Scheduling in Networks

Jean Walrand

EECS

University of California, Berkeley

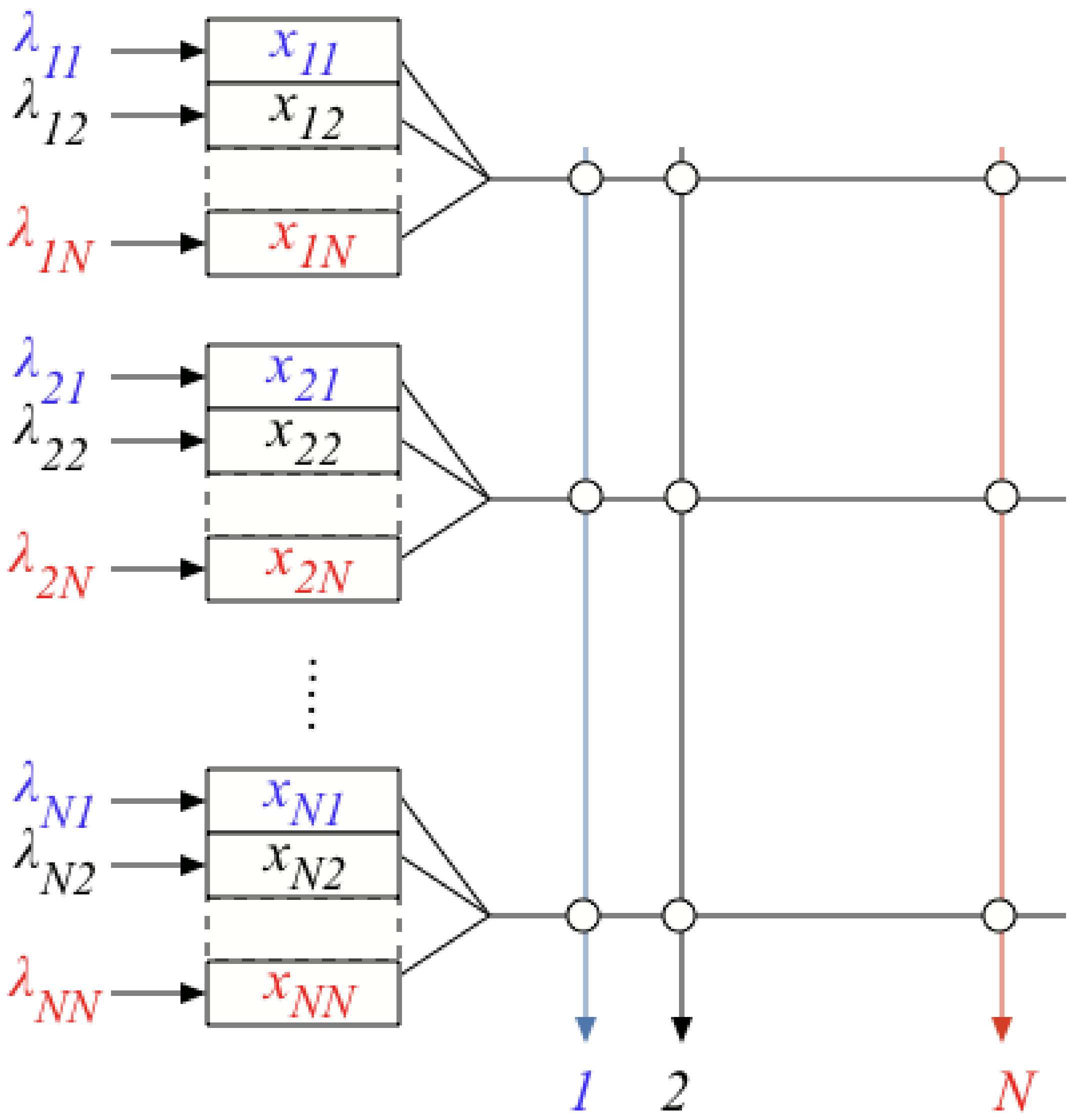
ITC, San Francisco, 9/2011

Ref: Jiang-Walrand: Scheduling and Congestion Control for Wireless and Processing Networks. Morgan-Claypool 2010

Outline

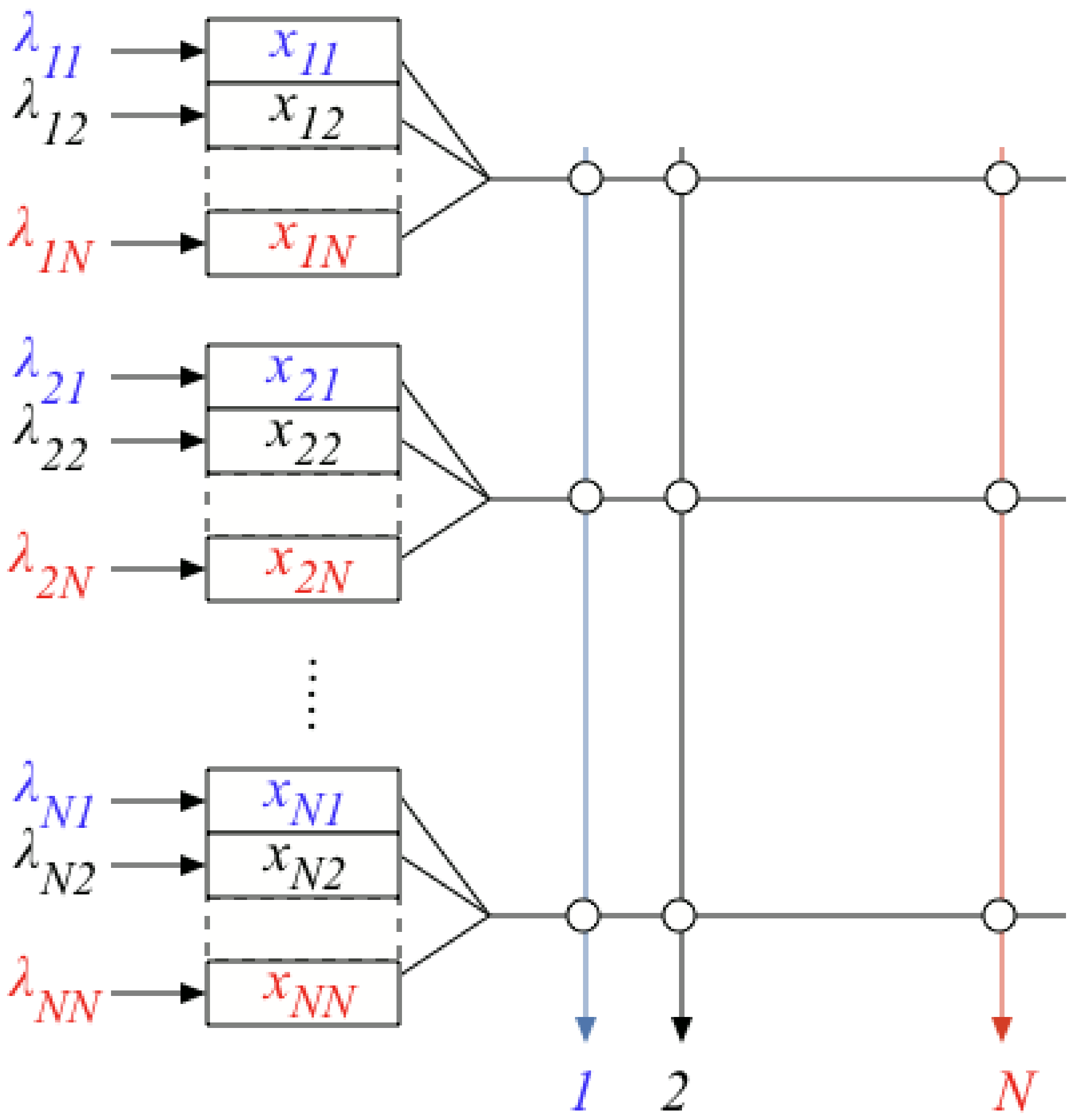
* Example 1: Switch
* Example 2: Ad Hoc Network
* Stability
* Delays
* Status
* Conclusions

Switch



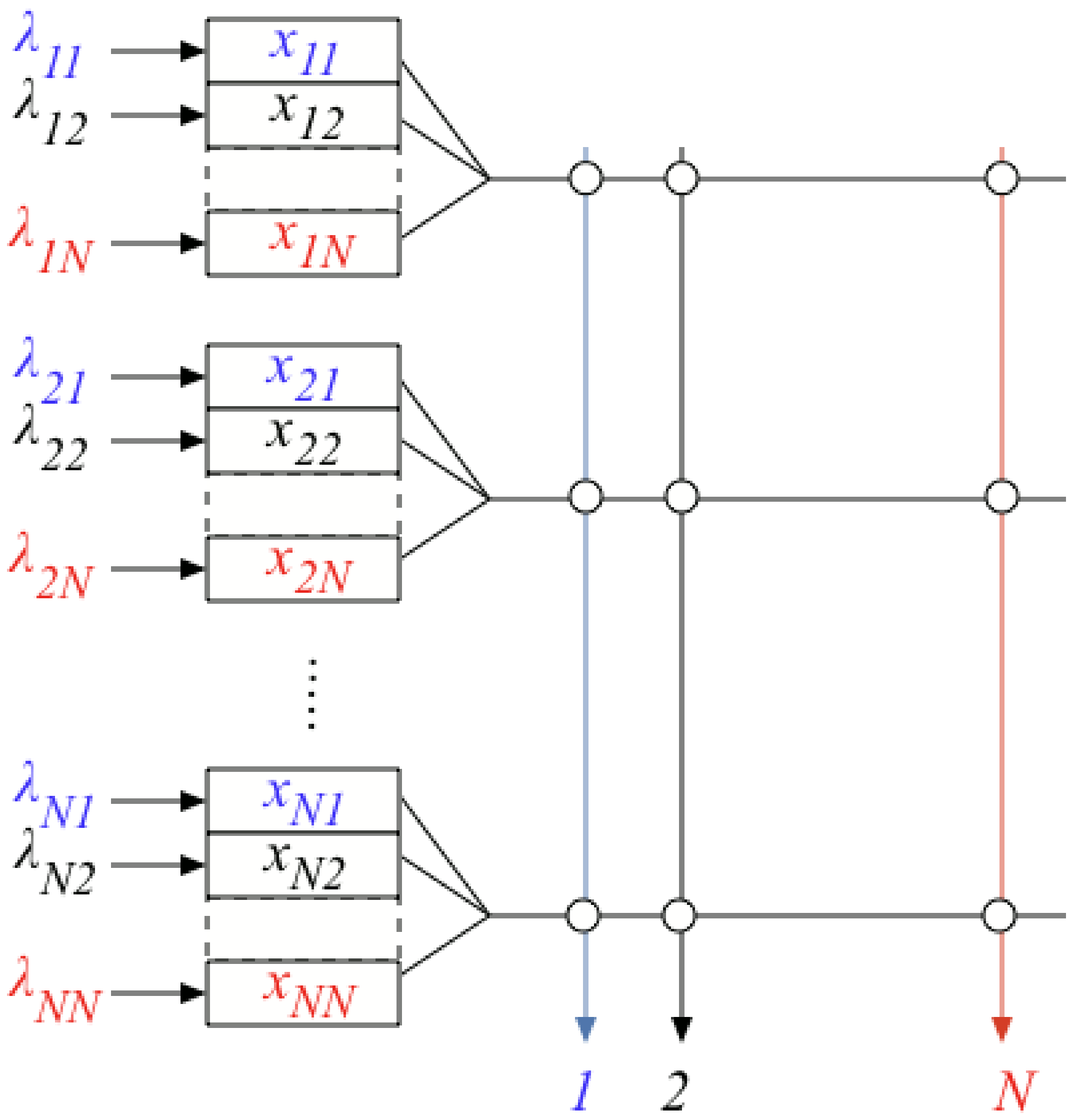
λ_{11}
λ_{12}
λ_{1N}
λ_{21}
λ_{22}
λ_{2N}
λ_{N1}
λ_{N2}
λ_{NN}
x_{11}
x_{12}
x_{1N}
x_{21}
x_{22}
x_{2N}
x_{N1}
x_{N2}
x_{NN}
1
2
N

Switch



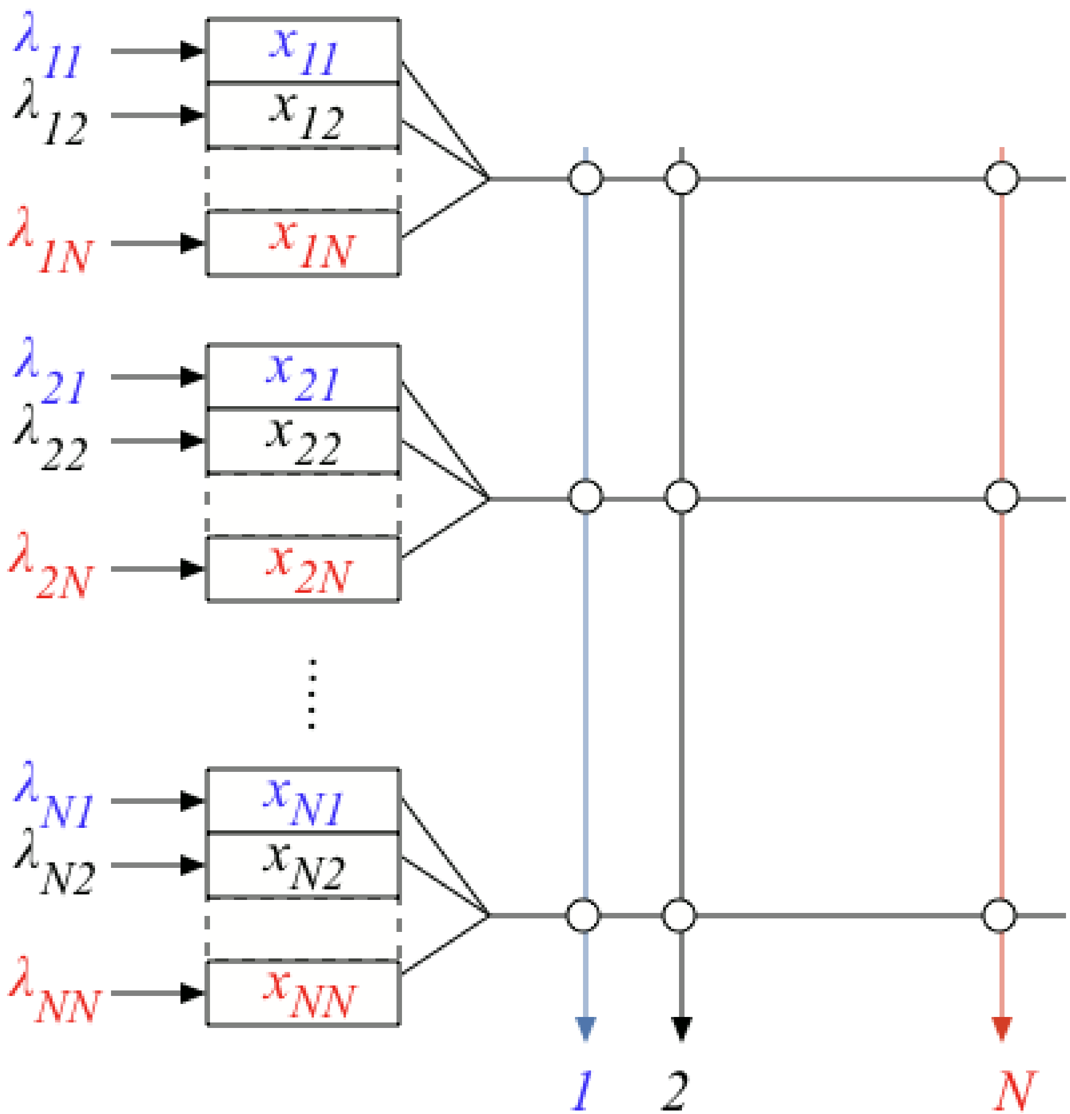
λ_{11}
λ_{12}
λ_{1N}
x_{11}
x_{12}
x_{1N}
λ_{21}
λ_{22}
λ_{2N}
x_{21}
x_{22}
x_{2N}
λ_{N1}
λ_{N2}
λ_{NN}
x_{N1}
x_{N2}
x_{NN}
1
2
N

Switch



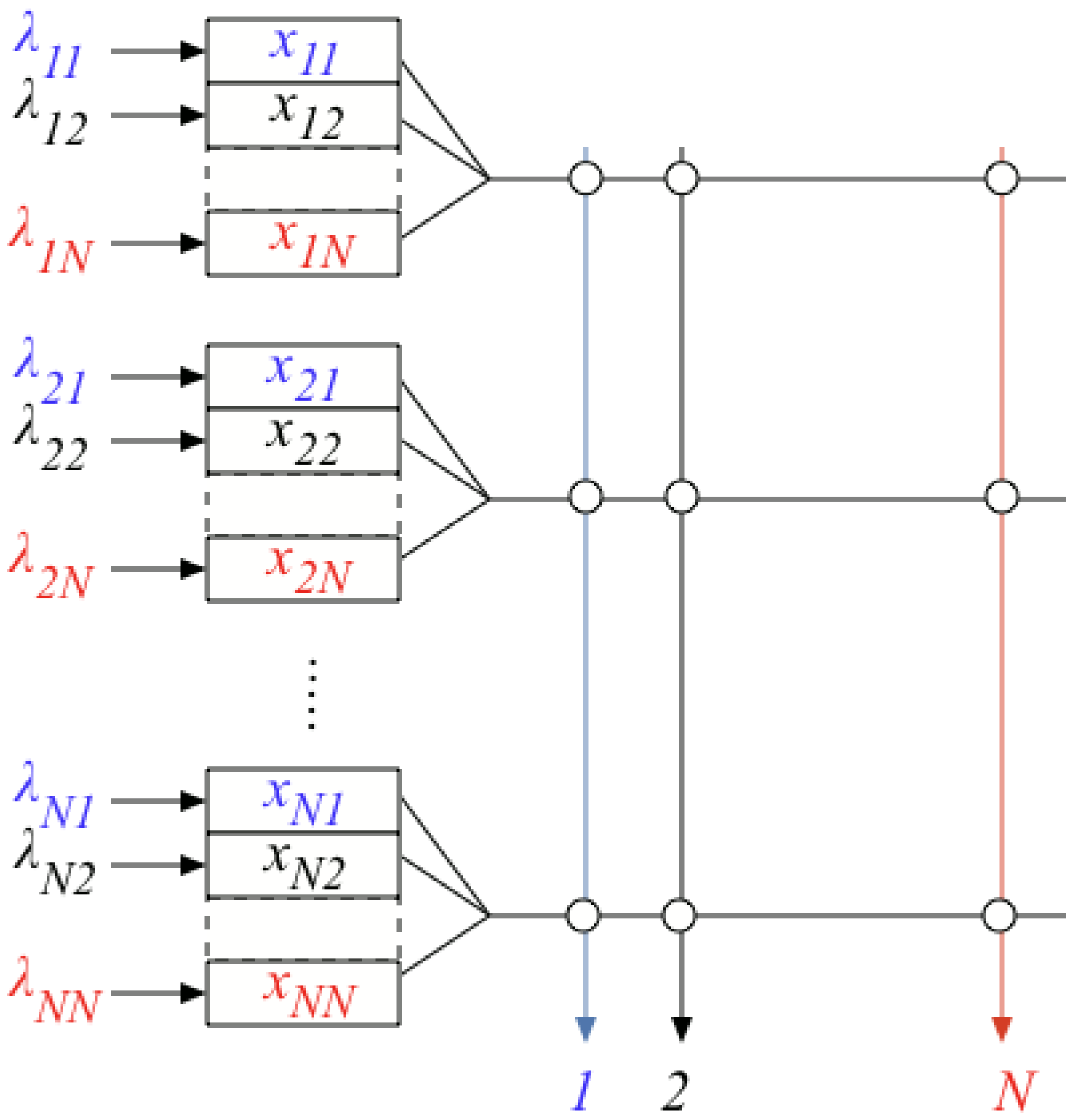
λ11
x11
λ12
x12
λ1N
x1N
λ21
x21
λ22
x22
λ2N
x2N
λN1
xN1
λN2
xN2
λNN
xNN
1
2
N

Switch



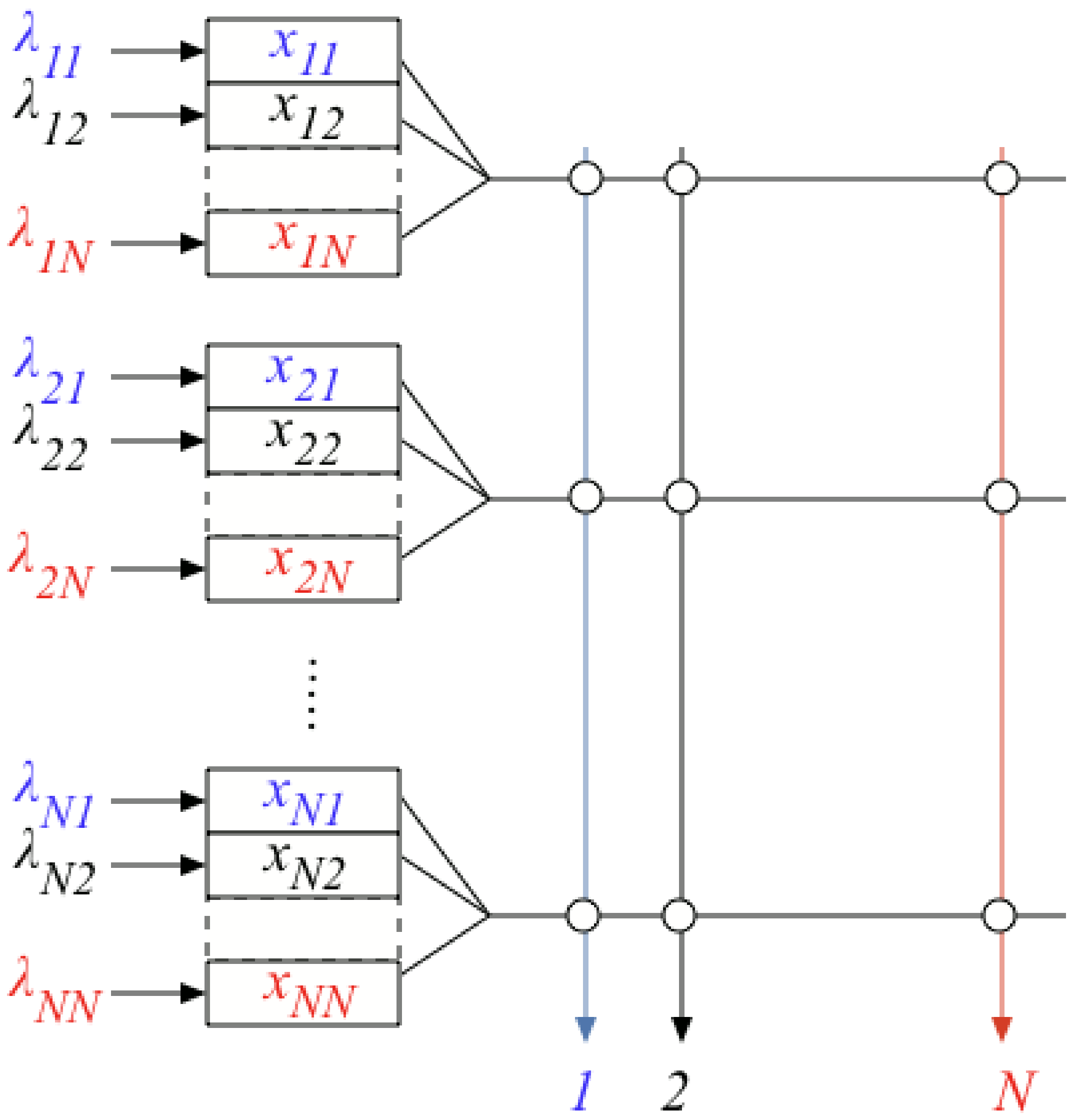
λ11
λ12
λ1N
x11
x12
x1N
λ21
λ22
λ2N
x21
x22
x2N
λN1
λN2
λNN
xN1
xN2
xNN
1
2
N

Switch



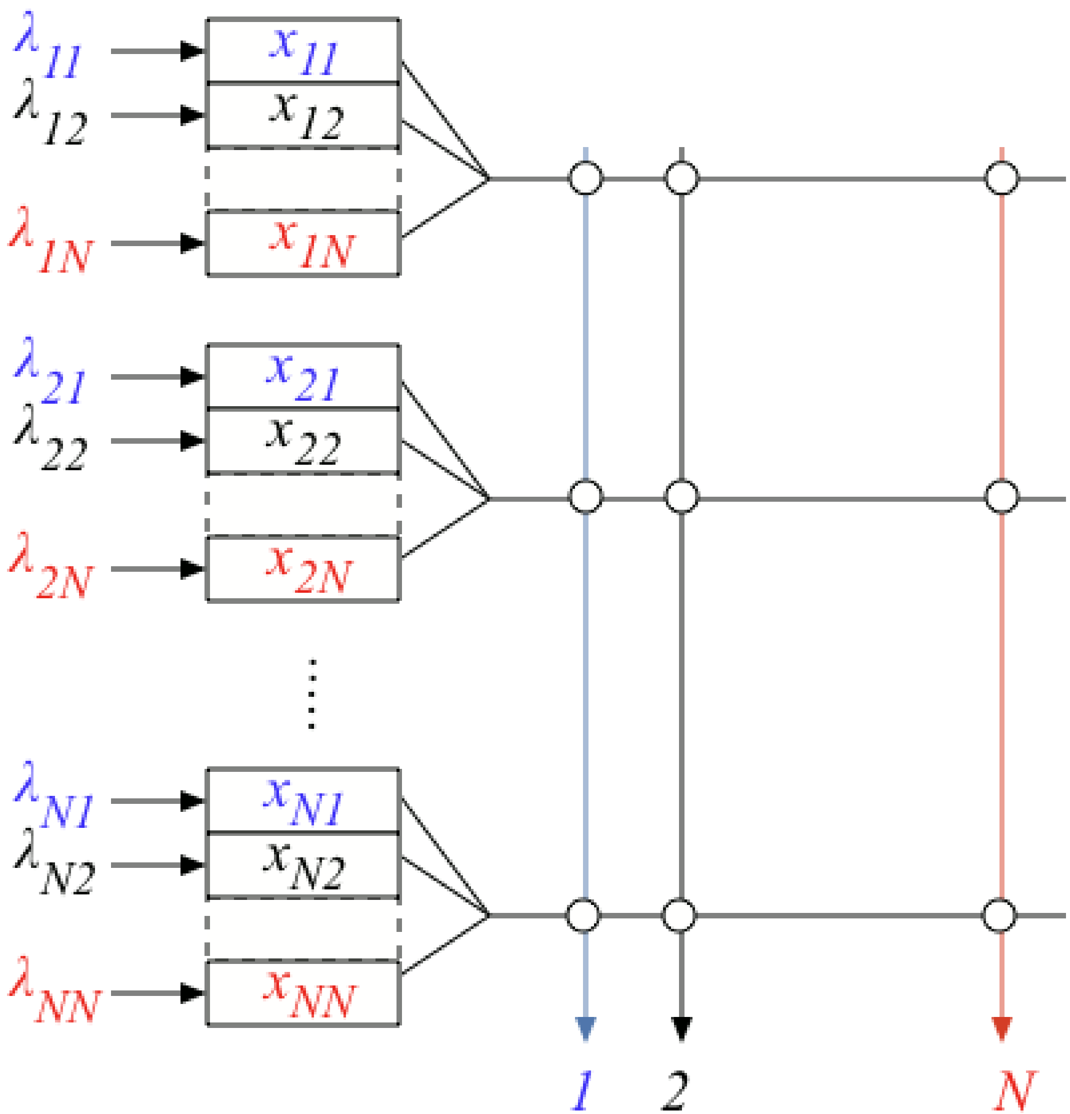
λ_{11}
λ_{12}
λ_{1N}
x_{11}
x_{12}
x_{1N}
λ_{21}
λ_{22}
λ_{2N}
x_{21}
x_{22}
x_{2N}
λ_{N1}
λ_{N2}
λ_{NN}
x_{N1}
x_{N2}
x_{NN}
1
2
N

Switch



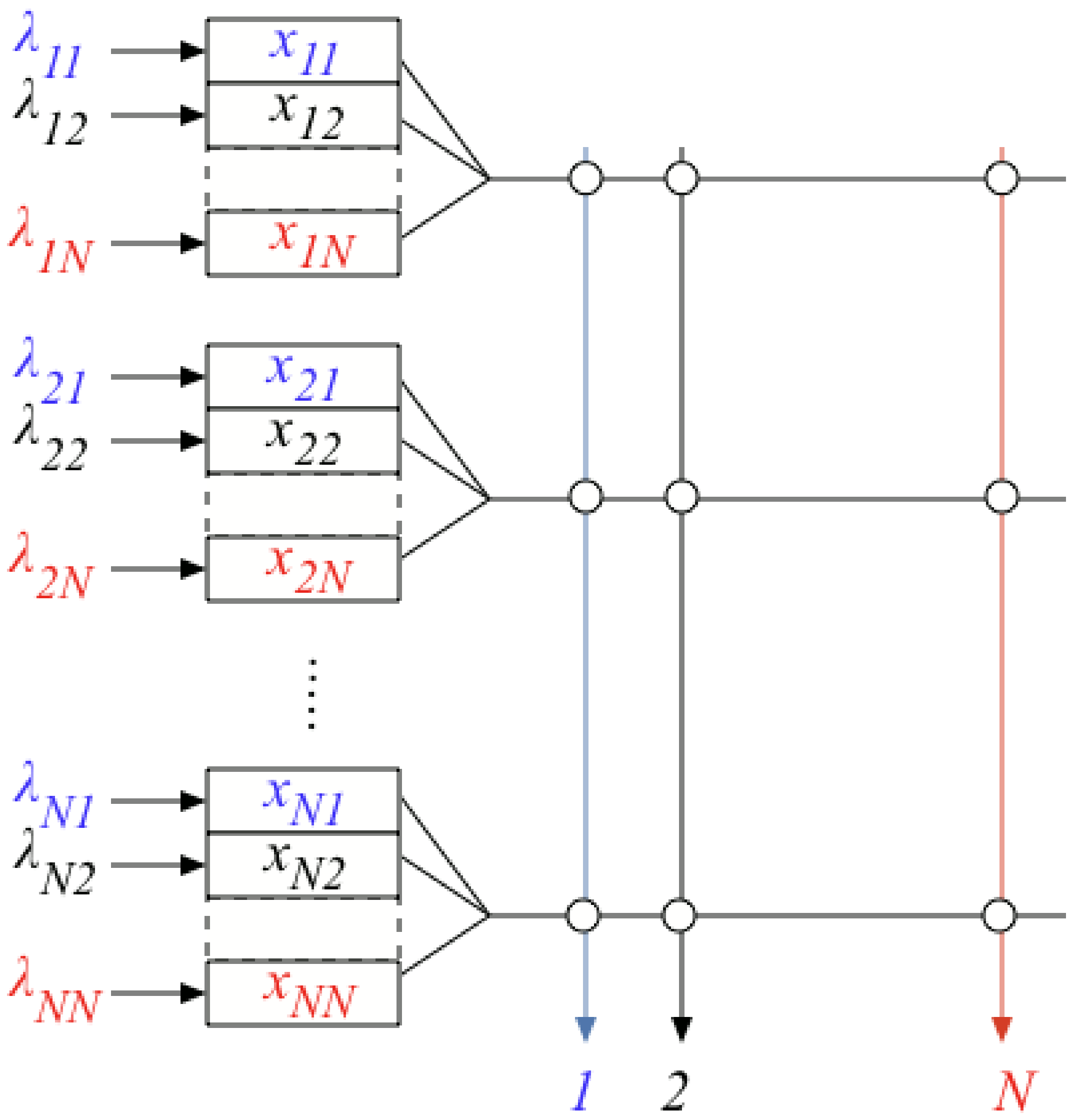
λ_{11}
λ_{12}
λ_{1N}
x_{11}
x_{12}
x_{1N}
λ_{21}
λ_{22}
λ_{2N}
x_{21}
x_{22}
x_{2N}
λ_{N1}
λ_{N2}
λ_{NN}
x_{N1}
x_{N2}
x_{NN}
1
2
N

Switch



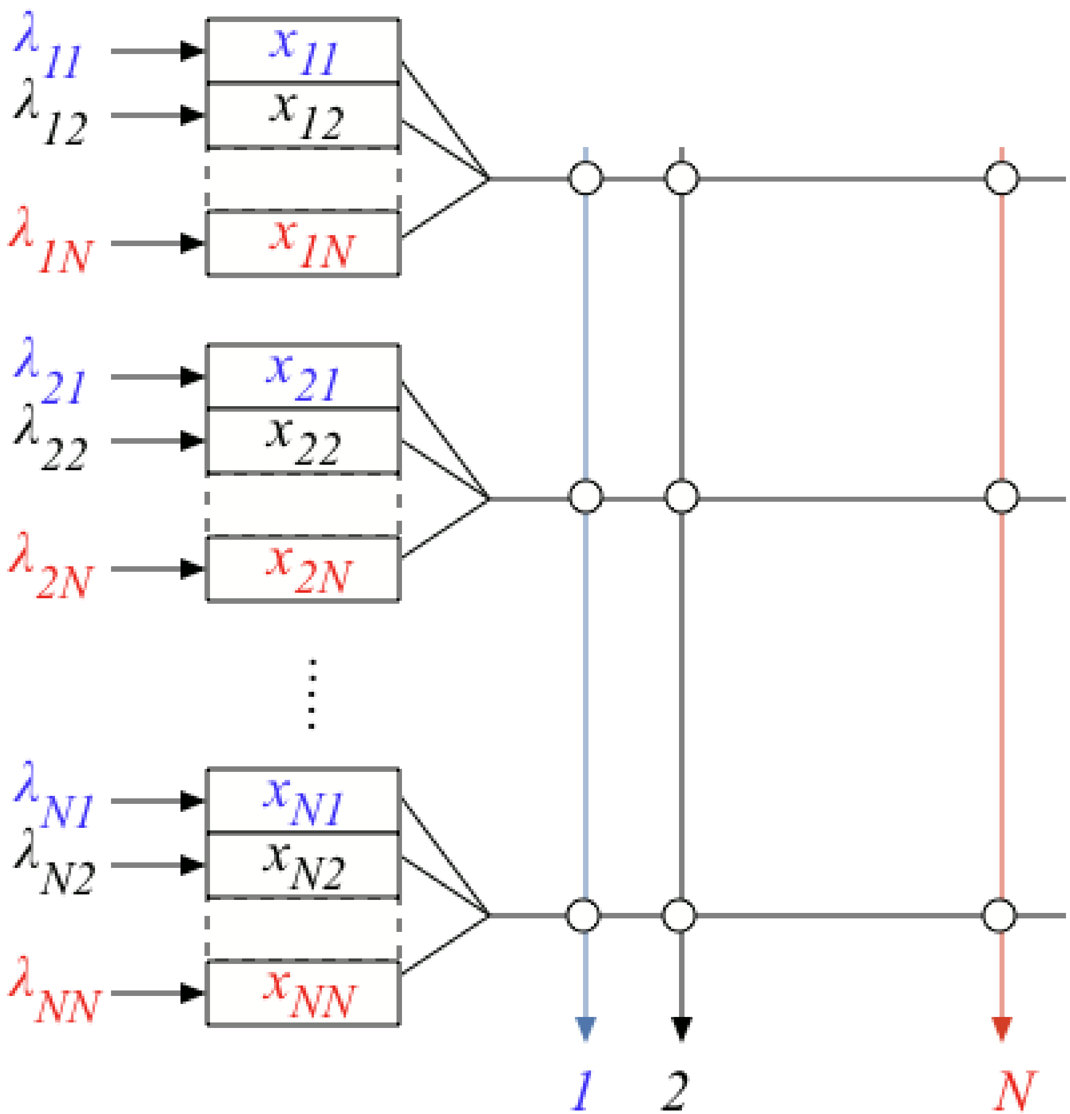
λ_{11}
λ_{12}
λ_{1N}
x_{11}
x_{12}
x_{1N}
λ_{21}
λ_{22}
λ_{2N}
x_{21}
x_{22}
x_{2N}
λ_{N1}
λ_{N2}
λ_{NN}
x_{N1}
x_{N2}
x_{NN}
1
2
N

Switch



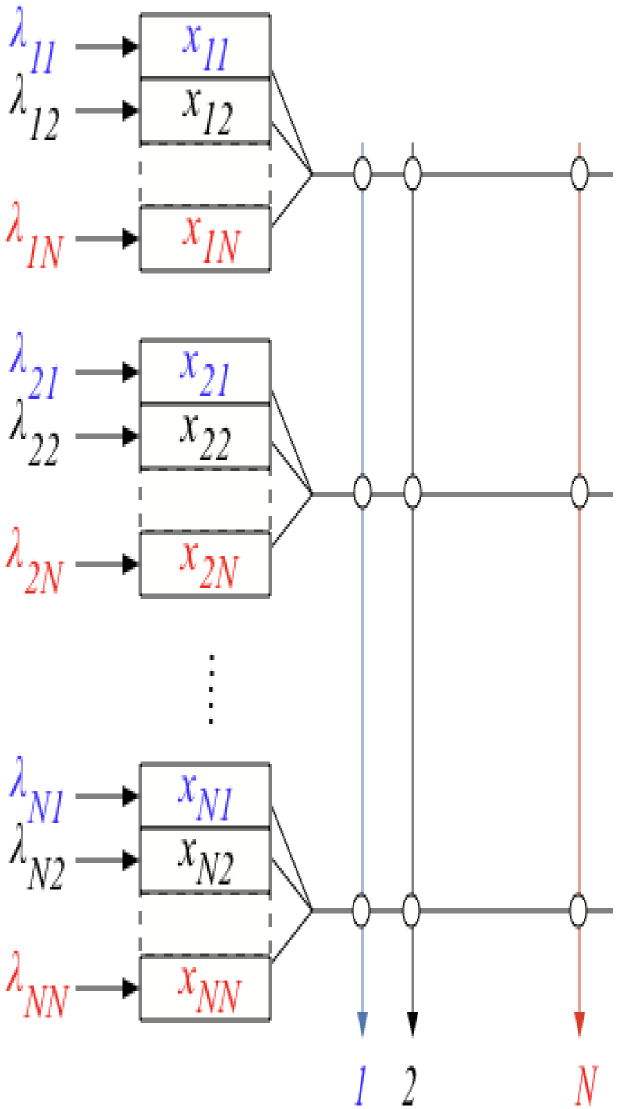
λ_{11}
λ_{12}
λ_{1N}
x_{11}
x_{12}
x_{1N}
λ_{21}
λ_{22}
λ_{2N}
x_{21}
x_{22}
x_{2N}
λ_{N1}
λ_{N2}
λ_{NN}
x_{N1}
x_{N2}
x_{NN}
1
2
N

Switch



λ_{11}
λ_{12}
λ_{1N}
x_{11}
x_{12}
x_{1N}
λ_{21}
λ_{22}
λ_{2N}
x_{21}
x_{22}
x_{2N}
λ_{N1}
λ_{N2}
λ_{NN}
x_{N1}
x_{N2}
x_{NN}
1
2
N

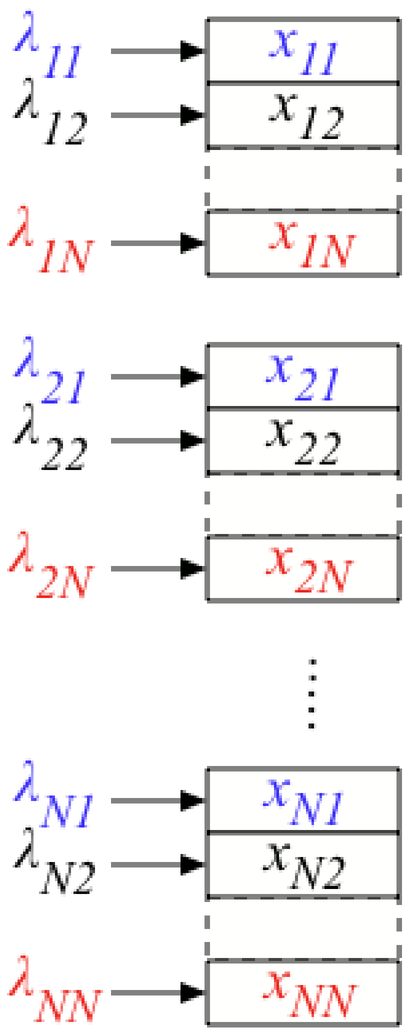
Switch



- Which packet should be sent next to output N?
- Goals?
  - High Throughput
  - Fairness
  - Low Delays
- Classical Answer:
  - Maximum Weighted Matching

    Much Too Complex!
- Simpler Answer:
  - Adaptive Random Requests

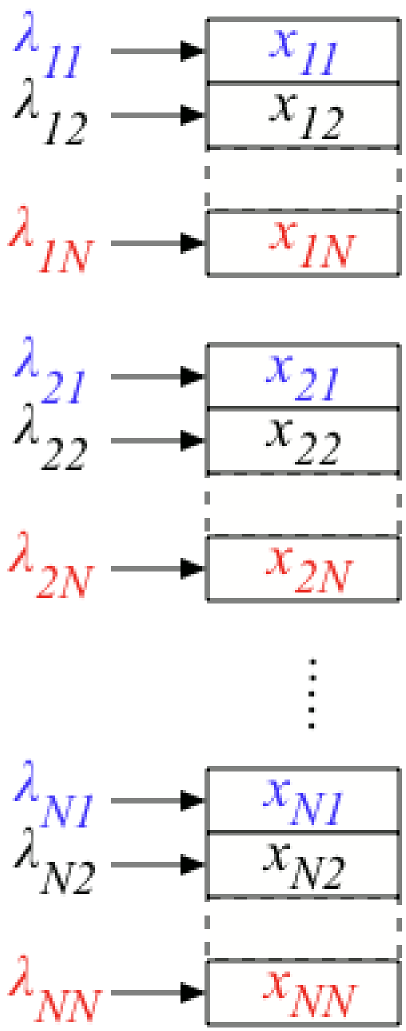
Switch



Adaptive Random Requests:

- Input 1 : Select random delay with mean $\exp\{-\alpha X_{1j}\}$ for every j
- If minimum delay is for j, input 1 checks if output j is busy
  - If not, it sends a packet to j
  - If yes, it repeats
- Same for the other inputs
- Basic Idea: Favor larger backlogs

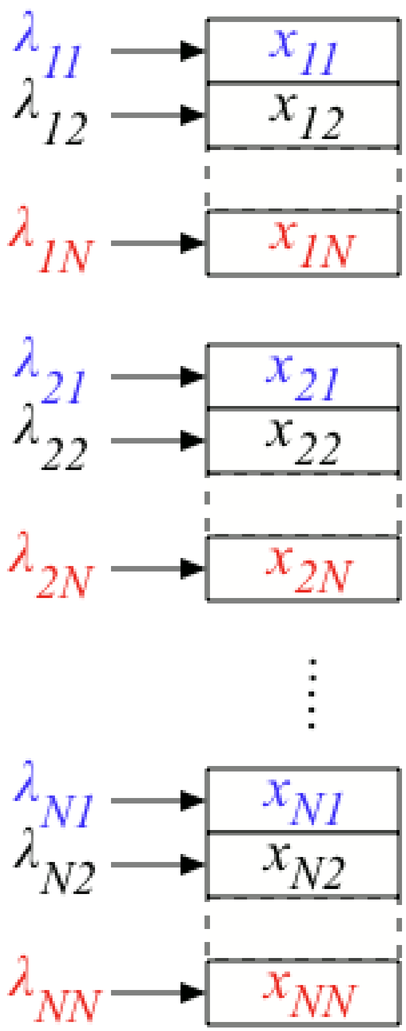
Switch



Adaptive Random Requests:

- Results:
  - ✓ Essentially 100% throughput
  - ✓ Delays can be controlled if we accept a small throughput reduction
  - ✓ Works with variable packet lengths
  - ✓ Fairness? Next slide.

Switch



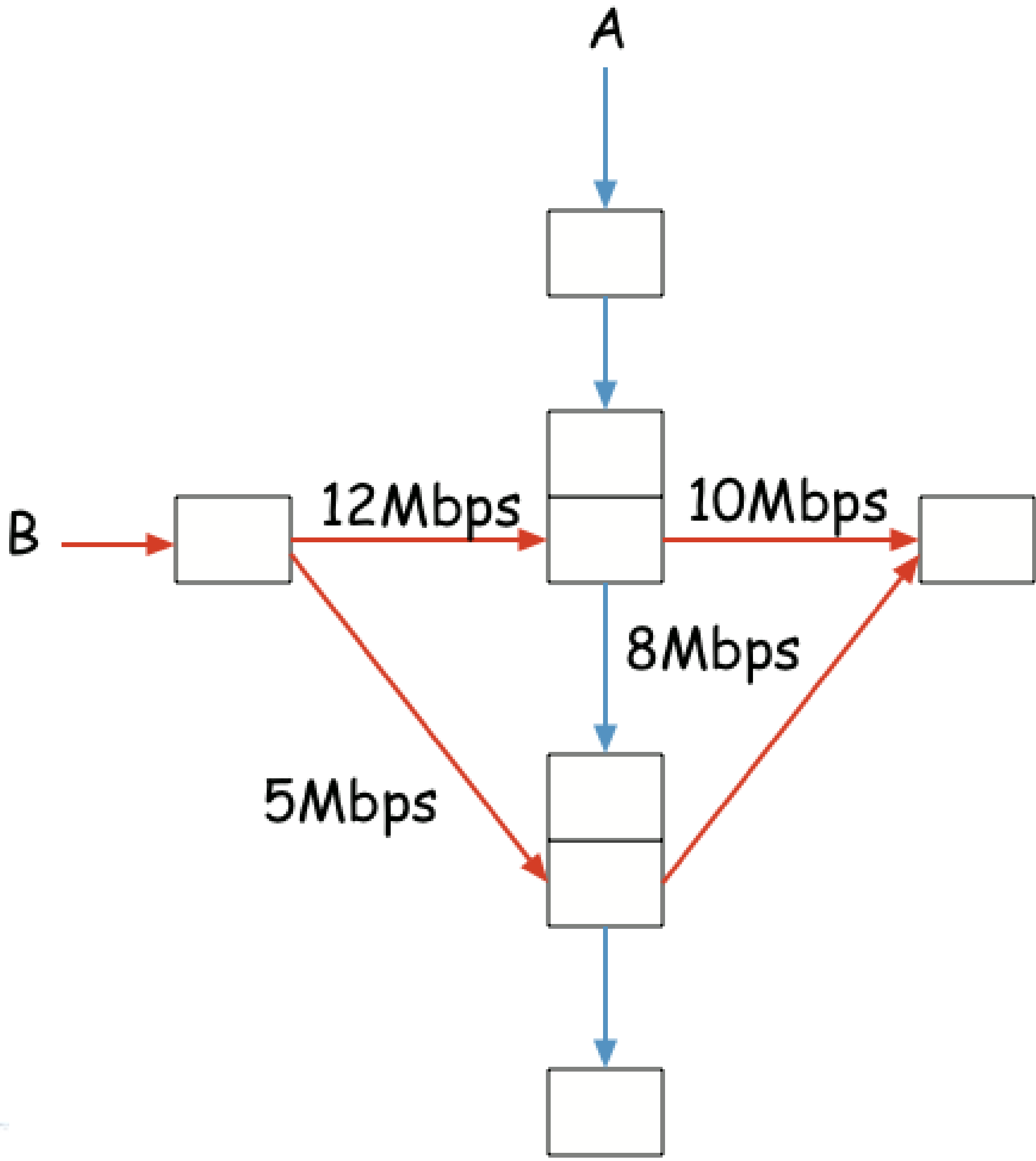
Adaptive Random Requests:

- Fairness:
  - Requires congestion control
  - Input ij reduces λ_{ij} if x_{ij} increases
  - Choose λ_{ij} to maximize

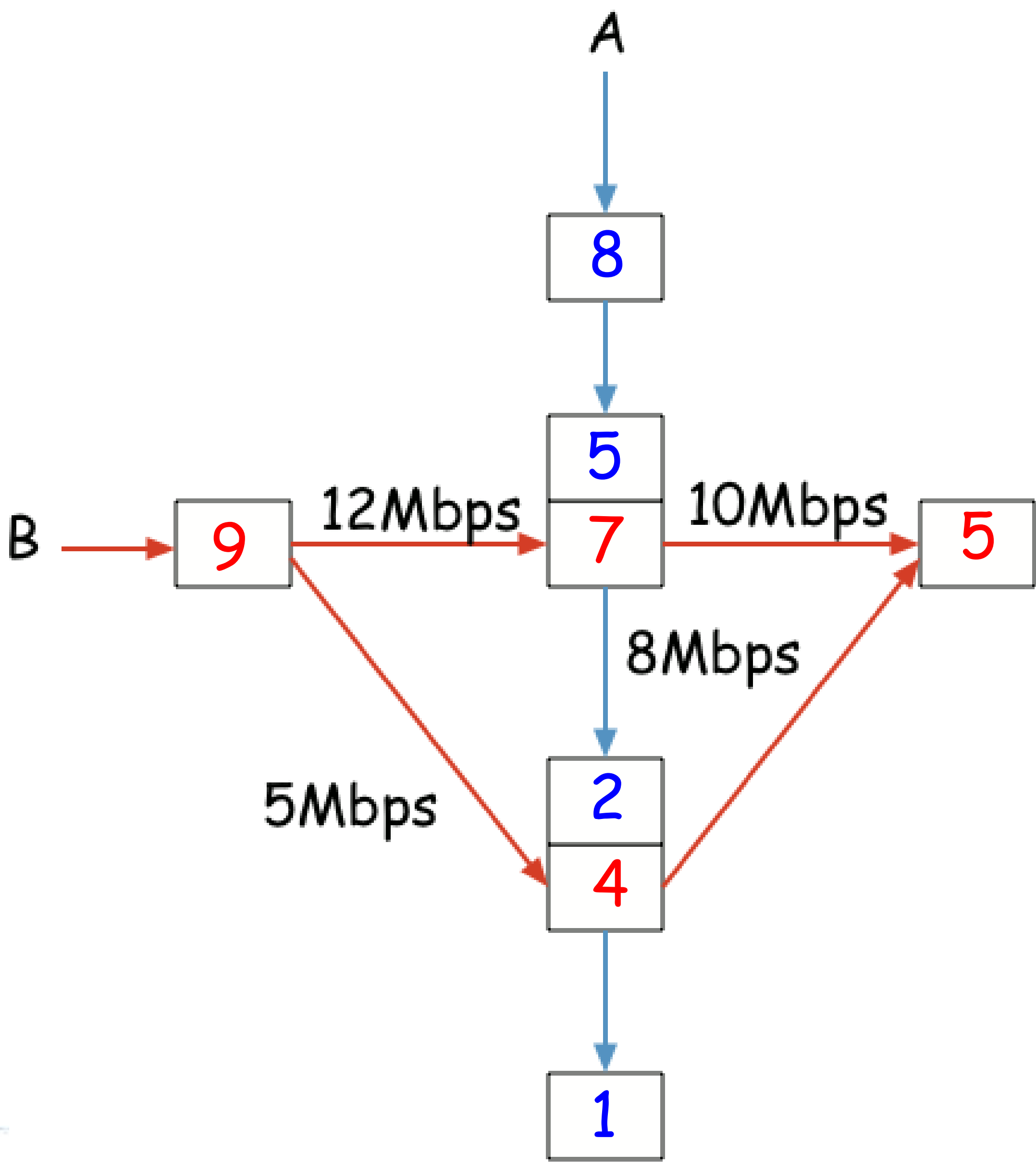
$$u_{ij}(\lambda_{ij}) - \beta x_{ij}\lambda_{ij}$$

- Result:
  - Essentially maximizes $\Sigma u_{ij}(\lambda_{ij})$

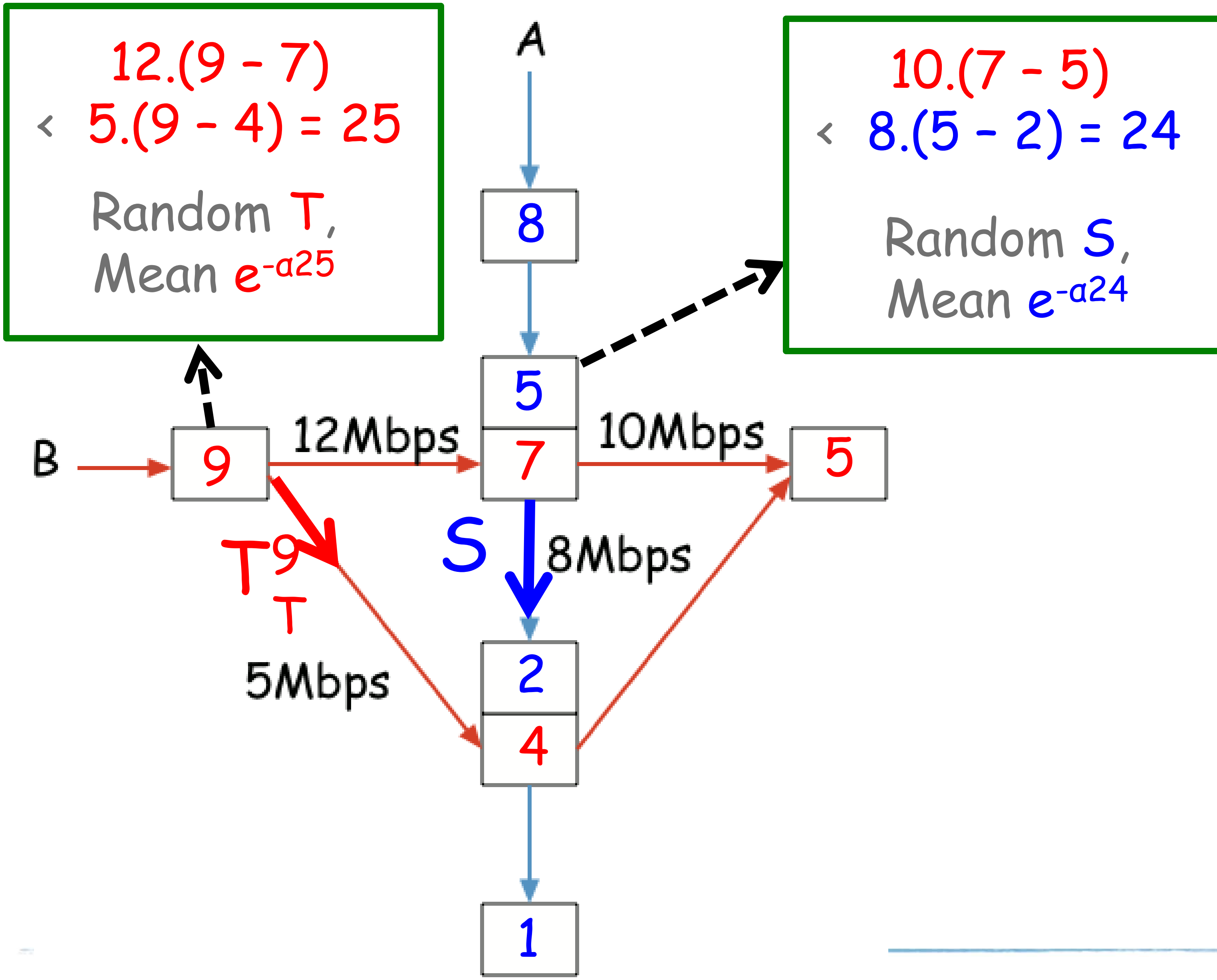
Ad Hoc Network



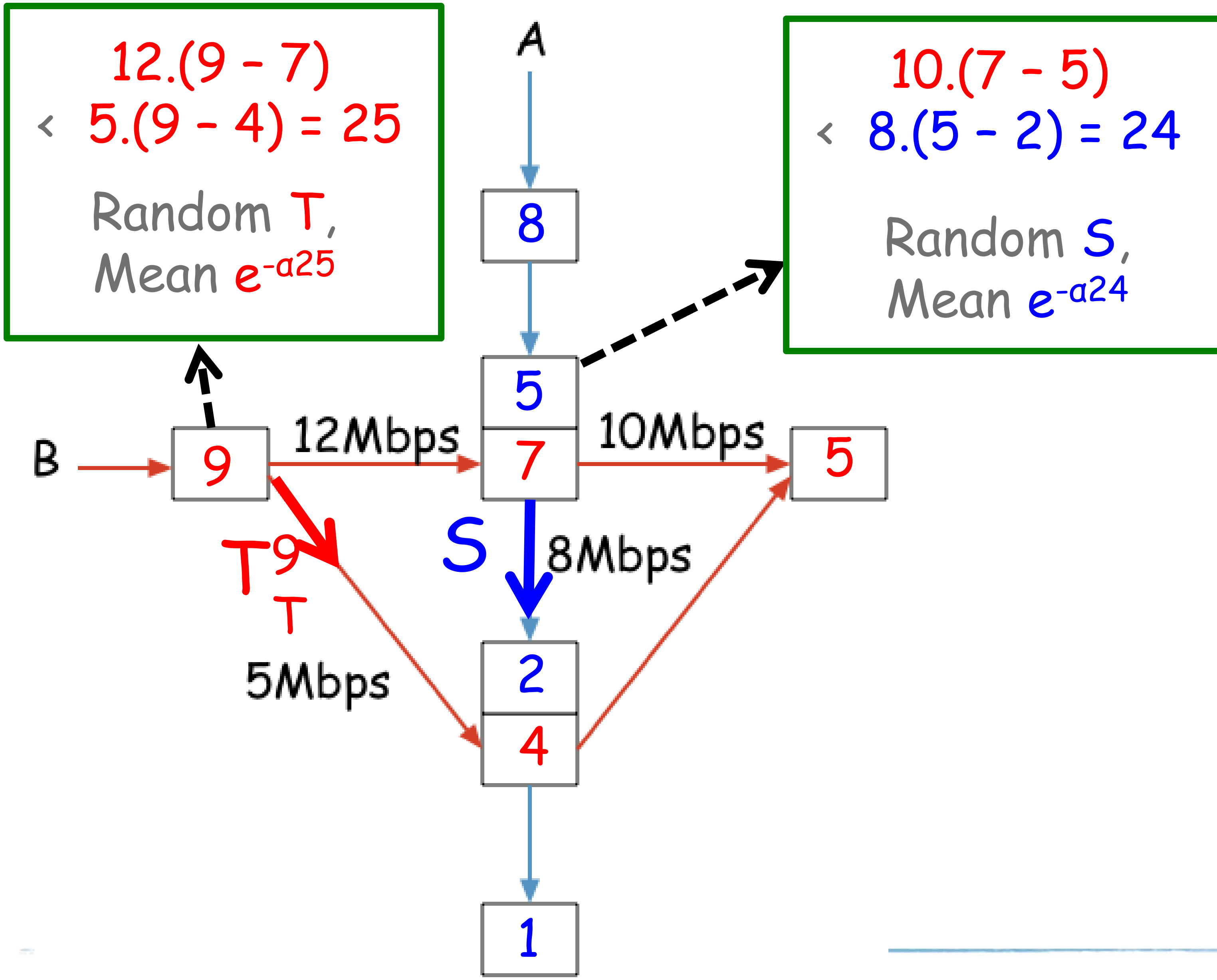
Ad Hoc Network



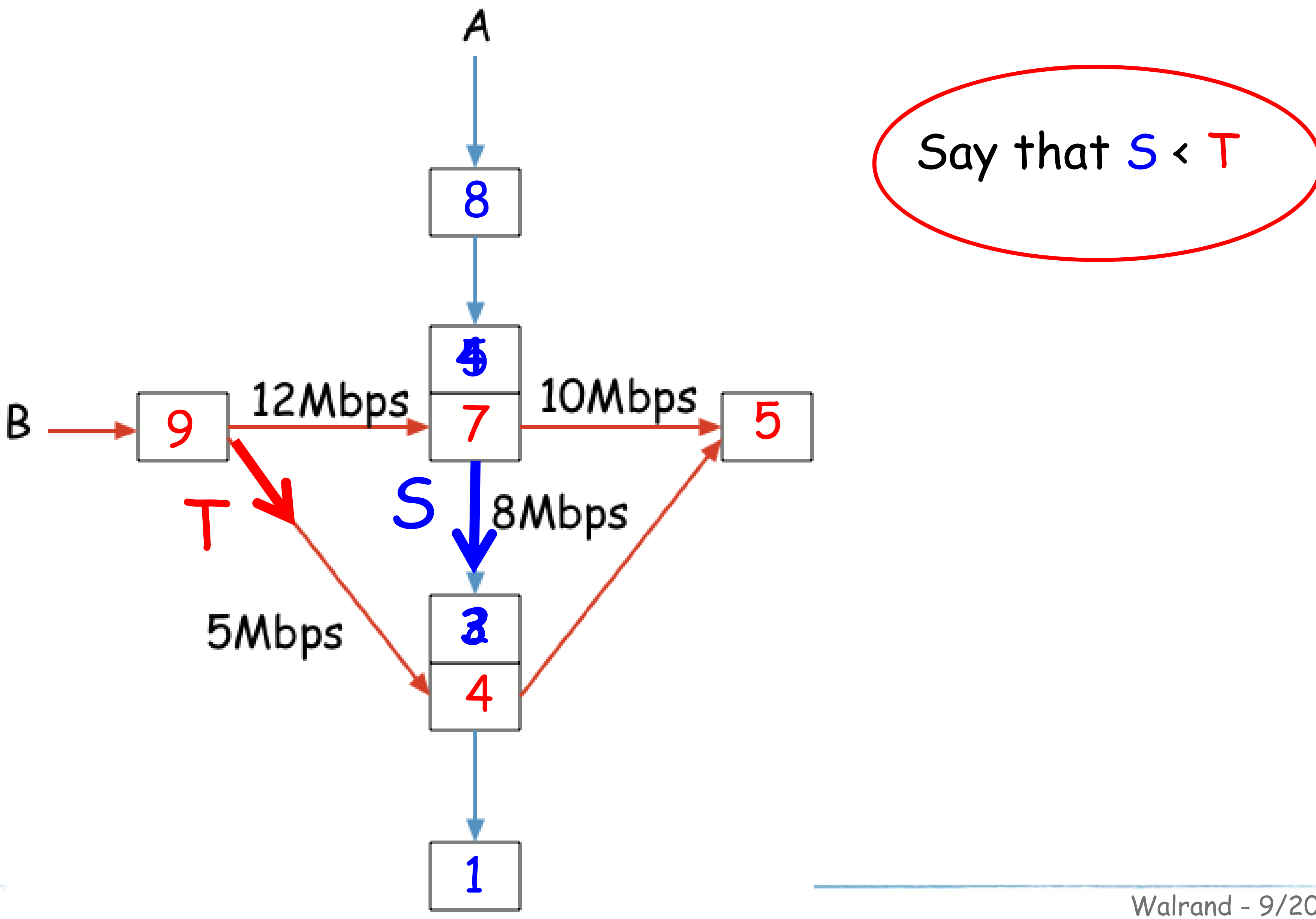
Ad Hoc Network



Ad Hoc Network

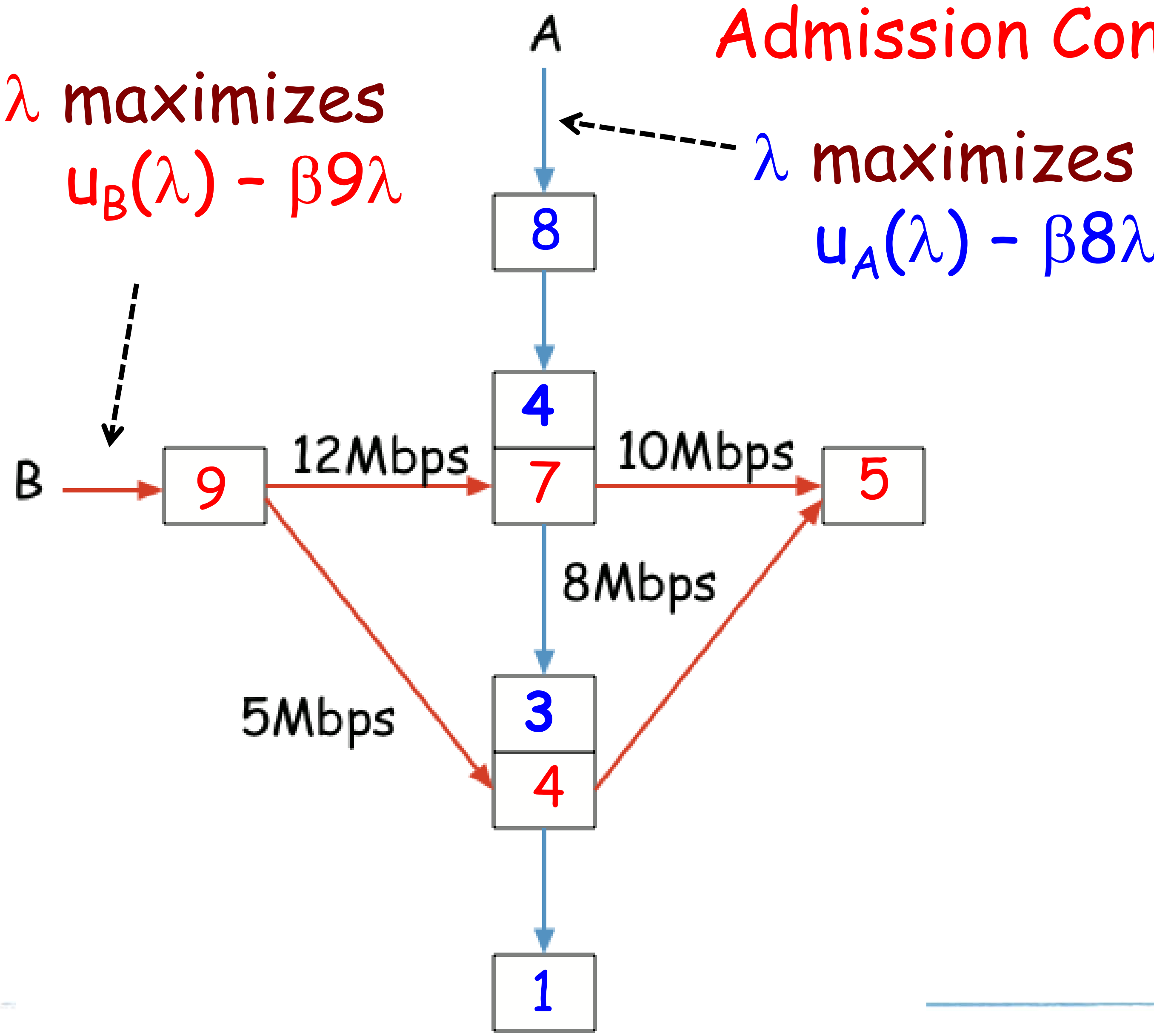


Ad Hoc Network

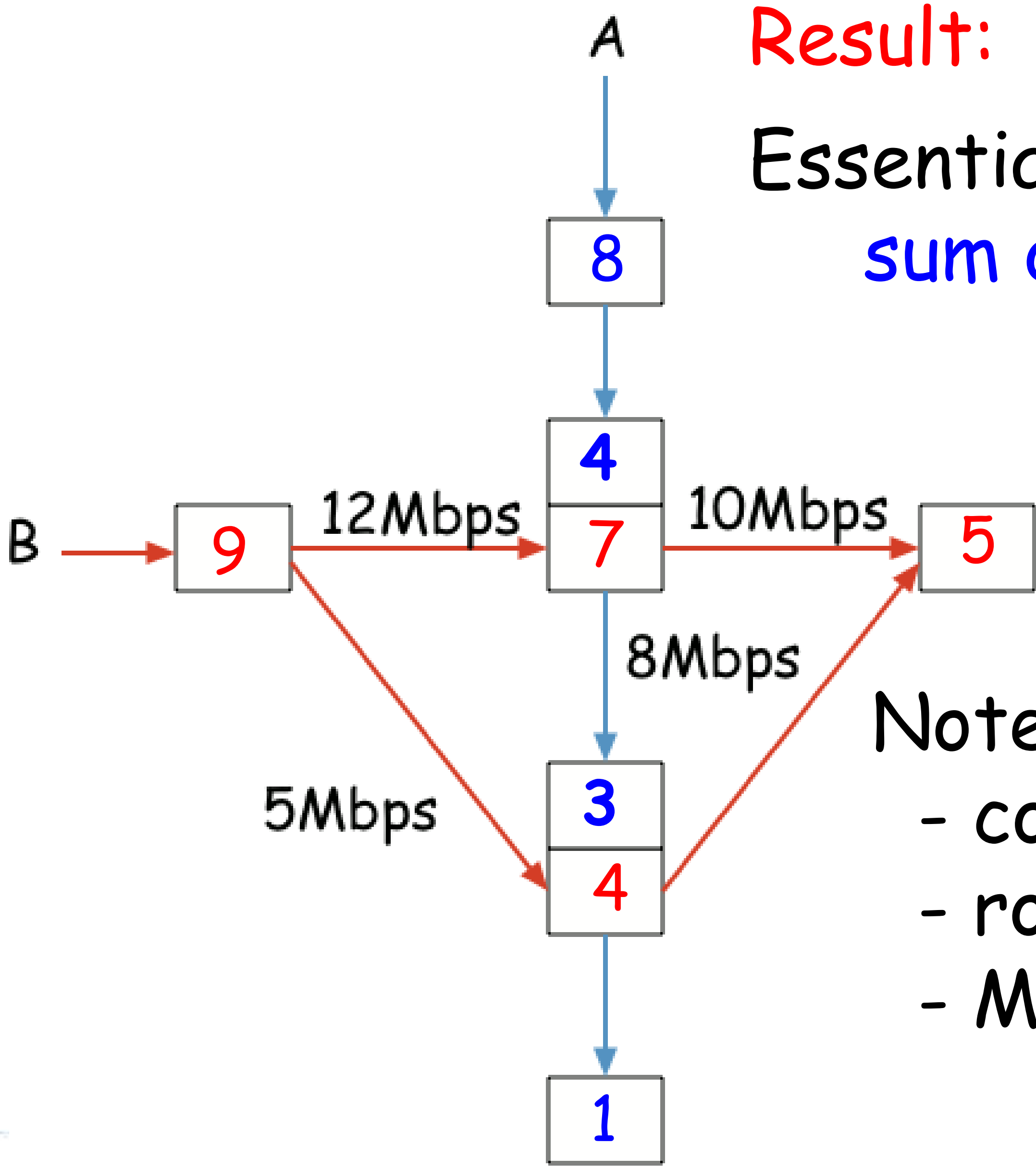


Ad Hoc Network

Admission Control:



Ad Hoc Network

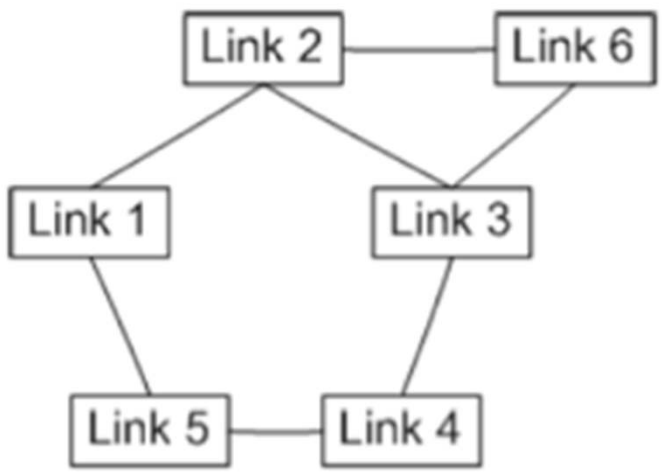


Result:

Essentially maximizes the sum of flow utilities

Note: Integrates
- congestion control
- routing
- MAC scheduling

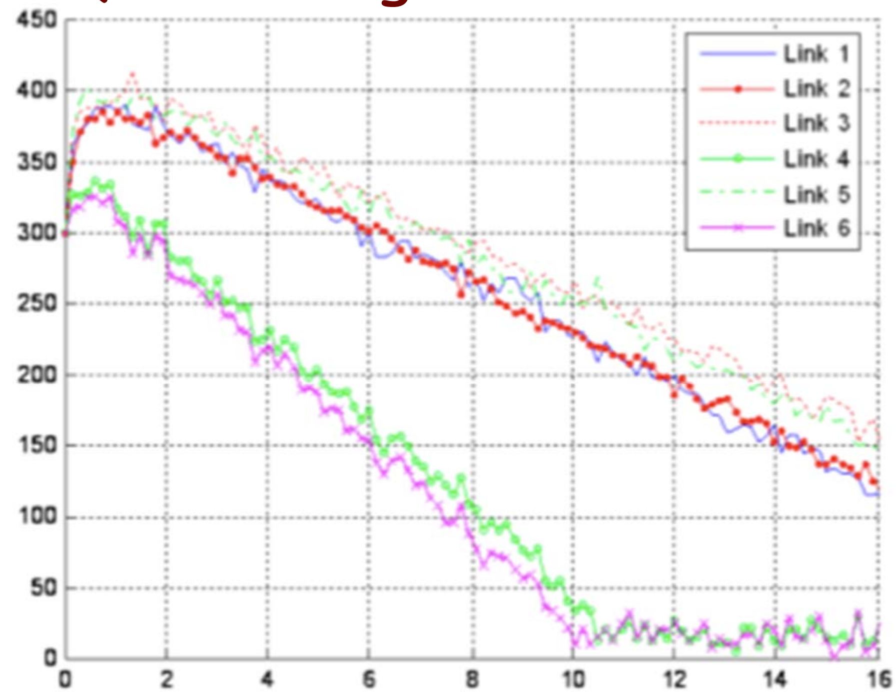
Ad Hoc Network



λ=0.98*
(0.5,0.2,0.5,0.3,0.5,0.3) †

Network

Queue Lengths

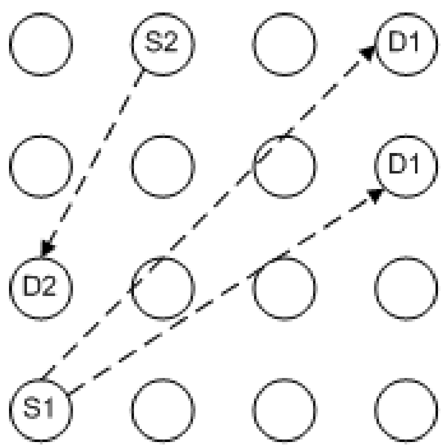


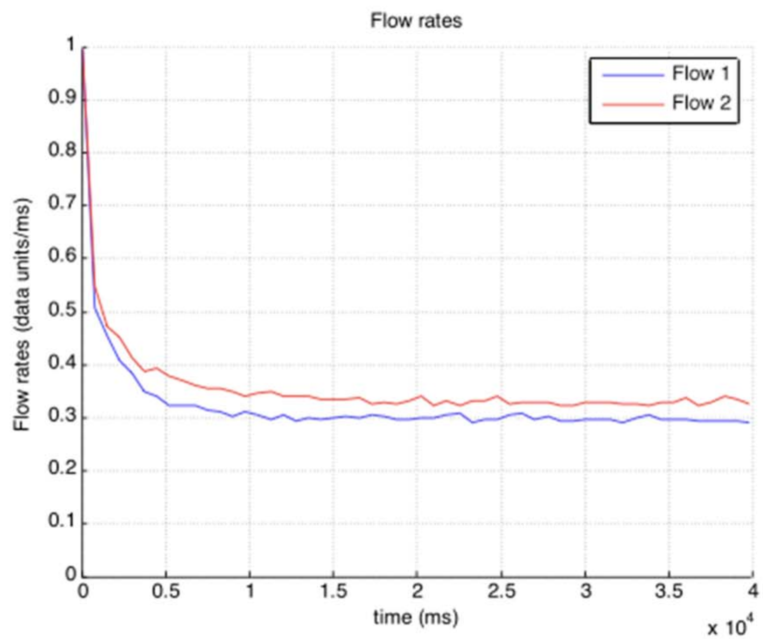
Time

† λ = 0.98*(convex combination of maximal independent sets)

0.2*{1, 3} + 0.3*{1, 4, 6} + 0.3*{3, 5} + 0*{2, 4} + 0.2*{2, 5}

Ad Hoc Network





Multipath routing allowed

Unicast S2 -> D2
Anycast S1 to any D1

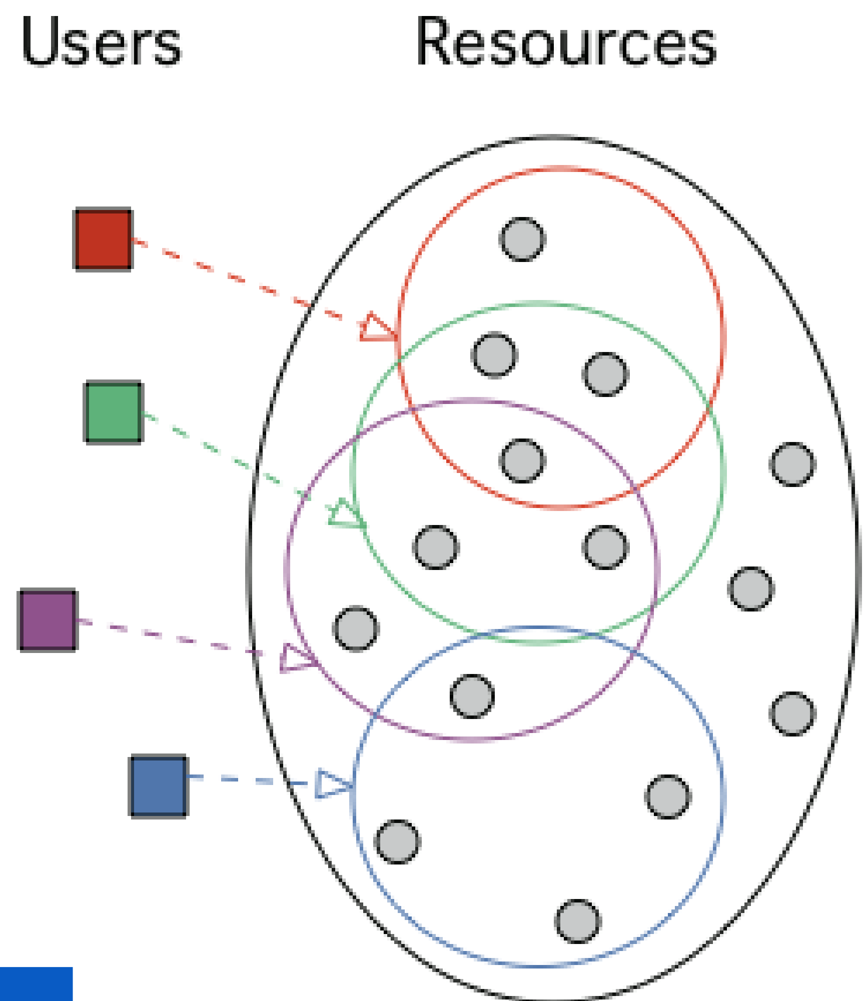
Resource Allocation

- Many users compete for resources
  - CPU, Memory in Cloud
  - Energy
  - Wireless Channels

- For scalability, the protocols must be distributed

- The protocols should be efficient and strategy-proof

- Optimal allocation is NP-hard and requires full knowledge



Resource Allocation

- Replace

$$MAX\ \Sigma_i u_i(x_i)$$

by

$$MAX\ \Sigma_i u_i(x_i) + \beta H(p)$$

H = entropy of allocation

- Magic:

From NP-hard, the problem becomes

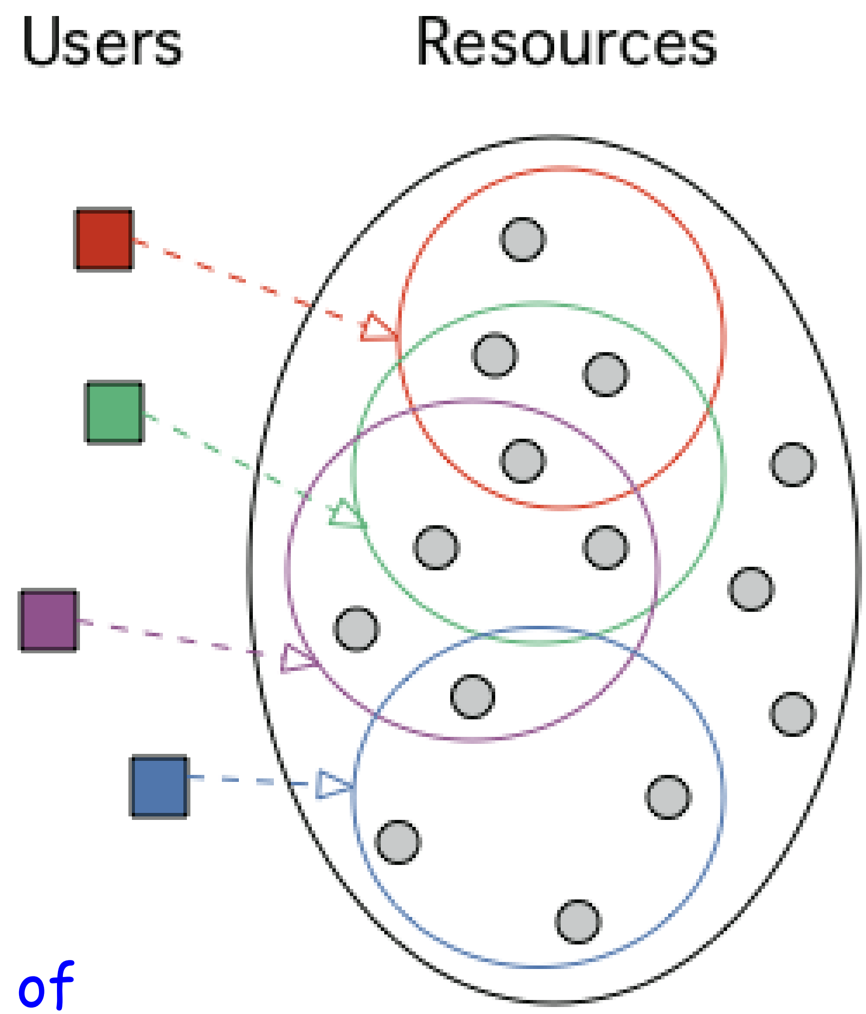
  - Distributed
  - Easy

The solution is

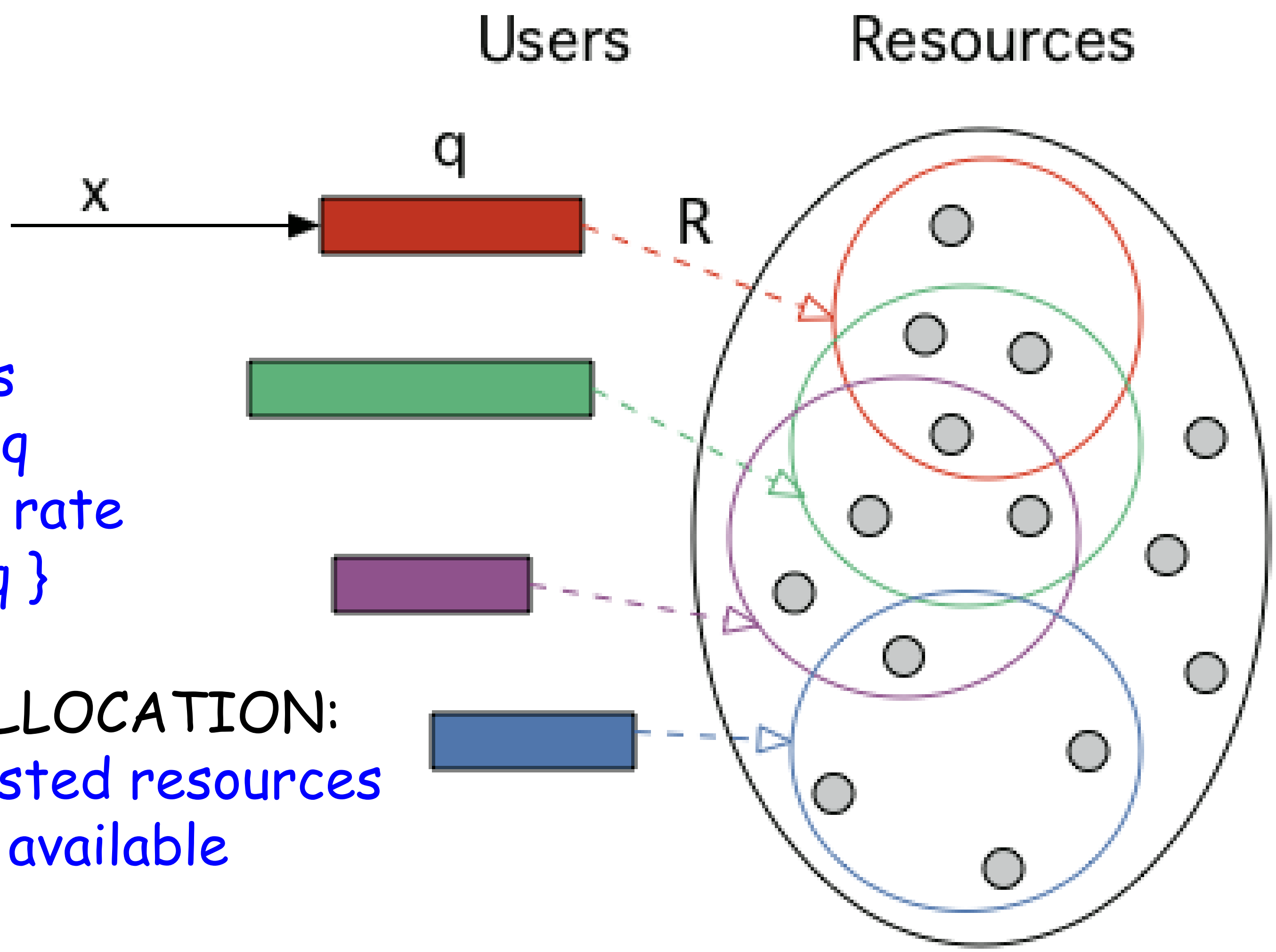
$$O(T/\beta)\text{-optimal}$$

T = mixing time ....

Bounds on T based on topology of resource conflicts.



Resource Allocation



USER i:

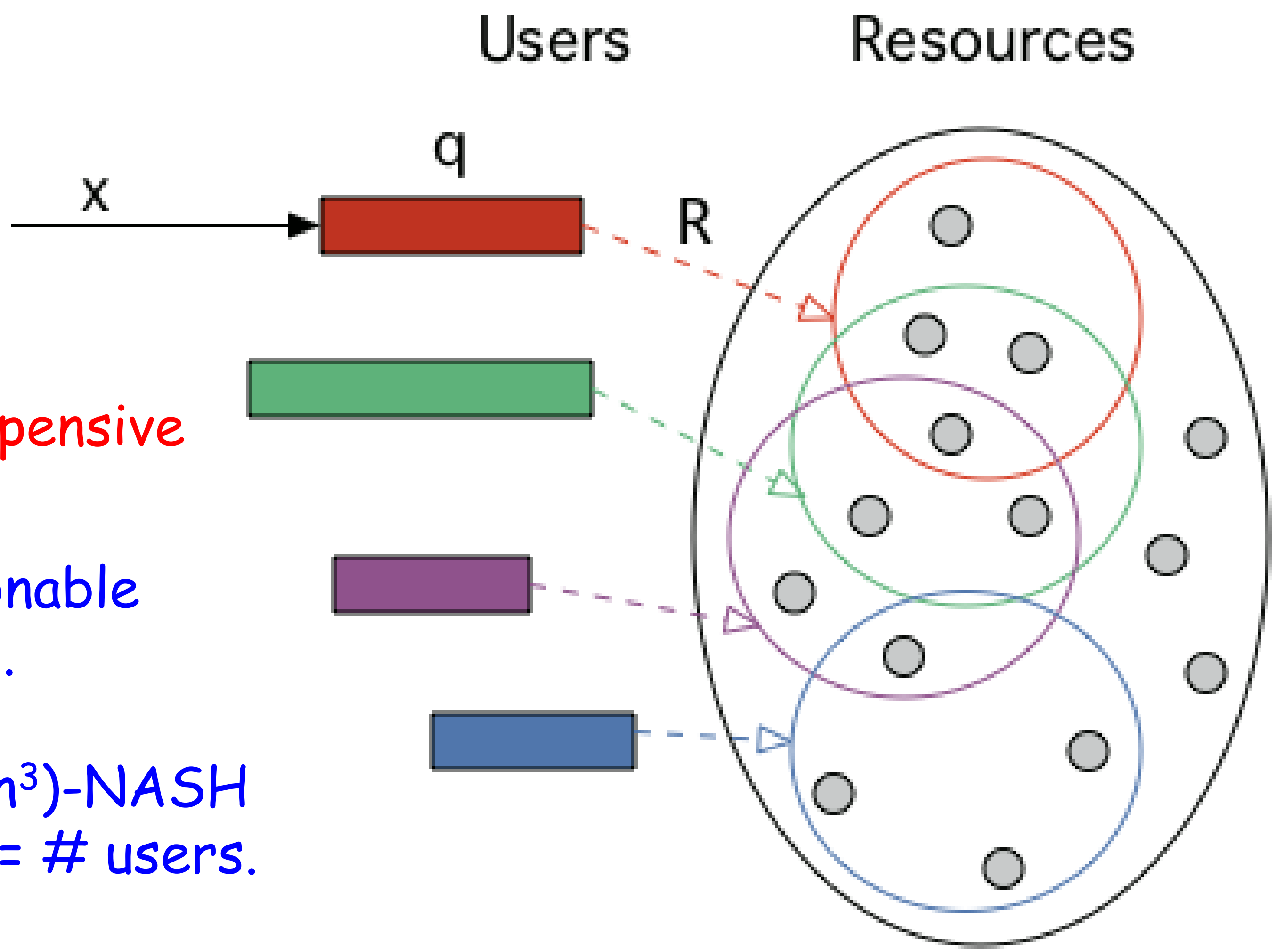
- x maximizes
  $u_i(x) - \beta x q$
- R = request rate
  $\approx \exp\{ \alpha q \}$

RESOURCE ALLOCATION:

- Grant requested resources if they are all available

Resource Allocation

What about strategic users?



USER i:

- Charge $\beta x q$

Intuition:

- Greed is expensive

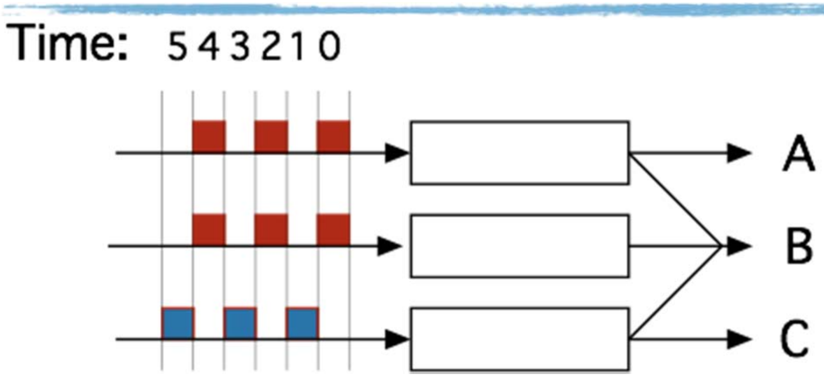
RESULT:

- Under reasonable assumptions ….

Scheme is $(1/n^3)$-NASH equilibrium; n = # users.

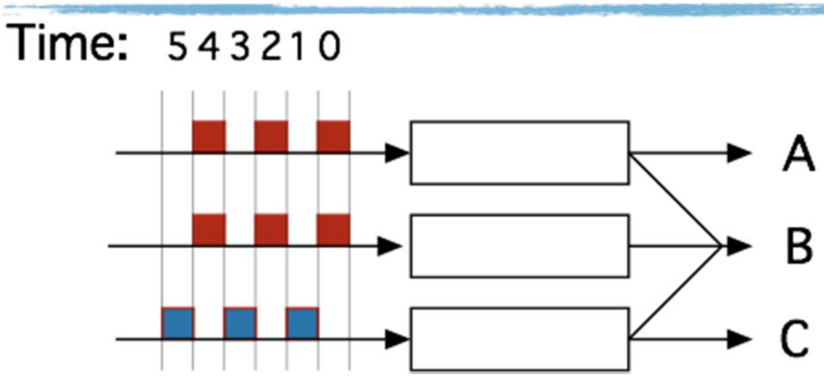
(Price is almost VCG.)

Processing Networks



Task: 1 from queue 1;
Task B: 1 from all queues;
Task C: 1 from queue 3

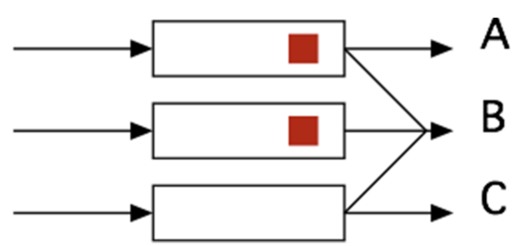
Processing Networks



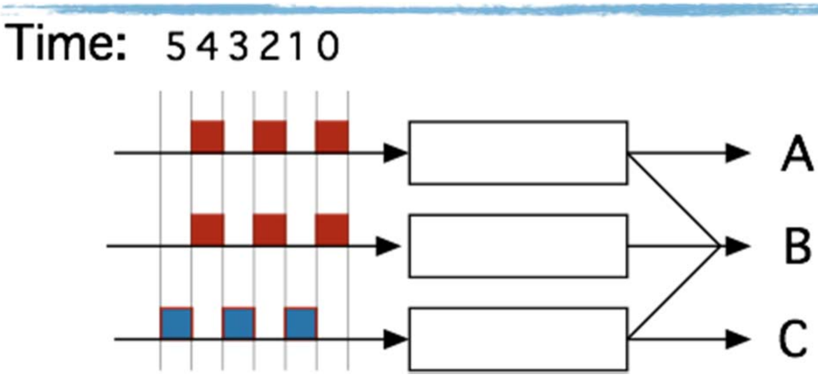
Task: 1 from queue 1;
Task B: 1 from all queues;
Task C: 1 from queue 3

MWM

T = 0



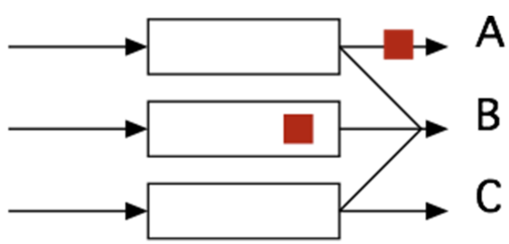
Processing Networks



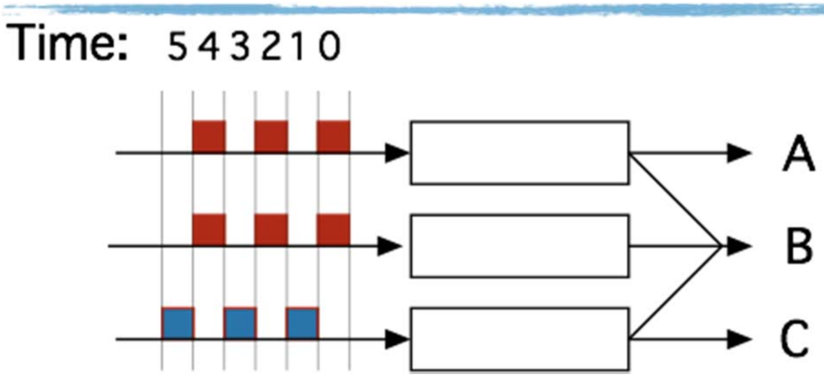
Task: 1 from queue 1;
Task B: 1 from all queues;
Task C: 1 from queue 3

MWM

T = 1-



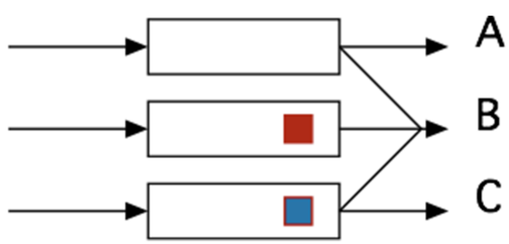
Processing Networks



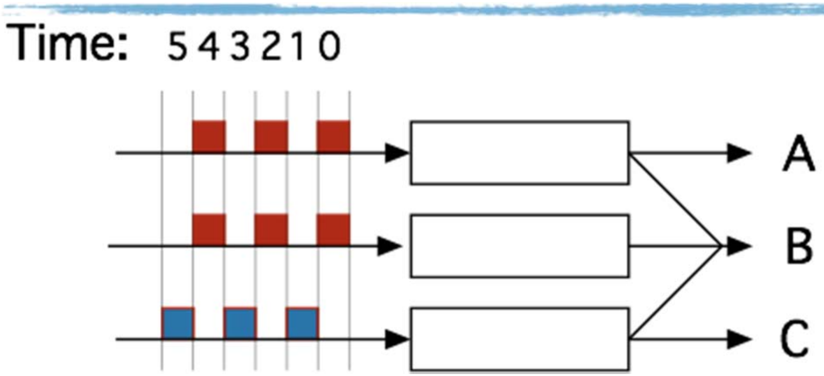
Task: 1 from queue 1;
Task B: 1 from all queues;
Task C: 1 from queue 3

MWM

T = 1



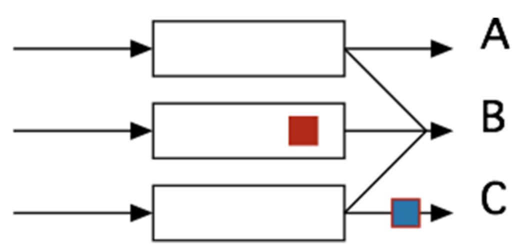
Processing Networks



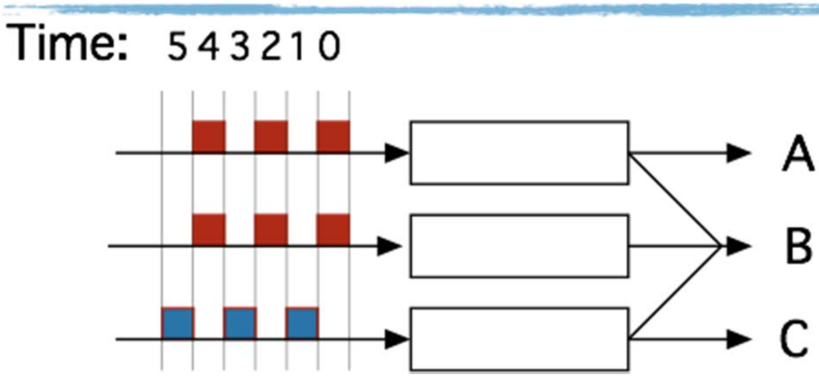
Task: 1 from queue 1;
Task B: 1 from all queues;
Task C: 1 from queue 3

MWM

T = 2-



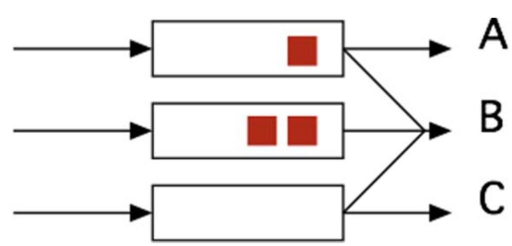
Processing Networks



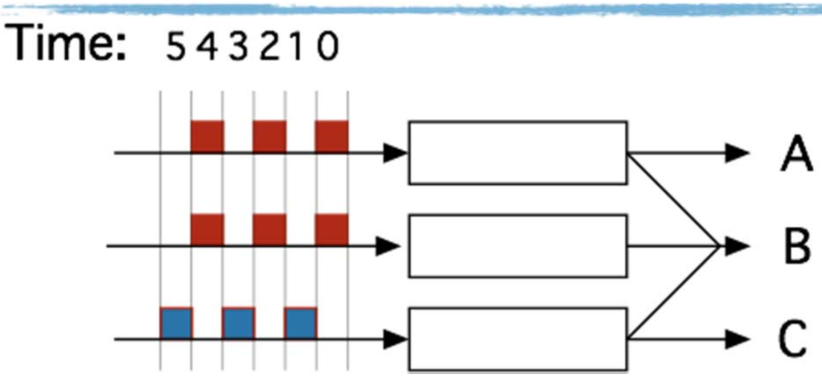
Task: 1 from queue 1;
Task B: 1 from all queues;
Task C: 1 from queue 3

MWM

T = 2



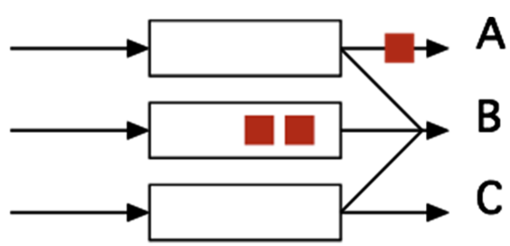
Processing Networks



Task: 1 from queue 1;
Task B: 1 from all queues;
Task C: 1 from queue 3

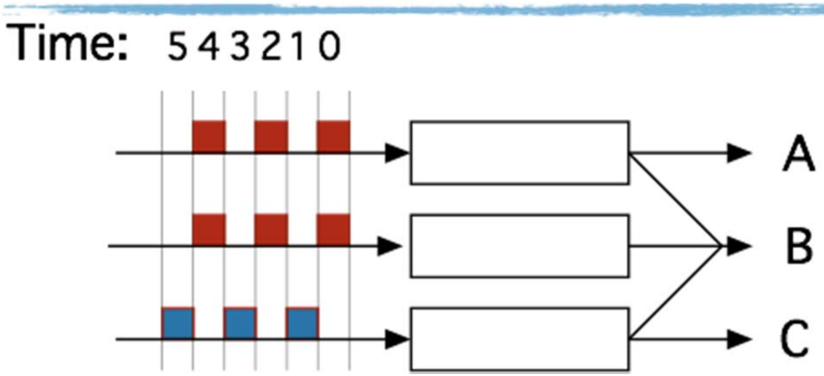
MWM

T = 3-



Maximum Weighted Matching is not stable.

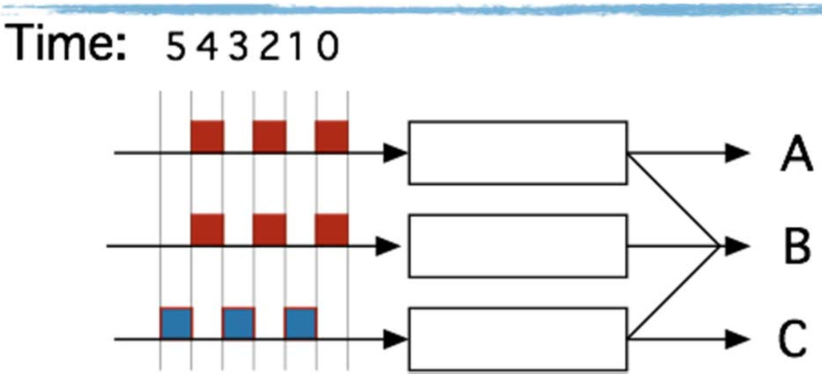
Processing Networks



Task: 1 from queue 1;
Task B: 1 from all queues;
Task C: 1 from queue 3

DWM: Use MWM based on Virtual Queues

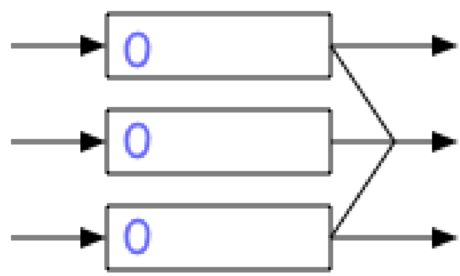
Processing Networks



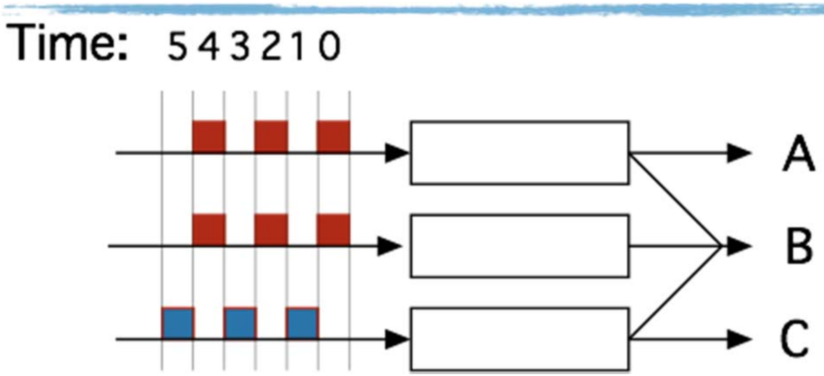
Task: 1 from queue 1;
Task B: 1 from all queues;
Task C: 1 from queue 3

DWM: Use MWM based on Virtual Queues

T = 0-



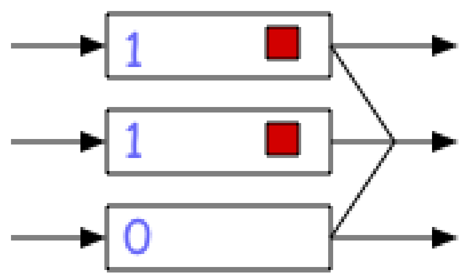
Processing Networks



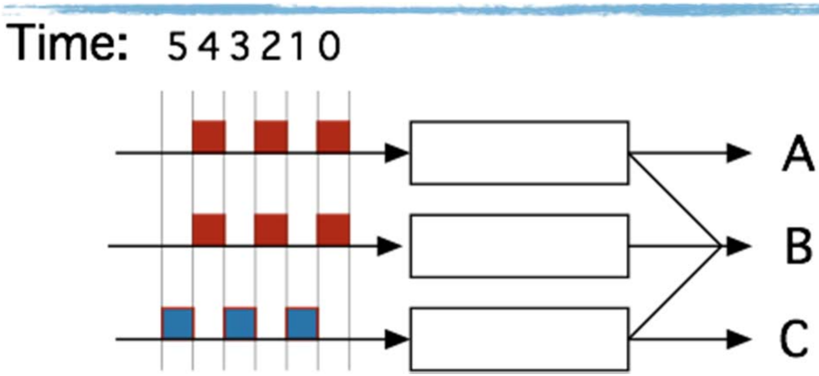
Task: 1 from queue 1;
Task B: 1 from all queues;
Task C: 1 from queue 3

DWM: Use MWM based on Virtual Queues

T = 0



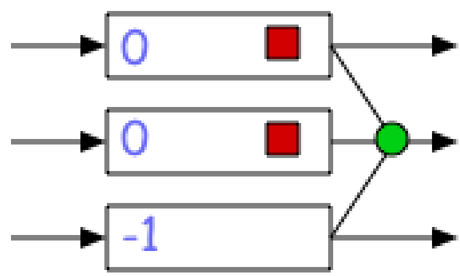
Processing Networks



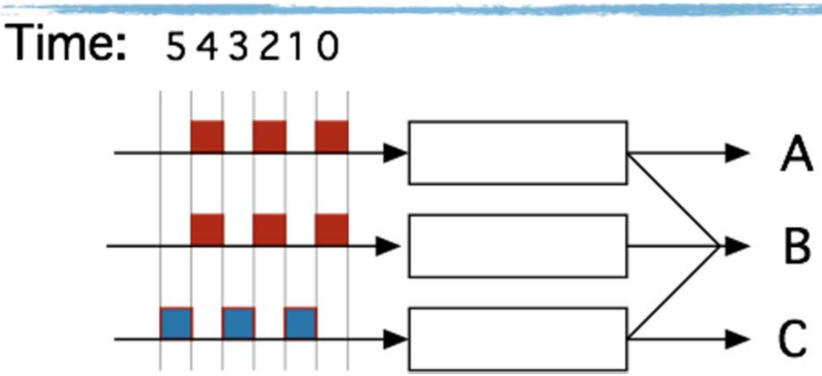
Task: 1 from queue 1;
Task B: 1 from all queues;
Task C: 1 from queue 3

DWM: Use MWM based on Virtual Queues

T = 1-



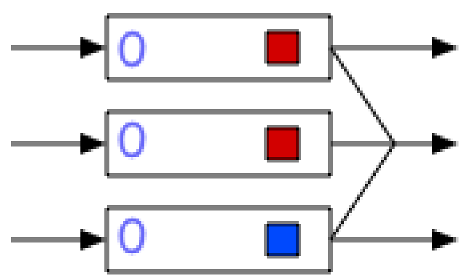
Processing Networks



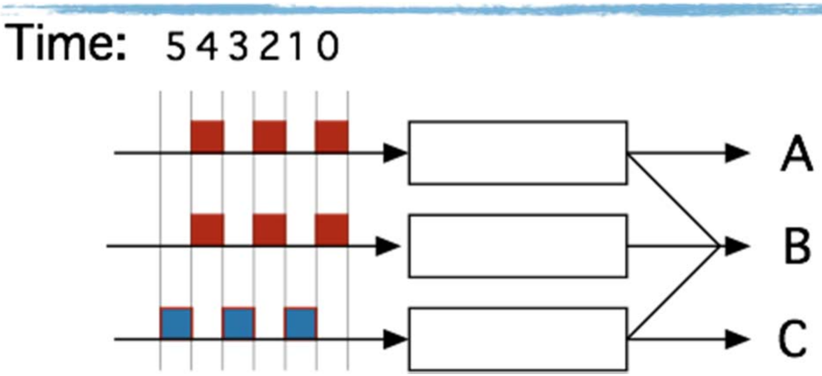
Task: 1 from queue 1;
Task B: 1 from all queues;
Task C: 1 from queue 3

DWM: Use MWM based on Virtual Queues

T = 1



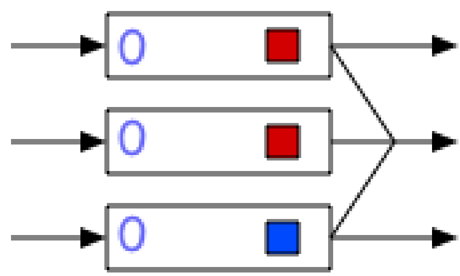
Processing Networks



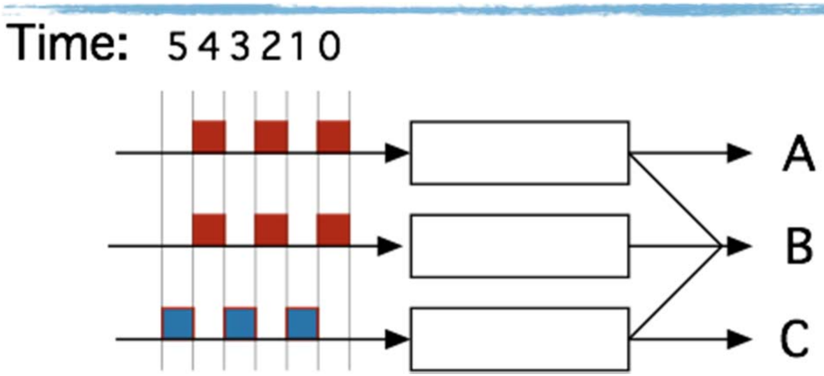
Task: 1 from queue 1;
Task B: 1 from all queues;
Task C: 1 from queue 3

DWM: Use MWM based on Virtual Queues

T = 2-



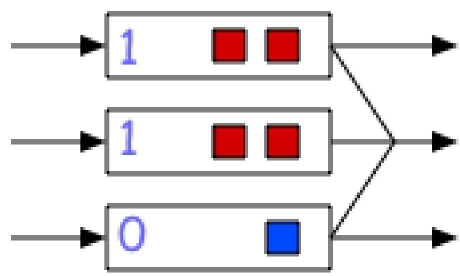
Processing Networks



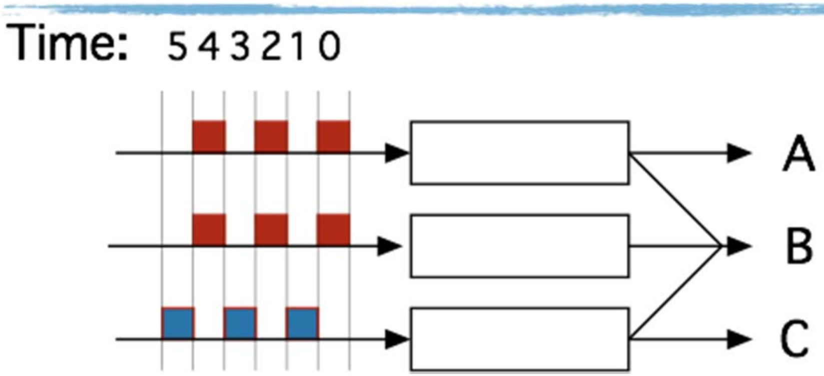
Task: 1 from queue 1;
Task B: 1 from all queues;
Task C: 1 from queue 3

DWM: Use MWM based on Virtual Queues

T = 2



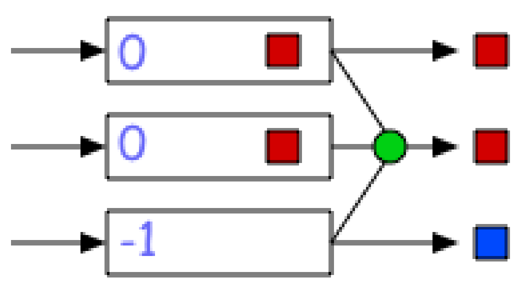
Processing Networks



Task: 1 from queue 1;
Task B: 1 from all queues;
Task C: 1 from queue 3

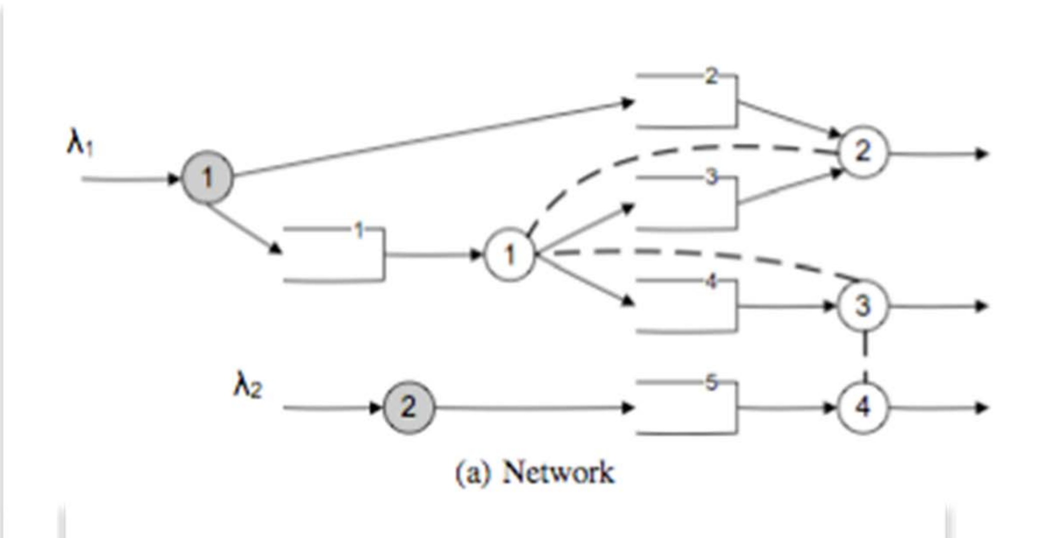
DWM: Use MWM based on Virtual Queues

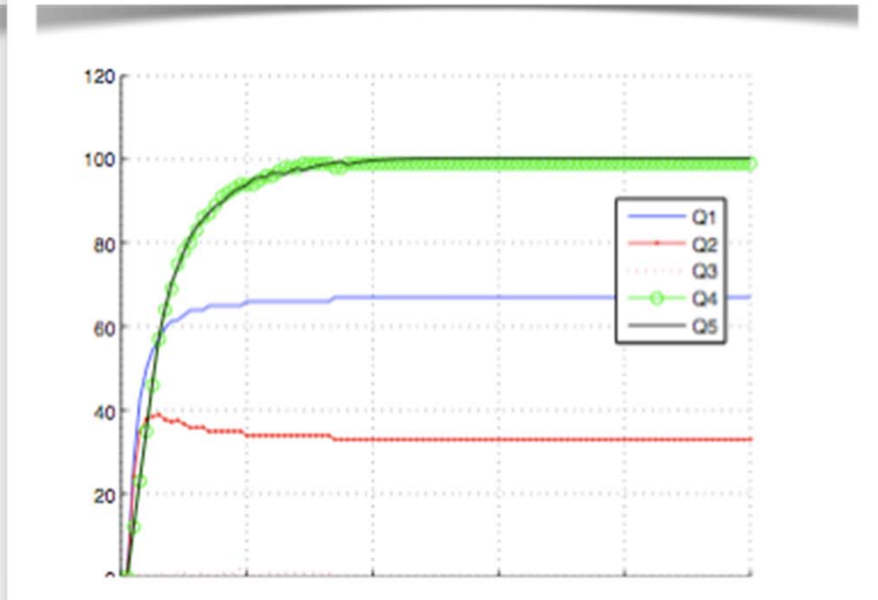
T = 3-



Deficit Maximum Weighted Matching is stable.
[Proof: Lyapunov argument.]

Processing Networks





Parts arrive
at 1 & 2 with rate λ_1
and at 5 with rate λ_2

Task 2 consumes one part from 2 and one from 3; ...

Tasks 1-2, 1-3, 3-4 conflict

Algorithm stabilizes the queues and achieves the max. utility

Mathematical Ideas

For distributed allocations there are three ideas:

* Random access protocols maximize the entropy subject to average allocation rates
* The dual gradient algorithm to solve this problem calculates the optimal access rates
* The implementable algorithm is a stochastic approximation version of the dual gradient algorithm

For processing networks, there is one idea:

* The virtual queues are stable, by Lyapunov.

These four ideas are in Libin Jiang's thesis. See monograph.

Maximum Entropy

Consider:

$$\text{Maximize } H(\pi) := -\sum_{S} \pi(S) \log \pi(S)$$

$$\text{Subject to } s_j(\pi) := \sum_{\{S|j\in S\}} \pi(S) \geq \lambda_j, \forall j \text{ and } \sum_{S} \pi(S) = 1$$

Maximum Entropy

Consider:

$$\text{Maximize } H(\pi) := -\sum_{S} \pi(S) \log \pi(S)$$

$$\text{Subject to } s_j(\pi) := \sum_{\{S \mid j \in S\}} \pi(S) \geq \lambda_j, \forall j \text{ and } \sum_{S} \pi(S) = 1$$

Lagrangian:

$$L(\pi, r) := -\sum_{S} \pi(S) \log \pi(S) - \sum_{j} r_j [\lambda_j - \sum_{\{S \mid j \in S\}} \pi(S)] + r_0 [\sum_{S} \pi(S) - 1]$$

Maximum Entropy

Consider:

$$\text{Maximize } H(\pi) := -\sum_{S} \pi(S) \log \pi(S)$$

$$\text{Subject to } s_j(\pi) := \sum_{\{S|j \in S\}} \pi(S) \geq \lambda_j, \forall j \text{ and } \sum_{S} \pi(S) = 1$$

Lagrangian:

$$L(\pi, r) := -\sum_{S} \pi(S) \log \pi(S) - \sum_{j} r_j [\lambda_j - \sum_{\{S|j \in S\}} \pi(S)] + r_0 [\sum_{S} \pi(S) - 1]$$

$$\frac{\partial}{\partial \pi(S_0)} L(\pi, r) = -1 - \log \pi(S_0) + \sum_{\{j|j \in S_0\}} r_j + r_0 = 0$$

$$\Rightarrow \pi(S) = K.\exp\{ \sum_{\{j|j \in S\}} r_j \}$$

Maximum Entropy

Consider:

$$\text{Maximize } H(\pi) := -\sum_S \pi(S) \log \pi(S)$$

$$\text{Subject to } s_j(\pi) := \sum_{\{S|j\in S\}} \pi(S) \geq \lambda_j, \forall j \text{ and } \sum_S \pi(S) = 1$$

Lagrangian:

$$L(\pi, r) := -\sum_S \pi(S) \log \pi(S) - \sum_j r_j[\lambda_j - \sum_{\{S|j\in S\}} \pi(S)] + r_0[\sum_S \pi(S) - 1]$$

$$\frac{\partial}{\partial \pi(S_0)} L(\pi, r) = -1 - \log \pi(S_0) + \sum_{\{j|j\in S_0\}} r_j + r_0 = 0$$

$$\Rightarrow \pi(S) = K.\exp\{\sum_{\{j|j\in S\}} r_j\} \qquad \Rightarrow \text{ CSMA with } R_j = \exp\{r_j\}$$

Maximum Entropy

Consider:

$$\text{Maximize } H(\pi) := -\sum_S \pi(S) \log \pi(S)$$

$$\text{Subject to } s_j(\pi) := \sum_{\{S|j\in S\}} \pi(S) \geq \lambda_j, \forall j \text{ and } \sum_S \pi(S) = 1$$

Lagrangian:

$$L(\pi, r) := -\sum_S \pi(S) \log \pi(S) - \sum_j r_j[\lambda_j - \sum_{\{S|j\in S\}} \pi(S)] + r_0[\sum_S \pi(S) - 1]$$

$$\frac{\partial}{\partial \pi(S_0)} L(\pi, r) = -1 - \log \pi(S_0) + \sum_{\{j|j\in S_0\}} r_j + r_0 = 0$$

$$\Rightarrow \pi(S) = K.\exp\{\sum_{\{j|j\in S\}} r_j\} \quad \Rightarrow \text{ CSMA with } R_j = \exp\{r_j\}$$

Question: What are the r_j?

Answer: $r_j \approx \alpha X_j$ (if $\lambda \in \Lambda$)

Maximum Entropy

Consider:

$$\text{Maximize } H(\pi) := -\sum_{S} \pi(S) \log \pi(S)$$

$$\text{Subject to } s_j(\pi) := \sum_{\{S \mid j \in S\}} \pi(S) \geq \lambda_j, \forall j \text{ and } \sum_{S} \pi(S) = 1$$

Gradient to find Lagrange multipliers

$$L(\pi, r) := -\sum_{S} \pi(S) \log \pi(S) - \sum_{j} r_j [\lambda_j - \sum_{\{S \mid j \in S\}} \pi(S)] + r_0 [\sum_{S} \pi(S) - 1]$$

Maximum Entropy

Consider:

$$\text{Maximize } H(\pi) := -\sum_S \pi(S)\log\pi(S)$$

$$\text{Subject to } s_j(\pi) := \sum_{\{S|j\in S\}} \pi(S) \geq \lambda_j, \forall j \text{ and } \sum_S \pi(S) = 1$$

Gradient to find Lagrange multipliers

$$L(\pi, r) := -\sum_S \pi(S)\log\pi(S) - \sum_j r_j[\lambda_j - \sum_{\{S|j\in S\}} \pi(S)] + r_0[\sum_S \pi(S) - 1]$$

$$r_j(n+1) = [r_j(n) + \alpha(n)\{\lambda_j - s_j(\pi(n))\}]^+$$

Maximum Entropy

Consider:

$$\text{Maximize } H(\pi) := -\sum_S \pi(S) \log \pi(S)$$

$$\text{Subject to } s_j(\pi) := \sum_{\{S|j \in S\}} \pi(S) \geq \lambda_j, \forall j \text{ and } \sum_S \pi(S) = 1$$

Gradient to find Lagrange multipliers

$$L(\pi, r) := -\sum_S \pi(S) \log \pi(S) - \sum_j r_j[\lambda_j - \sum_{\{S|j \in S\}} \pi(S)] + r_0[\sum_S \pi(S) - 1]$$

$$r_j(n+1) = [r_j(n) + \alpha(n)\{\lambda_j - s_j(\pi(n))\}]^+$$

$$r_j(n+1) \approx [r_j(n) + \alpha(n)\{arrivals_j(n) - services_j(n)\}]^+$$

Maximum Entropy

Consider:

$$\text{Maximize } H(\pi) := -\sum_{S} \pi(S) \log \pi(S)$$

$$\text{Subject to } s_j(\pi) := \sum_{\{S|j \in S\}} \pi(S) \geq \lambda_j, \forall j \text{ and } \sum_{S} \pi(S) = 1$$

$$r_j(n+1) \approx [r_j(n) + \alpha(n)\{arrivals_j(n) - services_j(n)\}]^+$$

Maximum Entropy

Consider:

$$\text{Maximize } H(\pi) := -\sum_{S} \pi(S) \log \pi(S)$$

$$\text{Subject to } s_j(\pi) := \sum_{\{S|j\in S\}} \pi(S) \geq \lambda_j, \forall j \text{ and } \sum_{S} \pi(S) = 1$$

$$r_j(n+1) \approx [r_j(n) + \alpha(n)\{arrivals_j(n) - services_j(n)\}]^+$$

If $\alpha(n) = \alpha$:

$$r_j(n+1) \approx [r_j(n) + \alpha\{arrivals_j(n) - services_j(n)\}]^+$$

Maximum Entropy

Consider:

$$\text{Maximize } H(\pi) := -\sum_S \pi(S) \log \pi(S)$$

$$\text{Subject to } s_j(\pi) := \sum_{\{S|j\in S\}} \pi(S) \geq \lambda_j, \forall j \text{ and } \sum_S \pi(S) = 1$$

$$r_j(n+1) \approx [r_j(n) + \alpha(n)\{arrivals_j(n) - services_j(n)\}]^+$$

If $\alpha(n) = \alpha$:

$$r_j(n+1) \approx [r_j(n) + \alpha\{arrivals_j(n) - services_j(n)\}]^+$$

Also, $X_j(n+1) = [X_j(n) + \{Arrivals_j(n) - Services_j(n)\}]^+$

Thus, $r_j(n) \approx \alpha X_j(n)$

Maximum Entropy

Consider:

$$\text{Maximize } H(\pi) := -\sum_S \pi(S) \log \pi(S)$$

$$\text{Subject to } s_j(\pi) := \sum_{\{S \mid j \in S\}} \pi(S) \geq \lambda_j, \forall j \text{ and } \sum_S \pi(S) = 1$$

Theorem

- Solution exists if $\lambda \in \Lambda$
- Moreover,

$$\pi(S) = K \exp\{ \sum_{\{j \mid j \in S\}} r_j \}$$

- Also,

$$r_j \approx \alpha X_j$$

$$\Rightarrow \text{ CSMA with } R_j = \exp\{\alpha X_j\}$$

Conclusions

* Random allocations with adaptive requests rates are ε-optimal in utility
  * The request rates increase with the backlog
  * Congestion control imposes a price based on backlog in the ingress node
  * This price make the scheme almost strategy-proof in a large system
* Processing networks are scheduled based on virtual queues
  * These queues can become negative

References

* CSMA & Product-Form
  * R.R. Boorstyn et al, 1987
  * X. Wang & K. Kar, 2005
  * S. Liew et al., 2007
* MWM
  * Tassiulas & Ephremides, 1992
* Primal-Dual Decomposition of NUM
  * Kelly et al., 1998
  * Chiang-Low-Calderbank-Doyle, 2007
* Backpressure Protocols + NUM
  * Lin & Shroff, 2004
  * Neely-Modiano-Li; Eryilmaz-Srikant; Stolyar 2005

References

* Adaptive-CSMA
  * Jiang, Walrand 2008
* Improvements of Adaptive-CSMA
  * Ni-Tan-Srikant 2009 (Combined with LQF)
  * Jiang-Shah-Shin-Walrand 2010 (Positive Recurrence)
* Adaptive-CSMA with collisions
  * Ni-Srikant; Jiang-Walrand; Liu et al. 2009
* Implementations
  * Warrier-Ha-Wason-Rhee, 2008*
  * Lee-Lee-Yi-Chong-Proutiere-Chiang, 2009

References

Monographs

* Jiang-Walrand. Scheduling and Congestion Control for Wireless and Processing Networks. Morgan-Claypool 2010.
* Neely. Stochastic Network Optimization with Application to Communication and Queueing Systems. Morgan Claypool 2010.
* Pantelidou-Ephremides. Scheduling in Wireless Networks. NOW, 2011