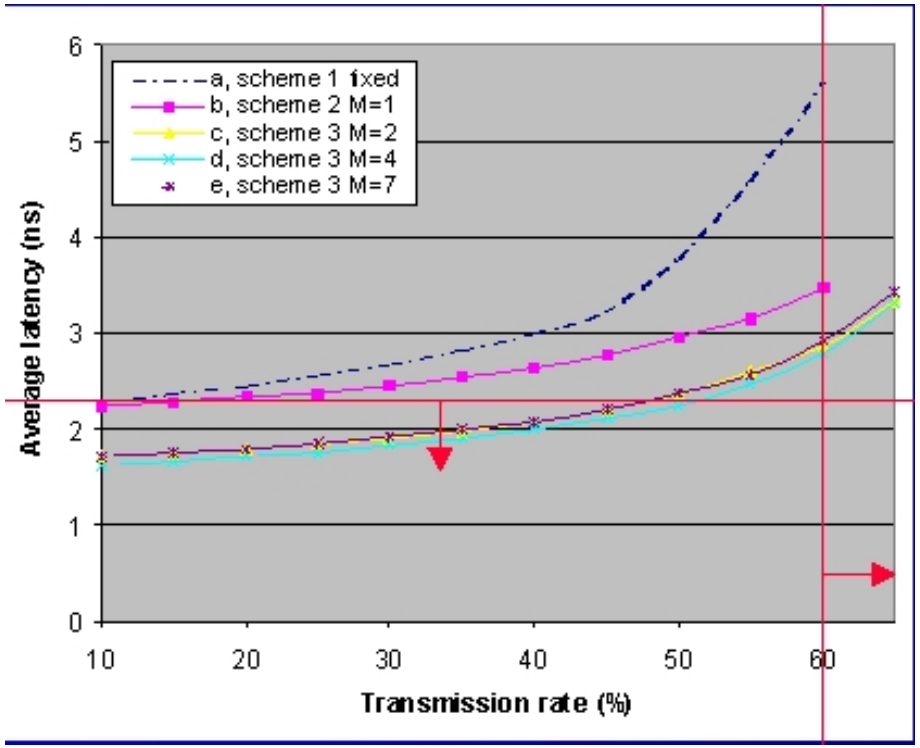
Globally Pseudosynchronous Clocking - cont'd

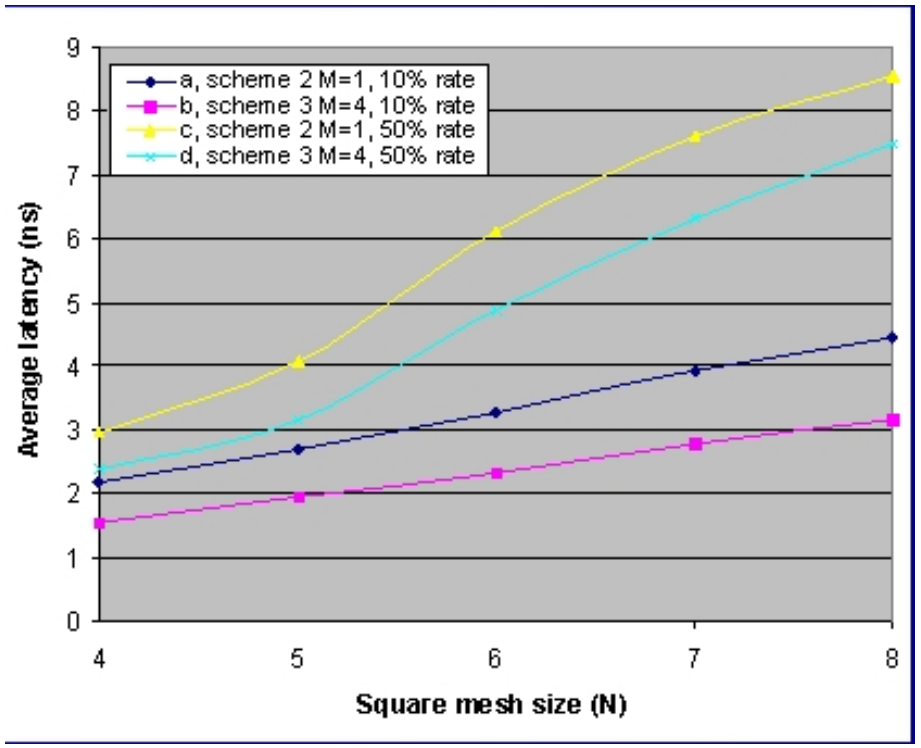


4x4 network, 50% emmision rate

- (a) Synchronous clocking, uniform routing
- (b) Synchronous clocking, centrifugal routing
- (c) pseudosynchronous clocking, 2 phases
- (d) pseudosynchronous clocking, 4 phases

Globally Pseudosynchronous Clocking - cont'd





Overview

Topology and Structure

The Network Layer and the Switch

TDM Allocation for Quality of Service

Regulated Flows

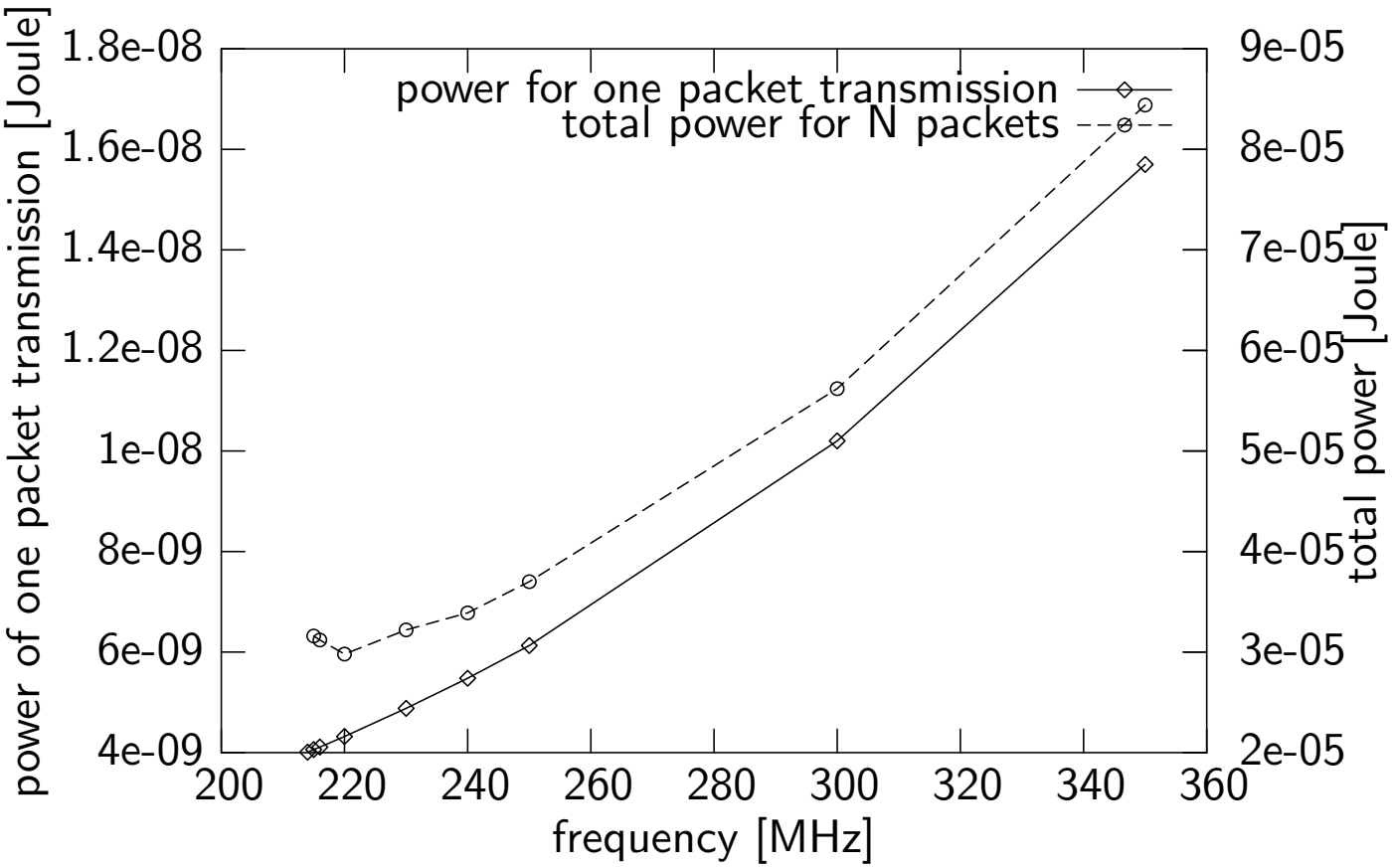
Data Protection

Clocking

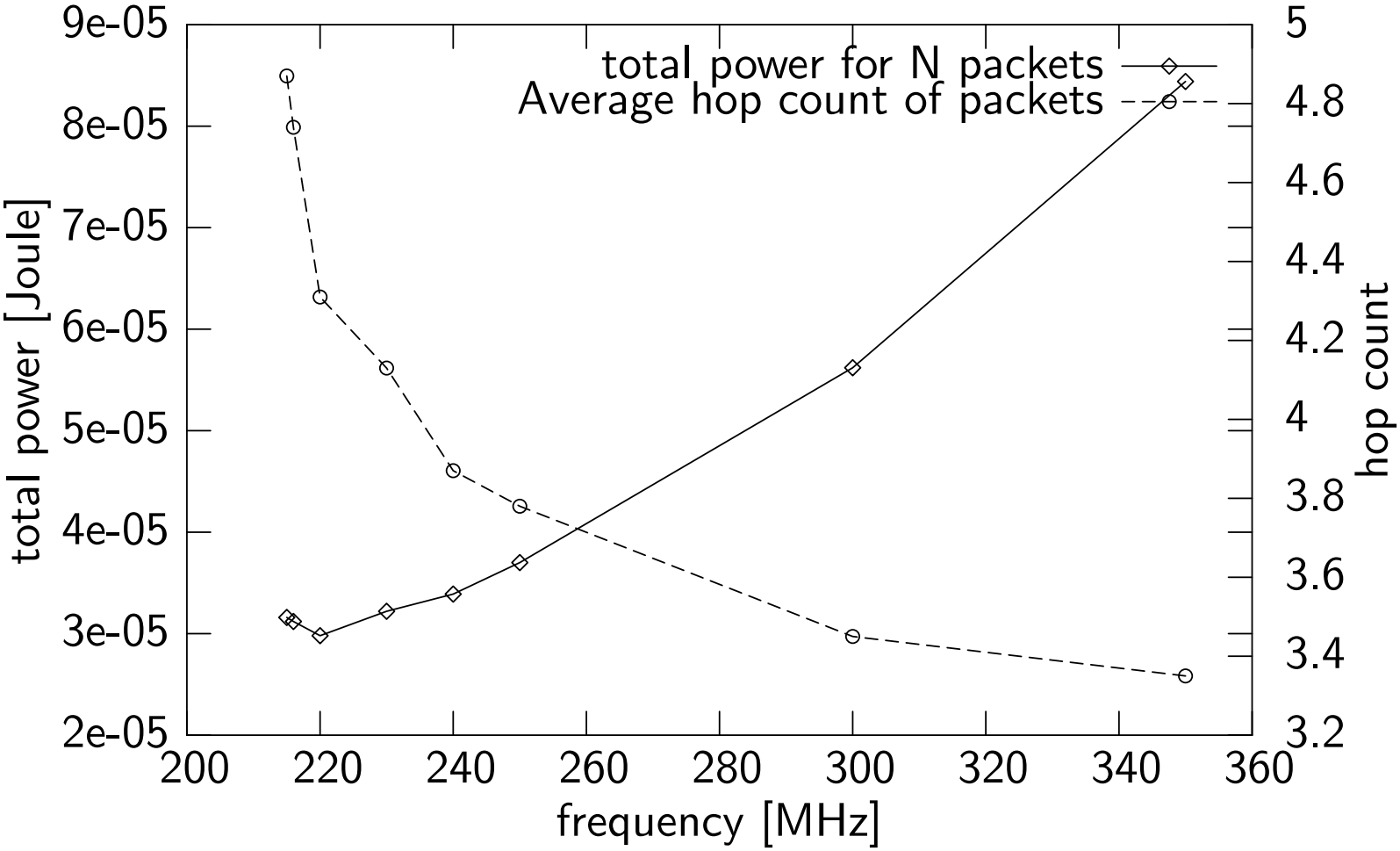
Dynamic Voltage Scaling

Network Simulator

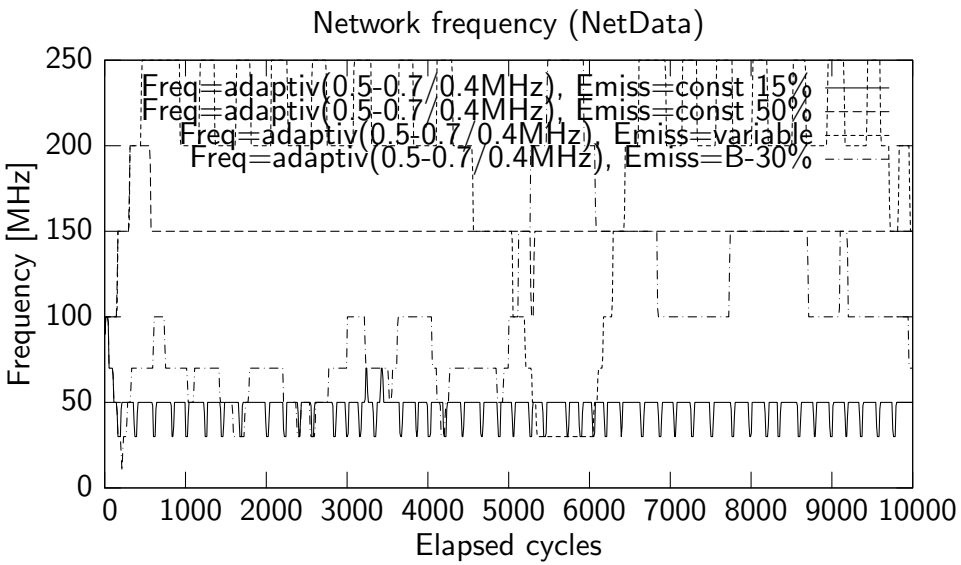
Potential of Dynamic Voltage Scaling - Power

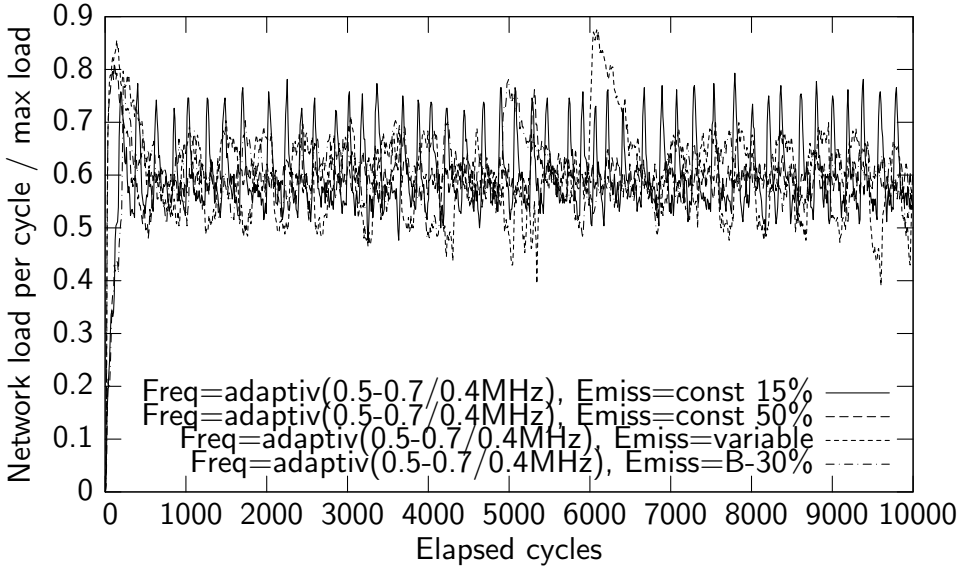


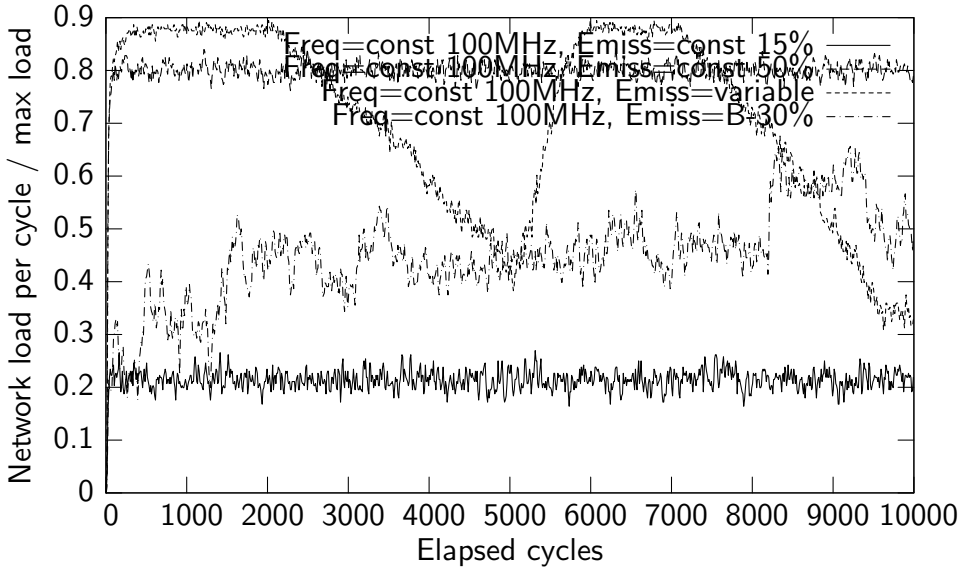
Potential of Dynamic Voltage Scaling - Hops



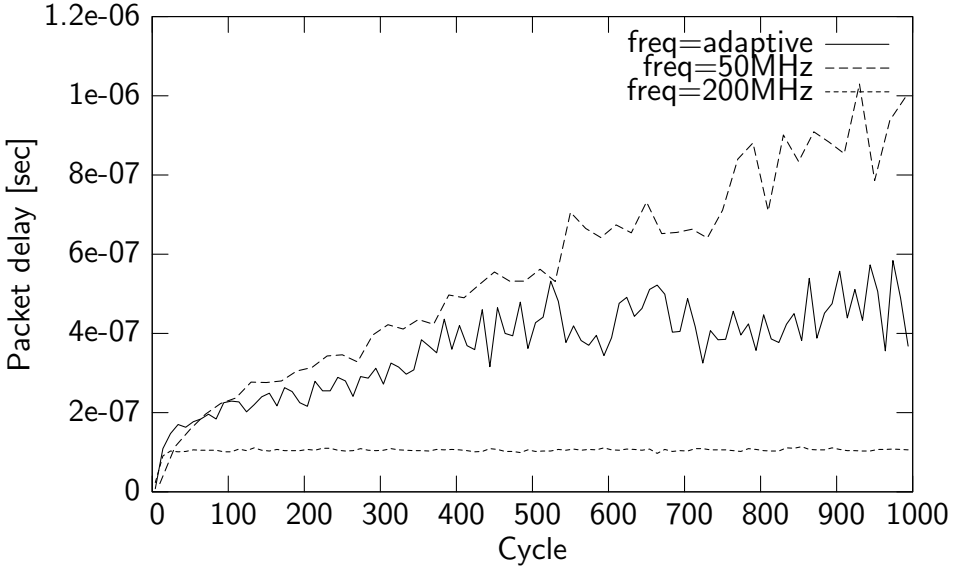
Potential of Dynamic Voltage Scaling - Load variations



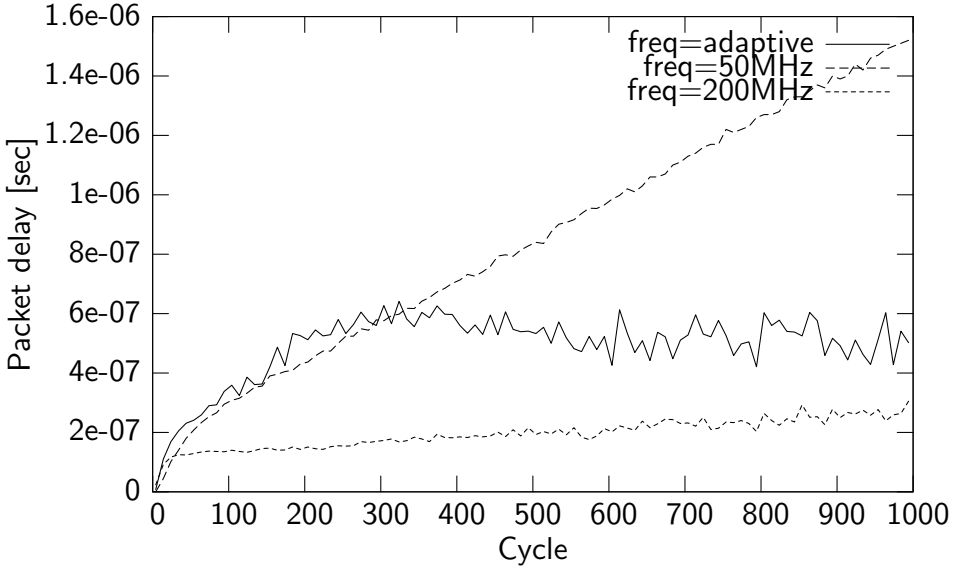




Potential of Dynamic Voltage Scaling - Delay

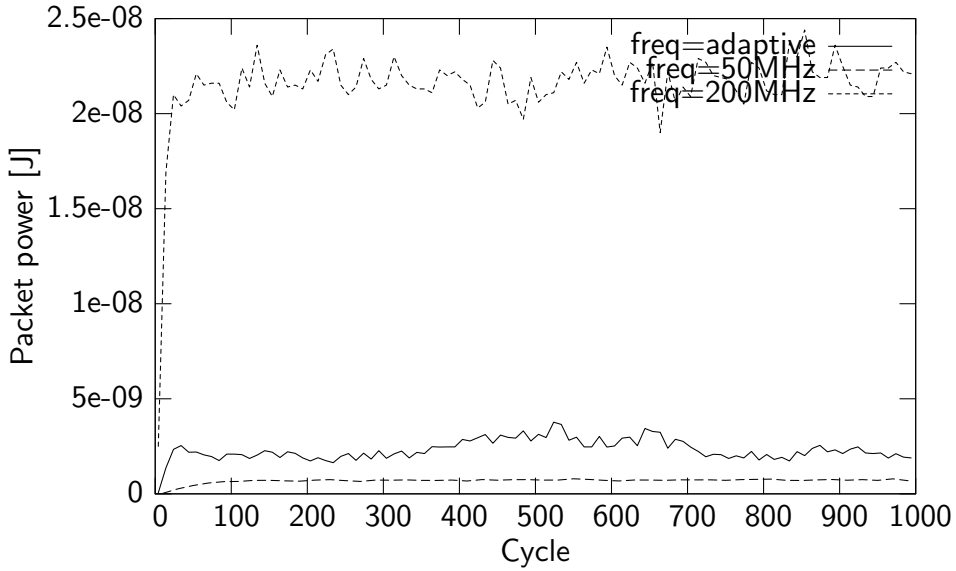


Emission probability = 30%

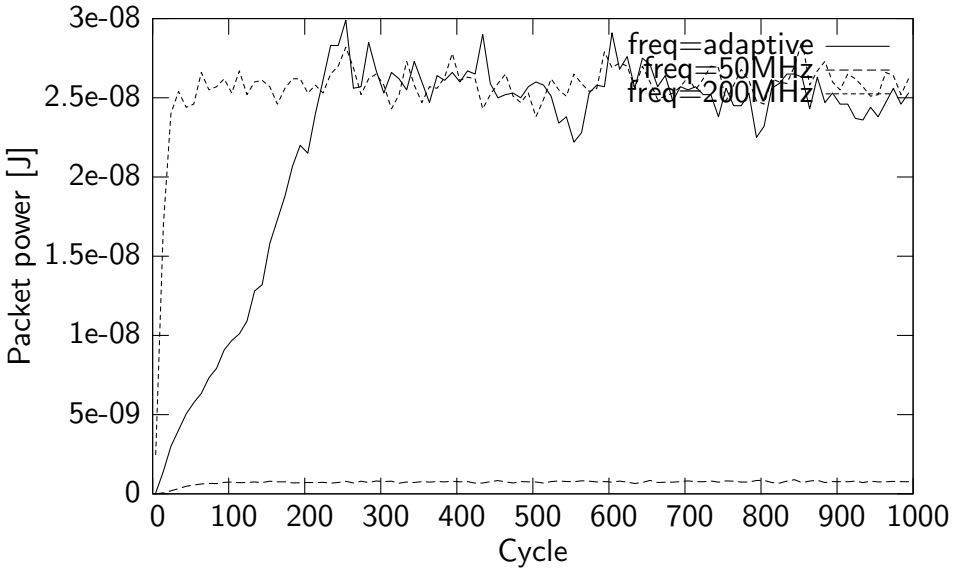


Emission probability = 70%

Potential of Dynamic Voltage Scaling - Power



Emission probability = 30%



Emission probability = 70%

Overview

Topology and Structure

The Network Layer and the Switch

TDM Allocation for Quality of Service

Regulated Flows

Data Protection

Clocking

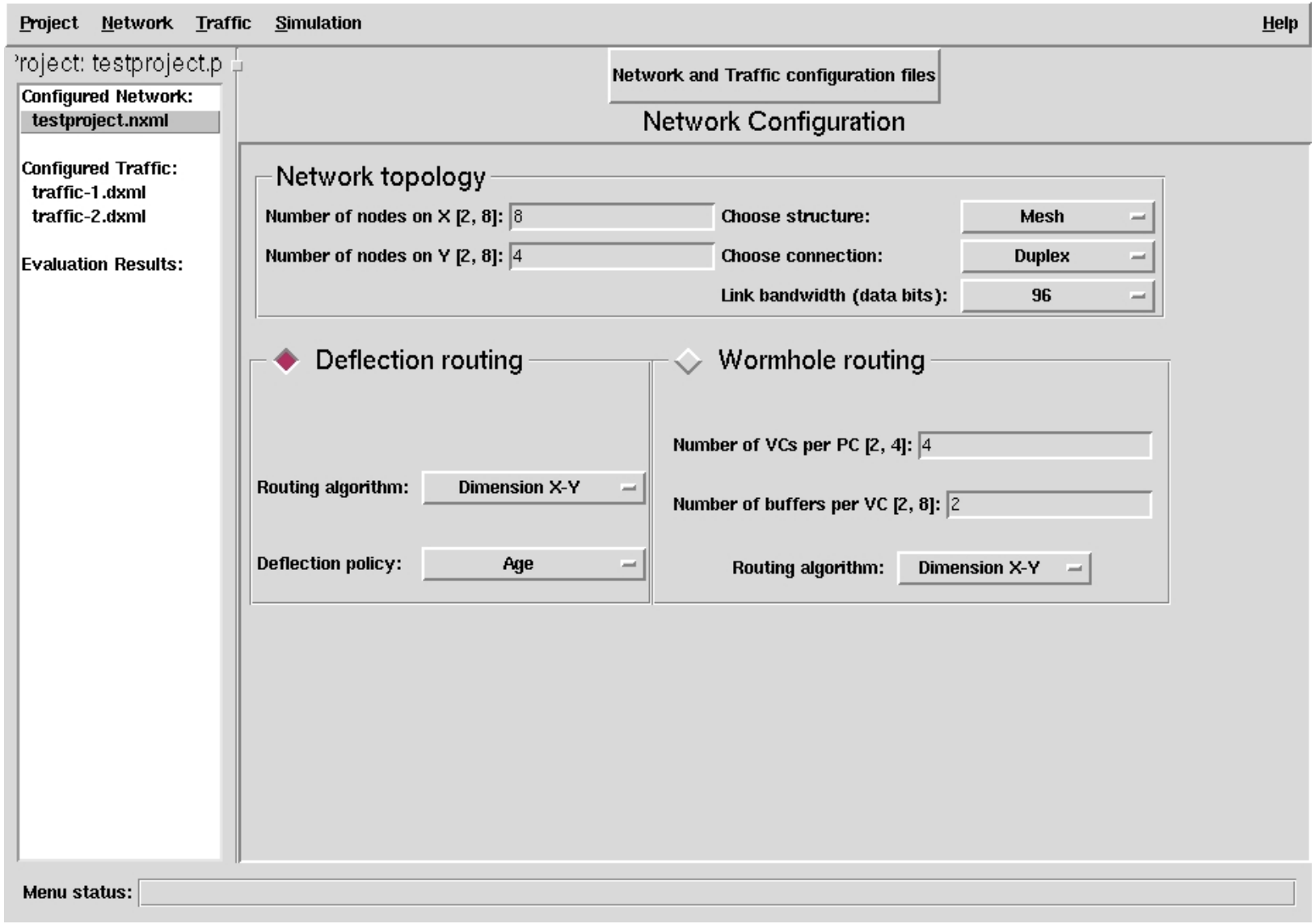
Dynamic Voltage Scaling

Network Simulator

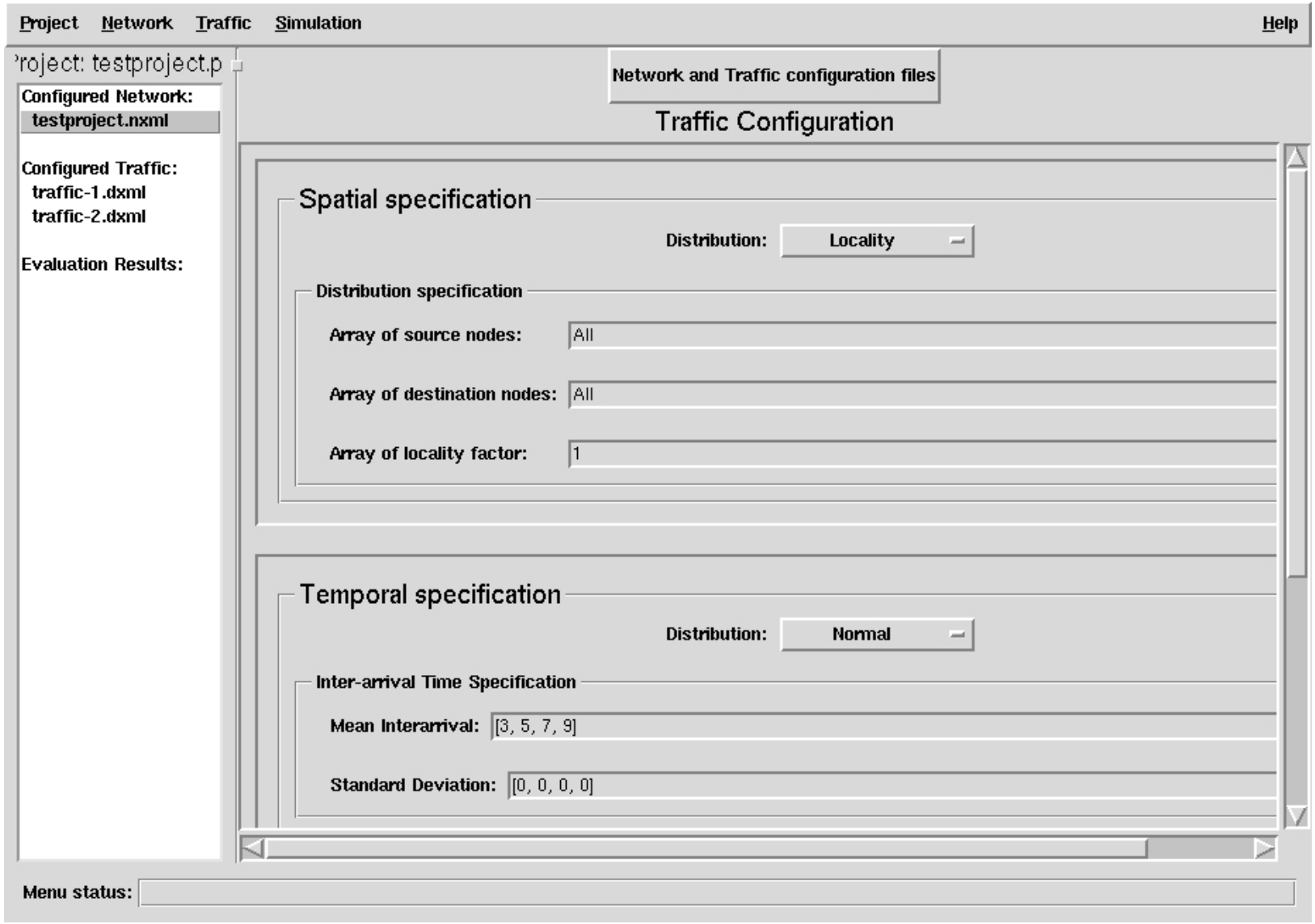
Nostrum Simulation Environment

- Nostrum NoC Simulation Environment (NNSE)
- Based on SystemC simulator
- Configuration paramaters:
  - ⋆ Size
  - ⋆ Topology (Mesh, Torus)
  - ⋆ Switching and routing (deflective, wormhole)
  - ⋆ Traffic pattern (temporal, spatial, random, locality, per channel, ...)
  - ⋆ Analysis plots (delay, load, power, ...)
- Useful to analyse the zillion trade-offs in NoC design.

NNSE Network Configuration



NNSE Traffic Configuration



Summary of Nostrum Status

- Nostrum defines a 2 D mesh topology;
- Protocol stack for link layer, network layer and session layer;
- Packet switched and virtual circuit communication services;
- Buffer-less, loss-less switch with no routing tables;
- 2 level data protection scheme;
- Session layer communication primitives;
- Flexible NoC Simulator;

Further information: www.imit.kth.se/info/FOFU/Nostrum/