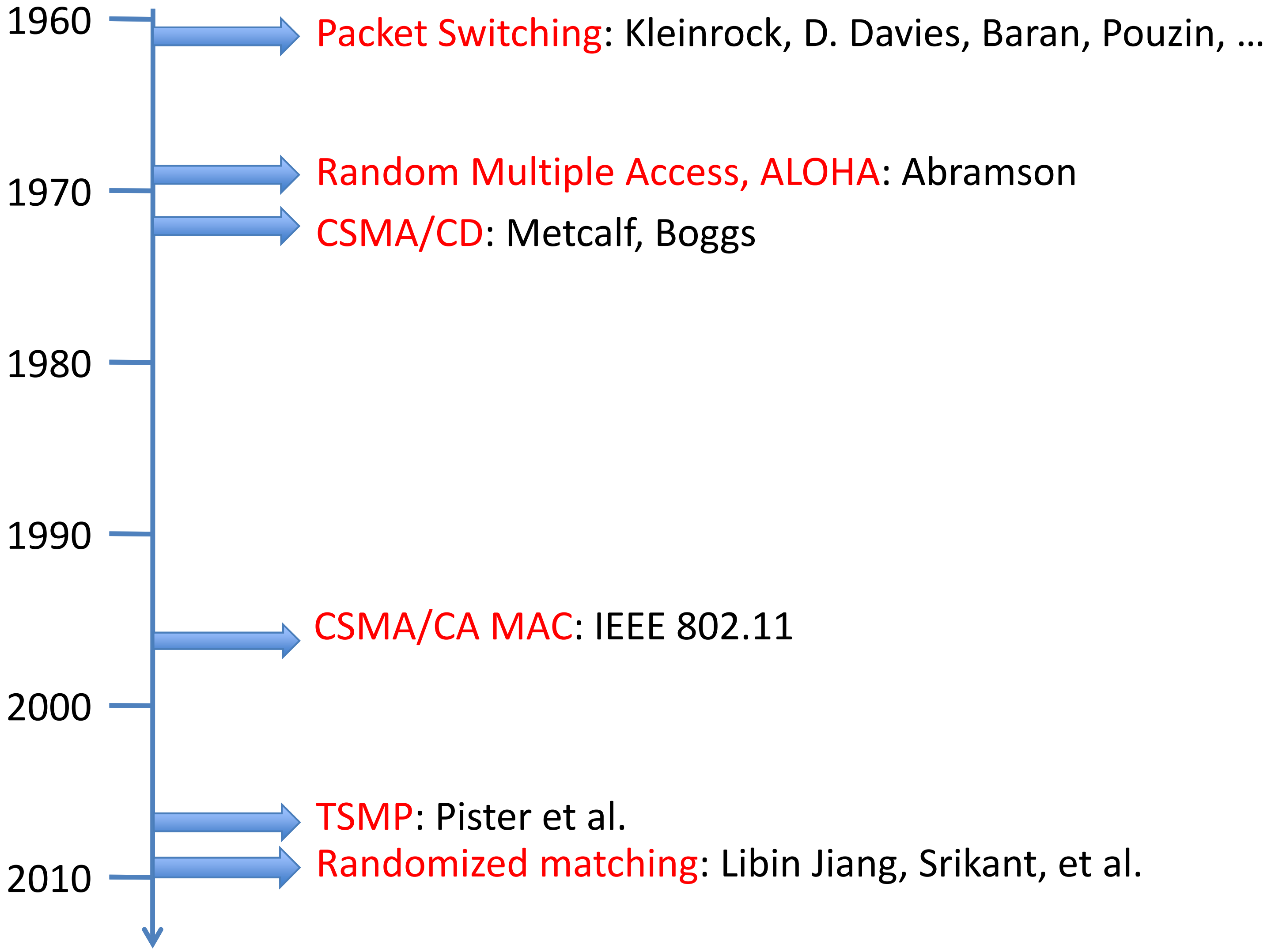
Ad-Hoc Mac:
Why and How?

Jean Walrand

University of California, Berkeley

WCNC, Doha, April 4, 2016

A Brief History of MACs:



1960

Packet Switching: Kleinrock, D. Davies, Baran, Pouzin, …

Random Multiple Access, ALOHA: Abramson

1970

CSMA/CD: Metcalf, Boggs

1980

1990

CSMA/CA MAC: IEEE 802.11

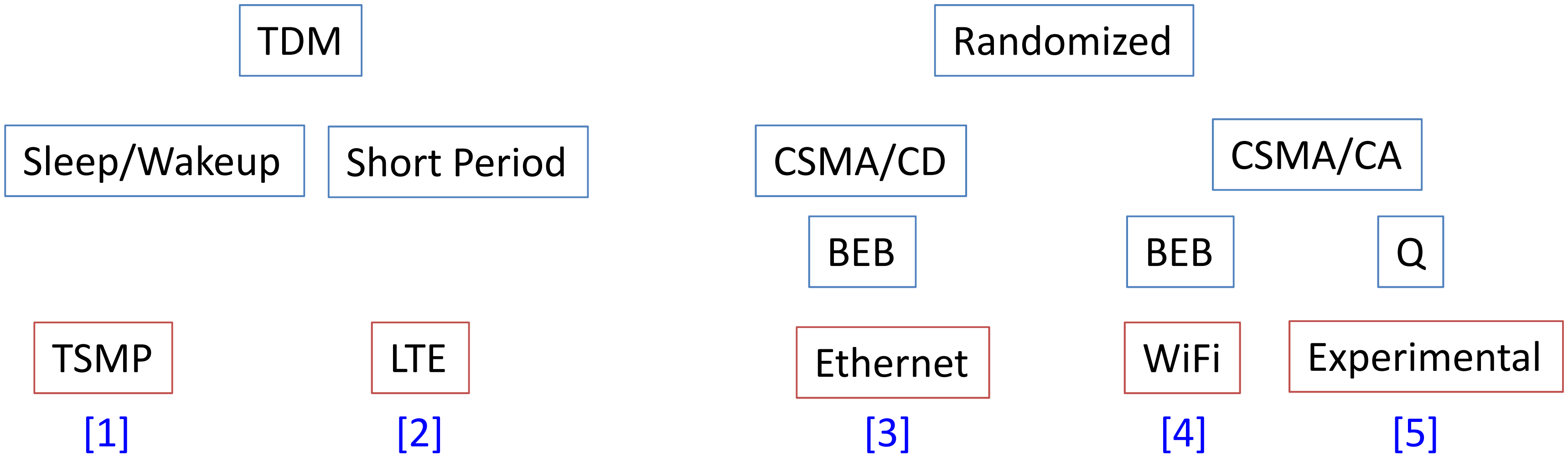
2000

TSMP: Pister et al.

Randomized matching: Libin Jiang, Srikant, et al.

2010

MAC Mechanisms



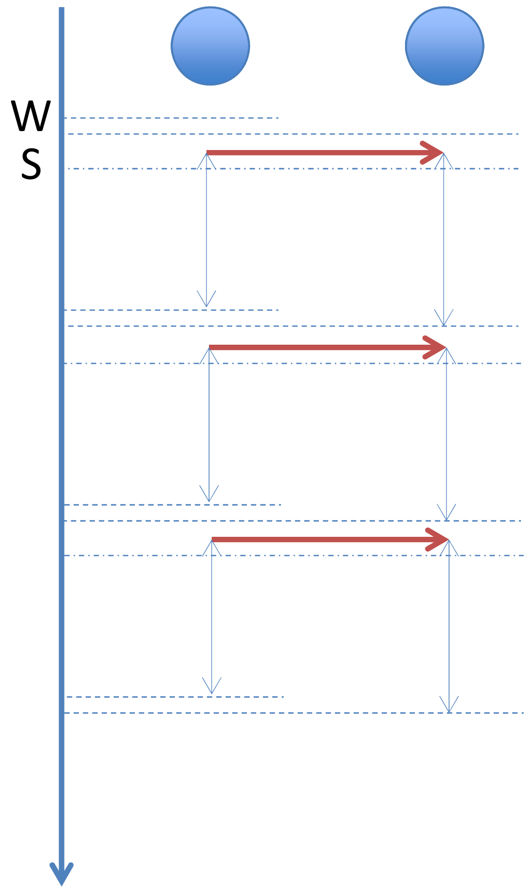
	Best			Worst
Energy	[1]			[2-5]
Delay	[2*] [3-4$^{\#}$]	[3-4*]	[5]	[1]
Throughput	[5]			[1]
Fairness	[1,2,5]			[4^{+}]

*In high utilization
$^{\#}$In low utilization

$^{+}$In ad hoc mode

TDM

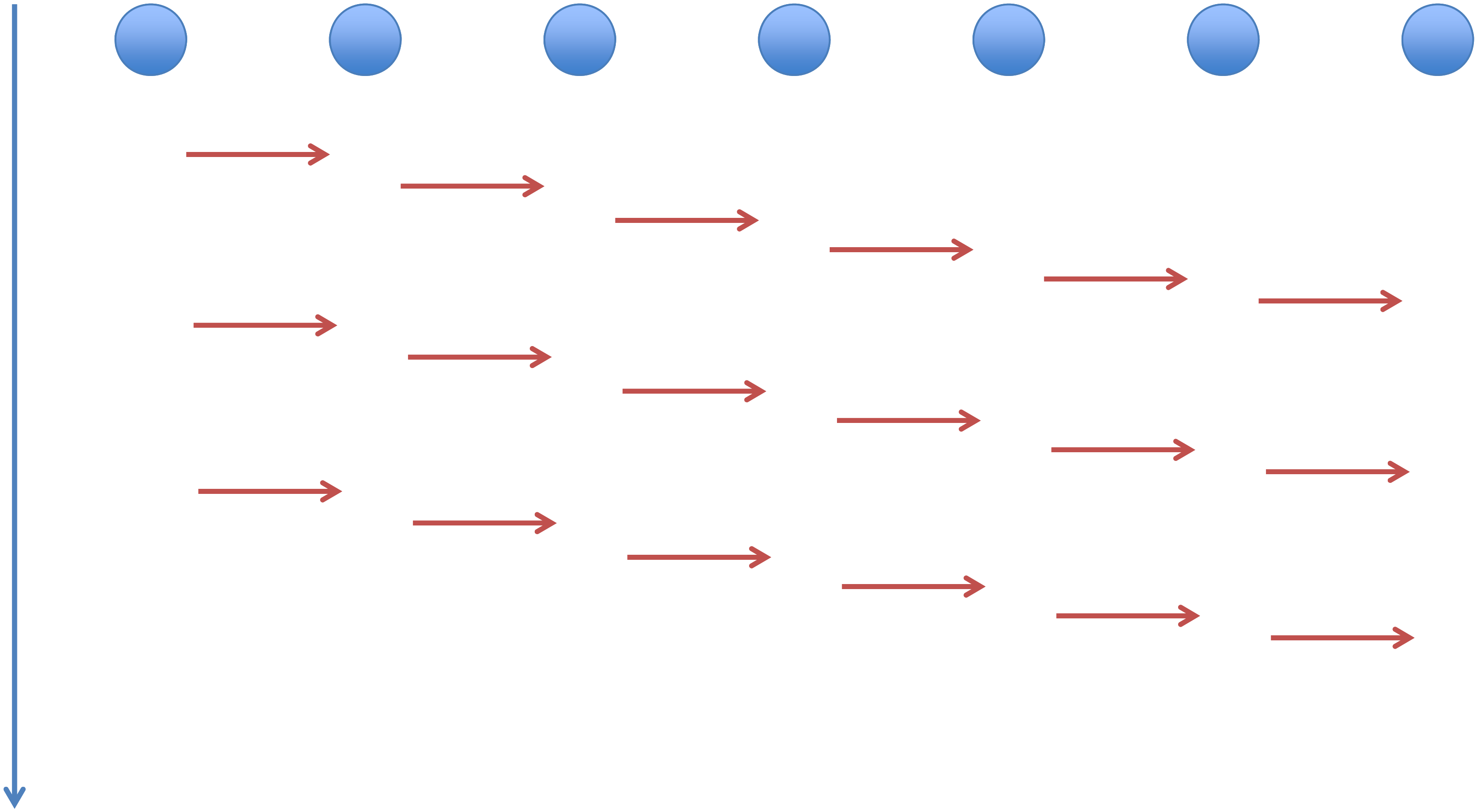
Sleep and wakeup



Time

TDM

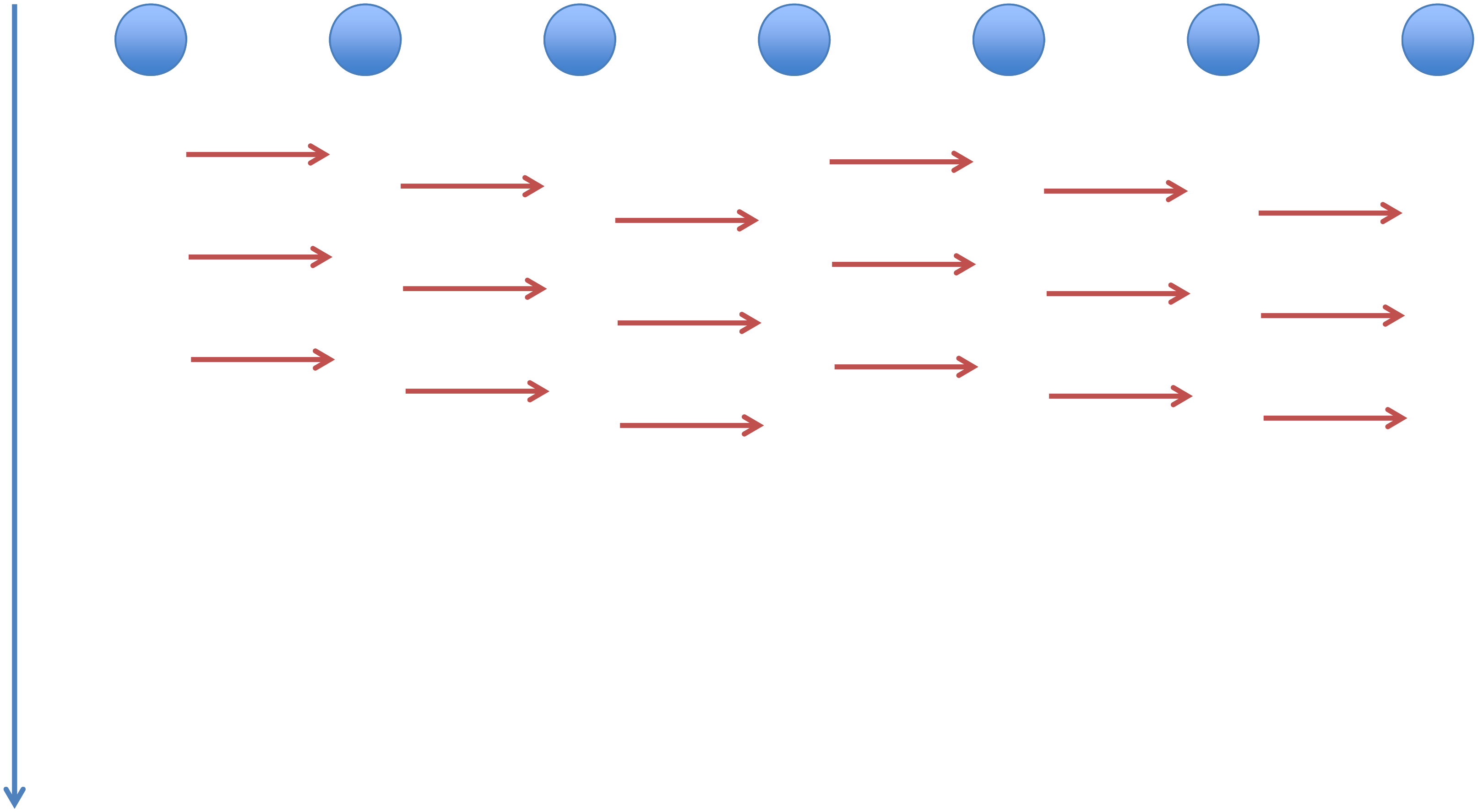
In a network



Time

TDM

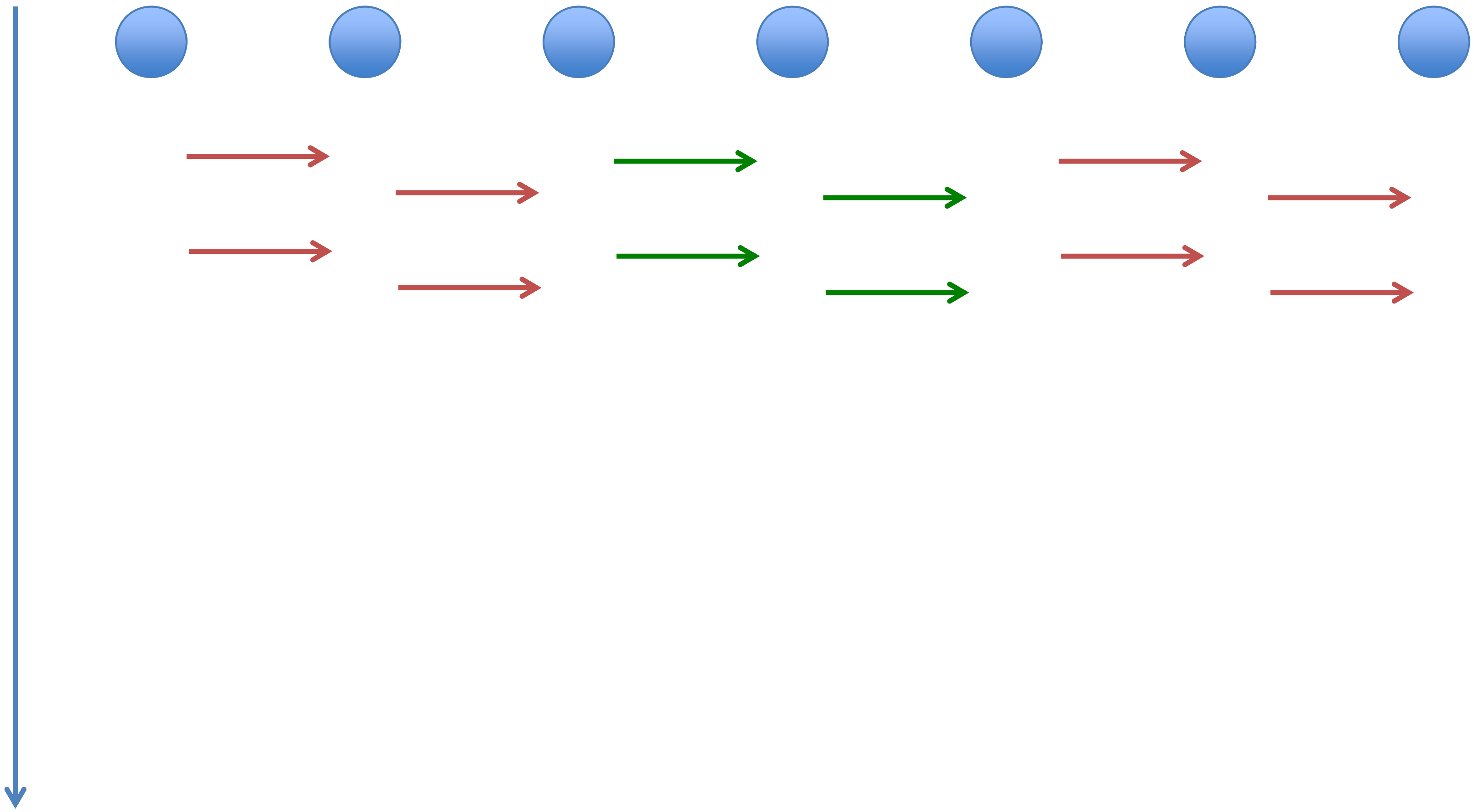
Pipelining



Time

TDM

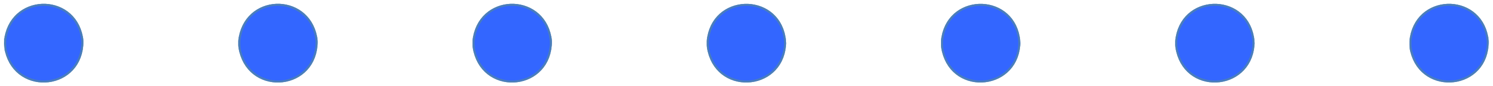
With multiple frequencies



Time

TDM

Self-organizing?



Nodes know their identity: 1, 2, …

1) Wake up nodes

2) Nodes exchange hello messages at random times to sync their clocks

3) Nodes measure their interference (1 transmits at time 0, others listen…)

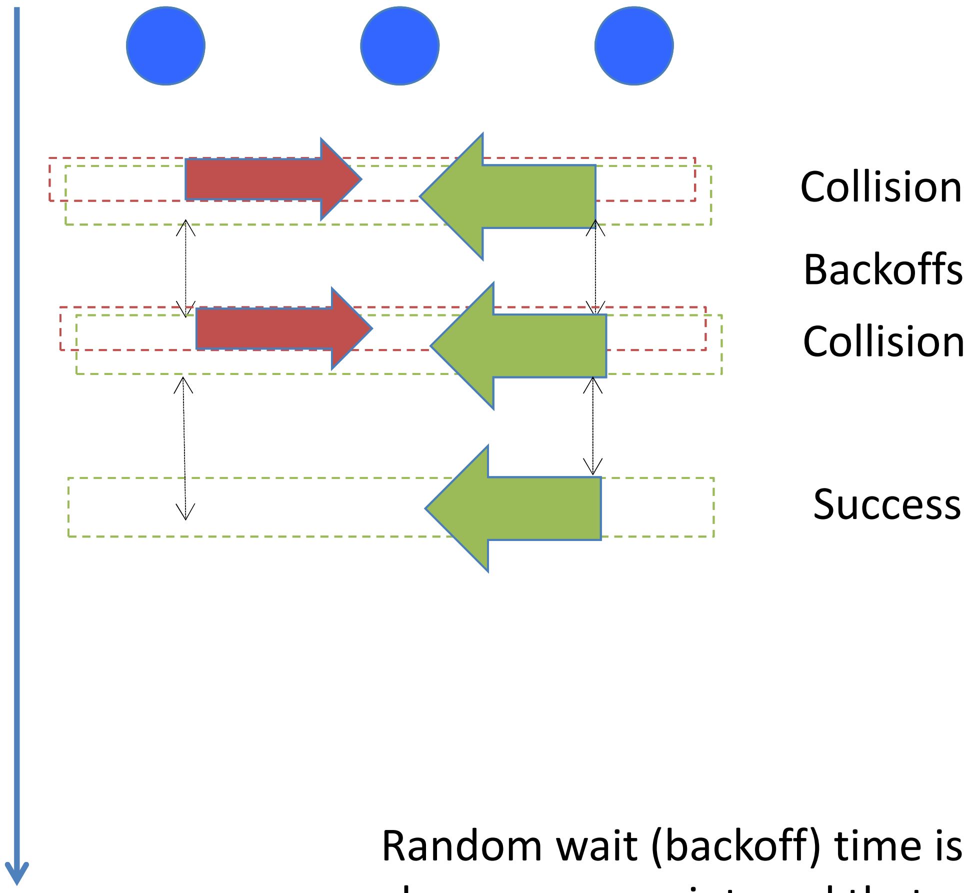
4) Nodes report interference pattern to server in “safe” mode

5) Server computes suitable timing of transmission

6) Server informs nodes in “safe” mode.

Random **CSMA/CA** **BEB**

Assume nodes all hear each other.



Collision

Backoffs

Collision

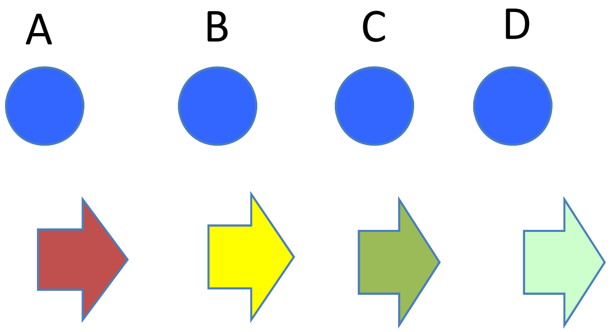
Success

Time

Random wait (backoff) time is chosen over an interval that doubles after every collision of the same packet transmission.

Random CSMA/CA BEB

Unfair to “nodes in the middle”



Assume nodes hear each other 2 steps away. Thus, A hears B and C, etc.

Say next backoffs are iid uniform in {1, 2}.

A succeeds if backoffs are (1, 2, 2, *)

B succeeds if backoffs are (2, 1, 2, 2)

C succeeds if backoffs are (*, 2, 1, 2)

➔ The probability of success is smaller for nodes in the middle.

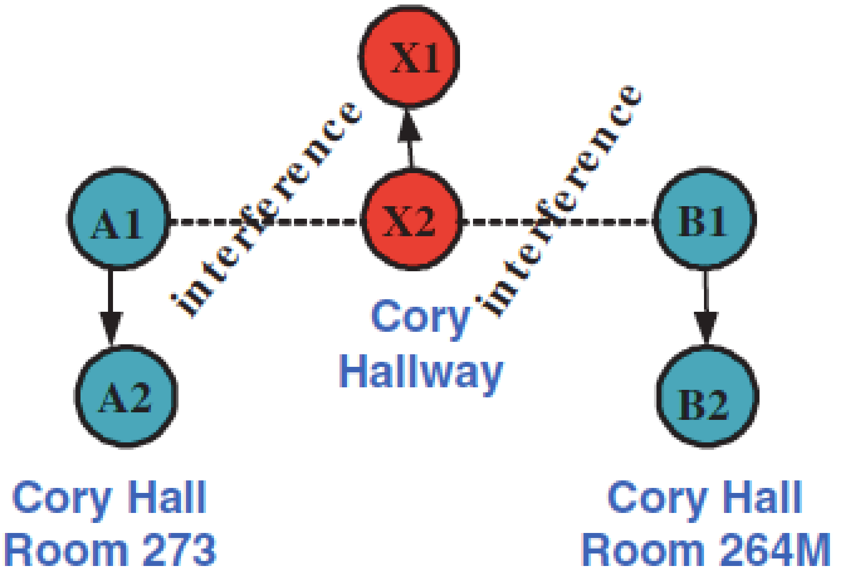
Random

CSMA/CA

BEB

Unfair to “nodes in the middle”

WiFi experiment:



Active Link	Received on A	Received on B	Received on X
A	6		
A,B	6	6	
A,X	3		3
A,B,X	4	4	2

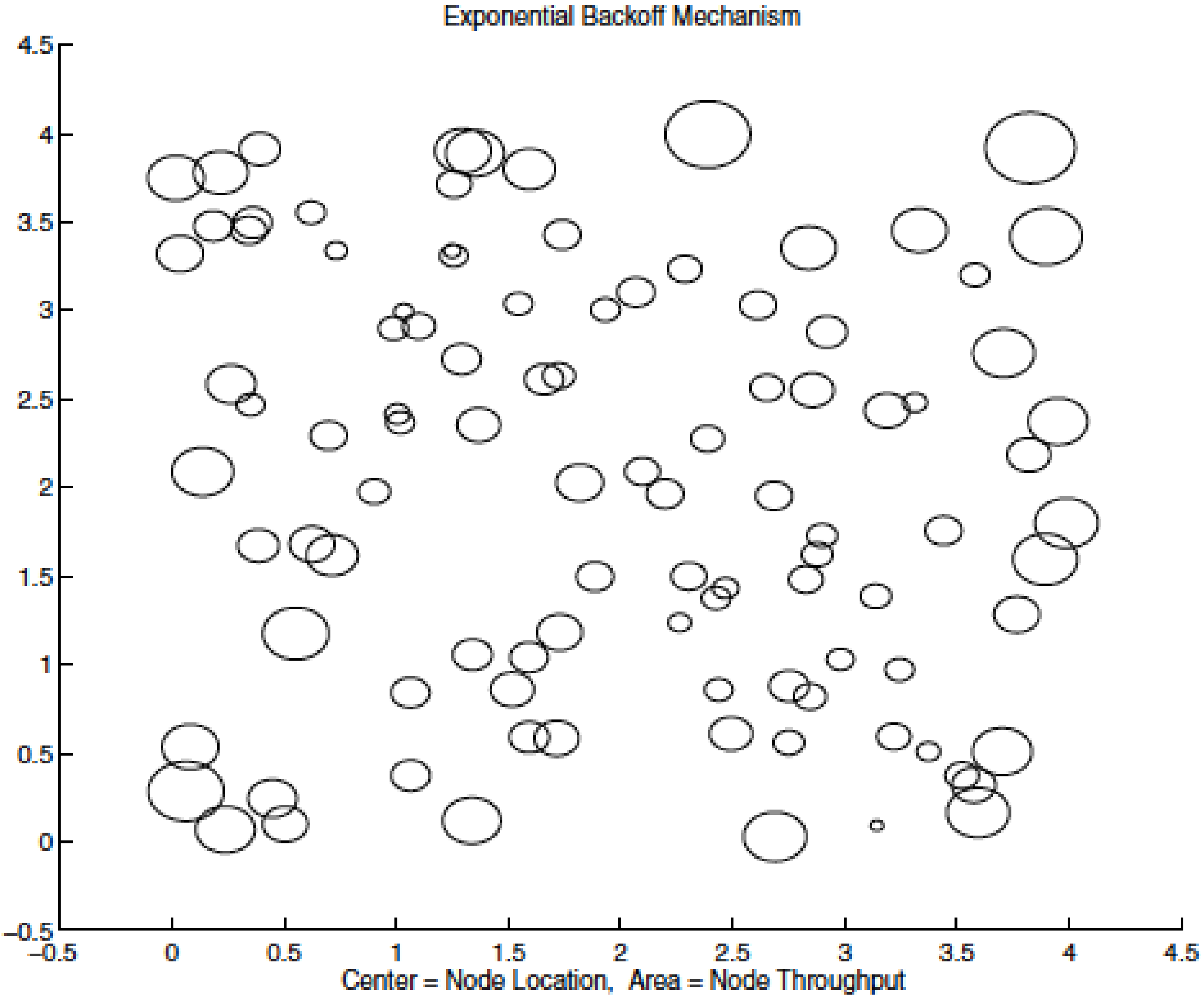
Random

CSMA/CA

BEB

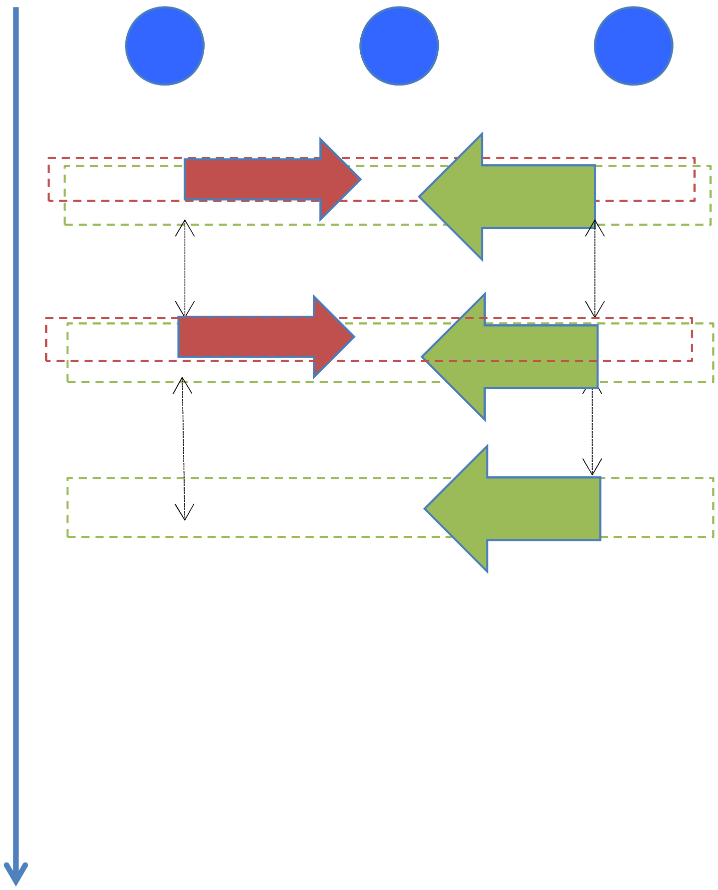
Unfair to “nodes in the middle”

Simulation:



Gupta, Walrand: Impatient Backoff Algorithm: Fairness in a Distributed Ad Hoc MAC. ICC, July 2007

Random CSMA/CA Q



Time

BEB

Random wait (backoff) time is chosen over an interval that doubles after every collision of the same packet transmission.

Q

Random wait (backoff) time is chosen over an interval that decreases with node’s backlog.

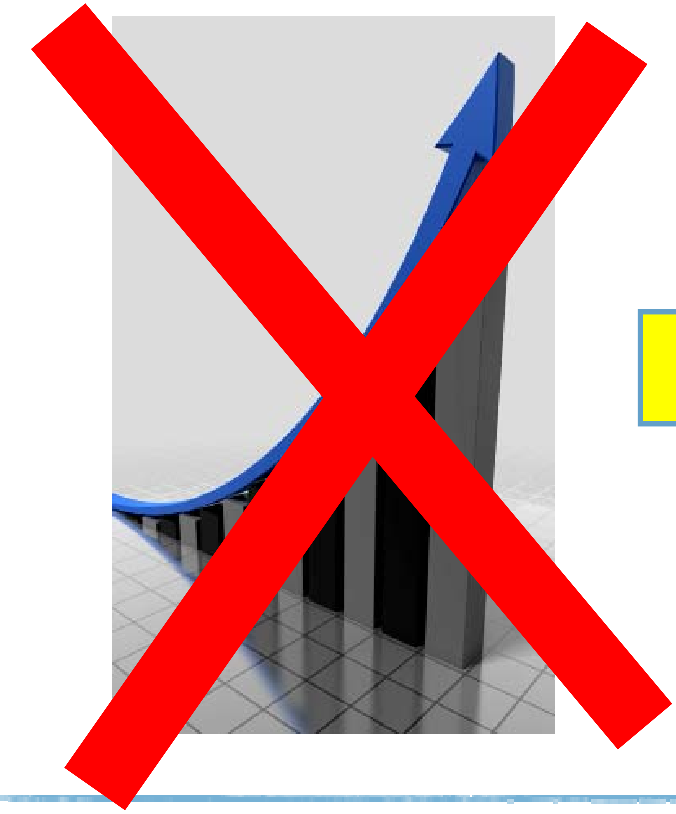
Libin Jiang, Jean C. Walrand: Convergence and stability of a distributed CSMA algorithm for maximal network throughput. CDC 2009

Adaptive CSMA

Libin Jiang

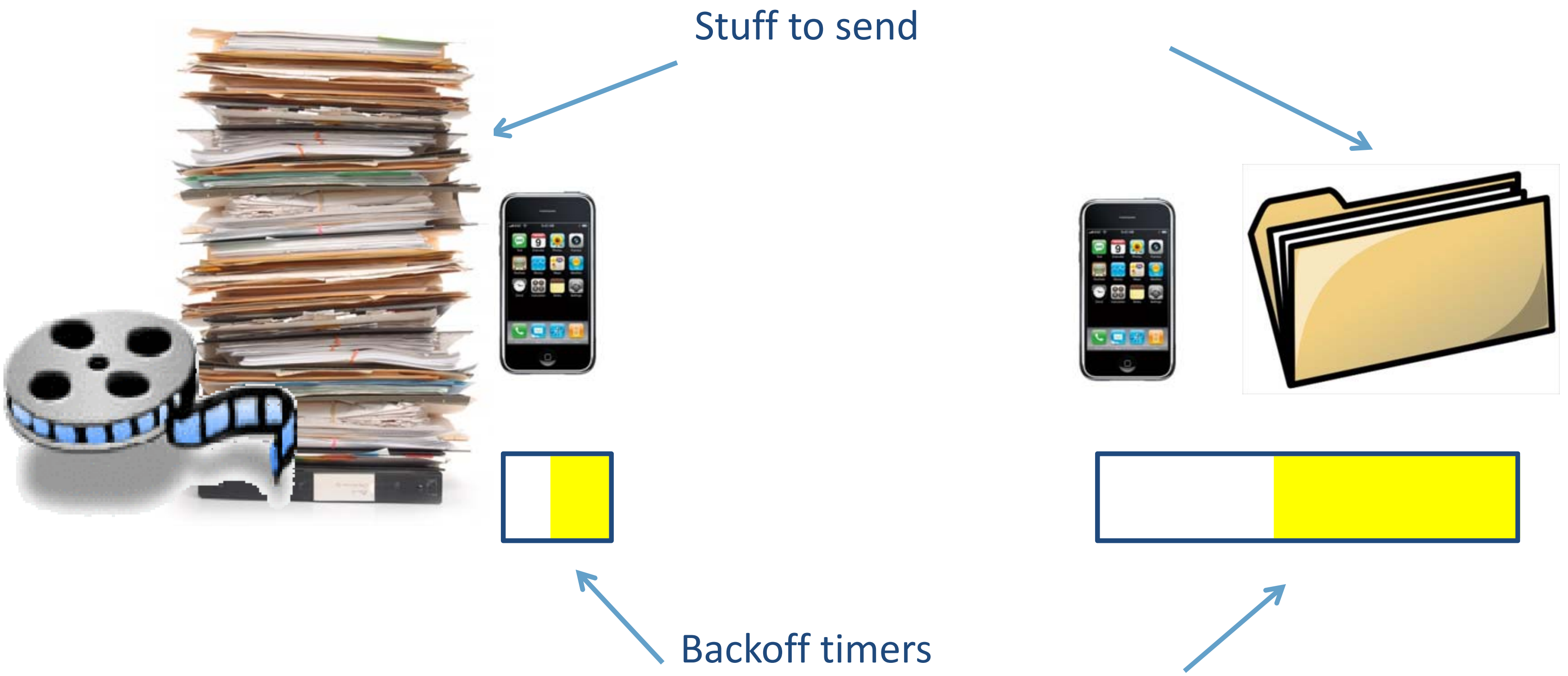
Replace

Exponential Backoff by Queue-Based Backoff



Adaptive CSMA

A node with a bigger backlog should choose its random backoff in a smaller range.



Stuff to send

Backoff timers

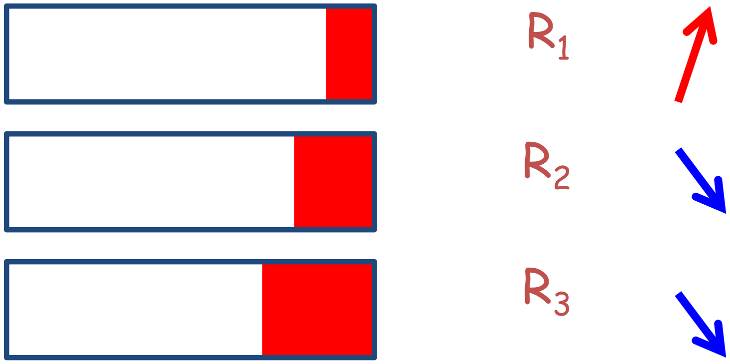
Adaptive CSMA

The node with a bigger backlog will get a higher throughput.

Adaptive CSMA

Let R = 1/(average timer value)
= rate of transmission attempts

Adjust R so that the link transmits fast enough.

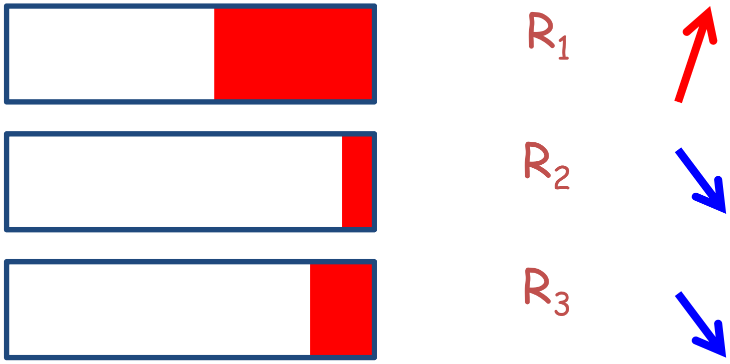


Backlog increases → increase R.
Backlog decreases → decrease R.

Adaptive CSMA

Let R = 1/(average timer value)
= rate of transmission attempts

Adjust R so that the link transmits fast enough.



Backlog increases → increase R.
Backlog decreases → decrease R.

Adaptive CSMA

How to choose the values of R?	$R = \text{Exp}\{\alpha.\text{Backlog}\}$
Do we need to know all the backlogs?	No
Could some nodes starve other nodes?	No
Can such a scheme be efficient and fair?	Yes

Dual Algorithm

Complex Problem: Congestion Control
Simple Solution: TCP: AIMD

Max Σ_j $U_j(x_j)$ s.t. Rate on link k < C_k

Dual algorithm → Local Solution

Frank Kelly

- Each link k posts a p

- Each user j chooses price

$U_j(x$

Dual Algorithm

Much subsequent work; backpressure protocols
(see references)

Related idea: Maximize drift of Lyapunov function
(see references)

Dual Algorithm

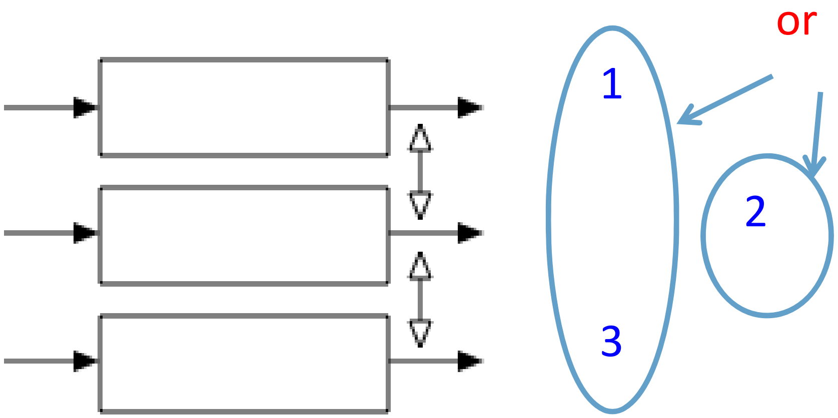
Complex Problem: Scheduling Conflicting Links

Simple Solution: MWM

(Maximum Weighted Matching)

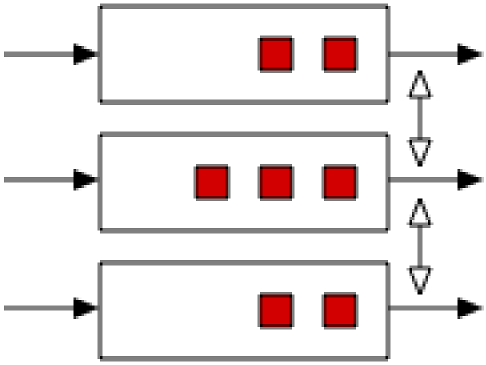
MWM

Conflicting nodes: At any time, either {2} or {1, 3} can transmit. Maximal Independent Sets.



At any time, the links in the maximal independent set with the largest sum of queue lengths transmit.

MWM



Two problems:

1) Finding out the backlogs of other nodes;
2) Finding the independent set with maximum sum.

At any time, the links in the maximal independent set with the largest sum of queue lengths transmit.

Property: Stabilizes the queues.
- [Tassiulas, Ephremides 92]

Dual Algorithm

Complex Problem: Random Access

Simple Solution: Adaptive CSMA

$$\text{Max } H(\pi(R)) \text{ s.t. } s_j > \lambda_j$$

Service rate at node j

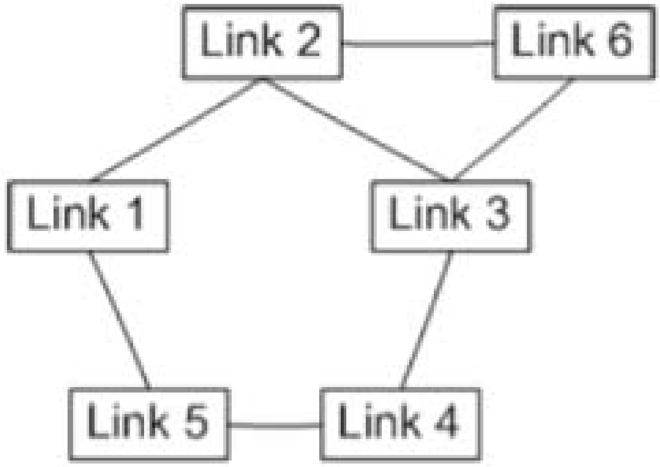
Entropy of distribution of independent sets

Dual Algorithm → Local Solution

R_j updated based on backlog of j

$R_j \approx \text{Exp}\{\alpha X_j\}$

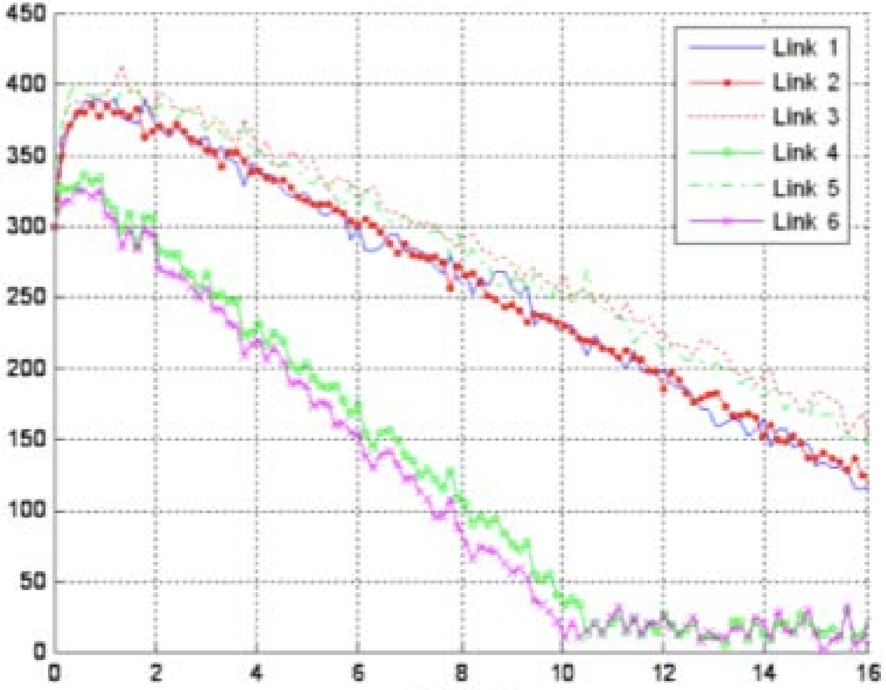
Example



λ=0.98*(0.5,0.2,0.5,0.3,0.5,0.3) †

Network

Queue Lengths

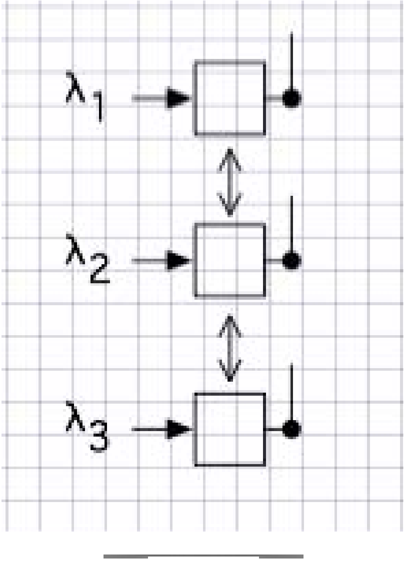


Time

† 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}

Congestion Control + Scheduling



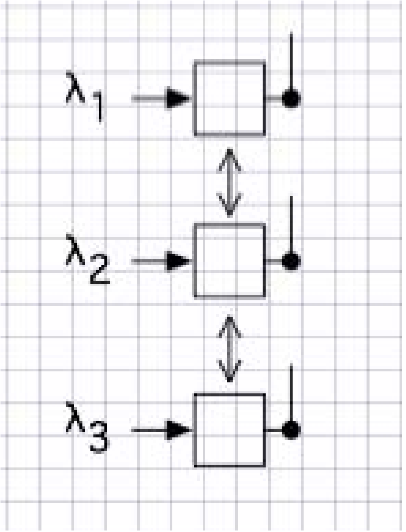
- Links want to maximize the “total utility”

  $u_1(\lambda_1) + u_2(\lambda_2) + u_3(\lambda_3)$

- Congestion control + scheduling

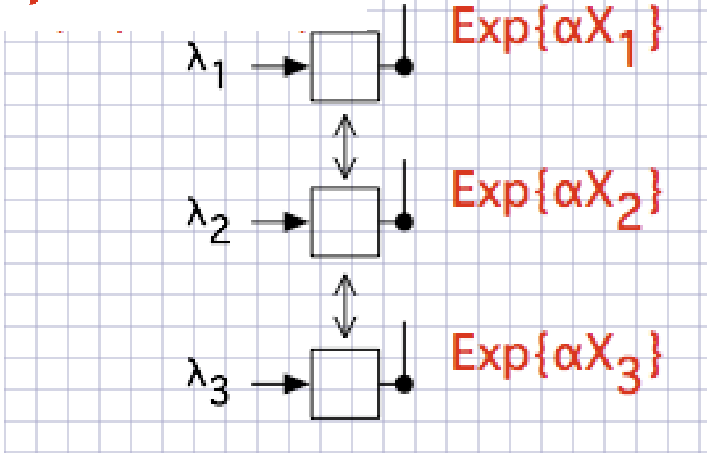
  [Adjust arrival rates]

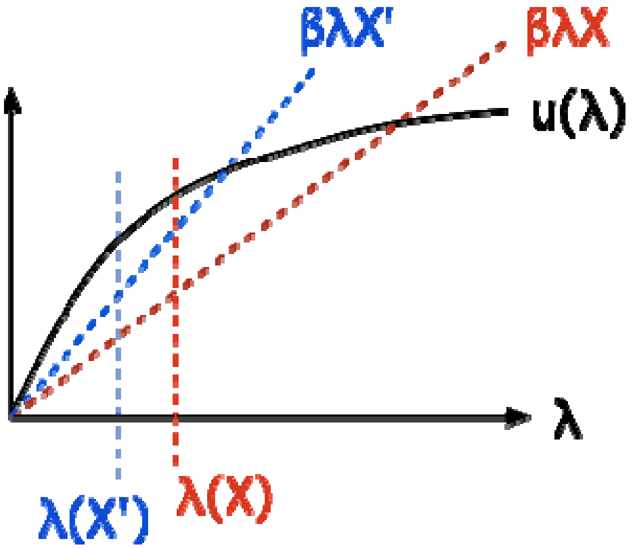
Congestion Control + Scheduling



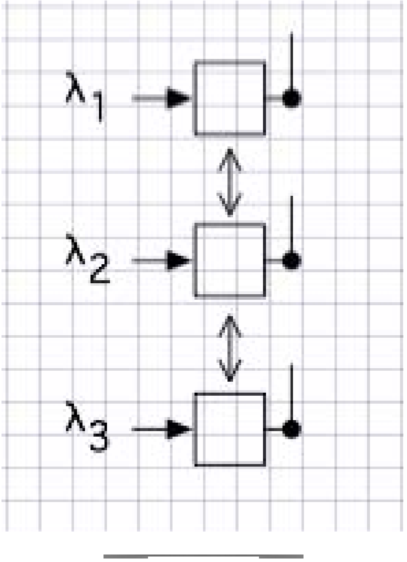
Node i maximizes

$$u_i(\lambda_i) - \beta\lambda_i X_i$$



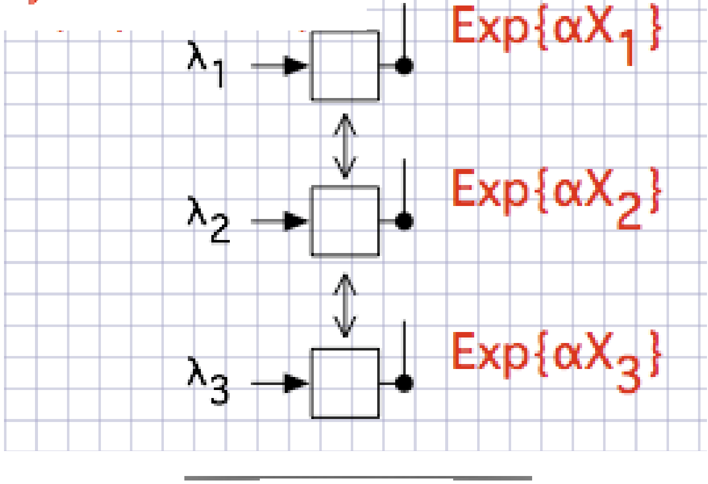


Congestion Control + Scheduling



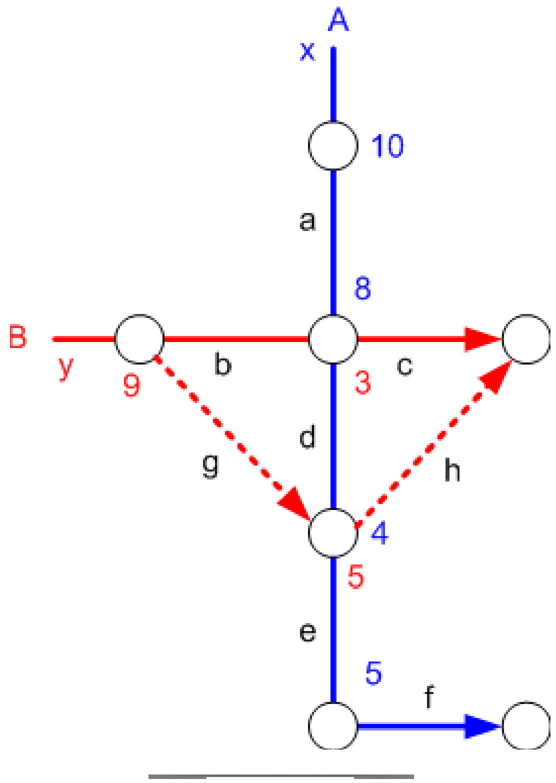
Node i maximizes

$$u_i(\lambda_i) - \beta\lambda_i X_i$$



- Approach: Q-CSMA + input rate control
- Fact: (Essentially) achieves maximum utility

Congestion Control + Scheduling + Routing

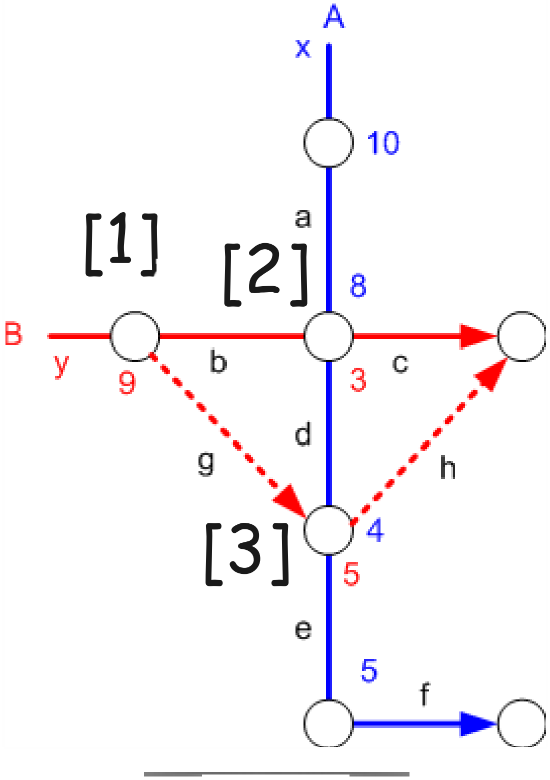


Wireless links, with interference

Goal: maximize total utility of flows

Congestion control, scheduling, and routing

Congestion Control + Scheduling + Routing



Node [1]:

If (9 - 5)C(g) > (9 - 3)C(b): [1] -> [3];
Else: [1] -> [2]

Let b[1] = max{(9 - 5)C(g), (9 - 3)C(b)}

Let T[1] be exponentially distributed
with rate Exp{ α b[1] }

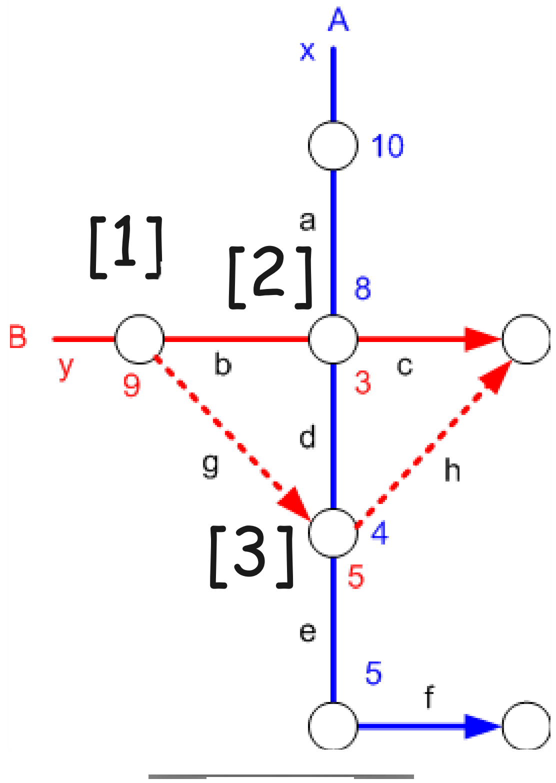
Node [2]:

If (8 - 4)C(d) > (3 - 0)C(c): (d); Else: (c)

Let b[2] = max{(8 - 4)C(d), (3 - 0)C(c)}

Let T[2] be exponentially distributed
with rate Exp{ α b[2] }

Congestion Control + Scheduling + Routing



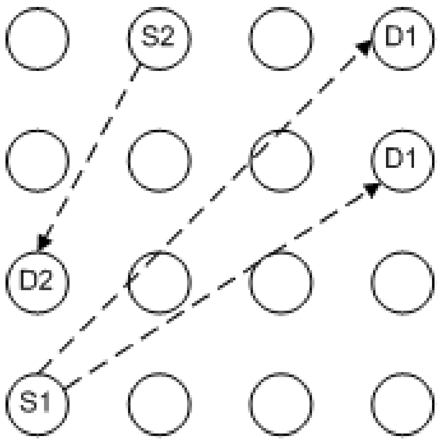
Then, use CSMA with those backoff delays.

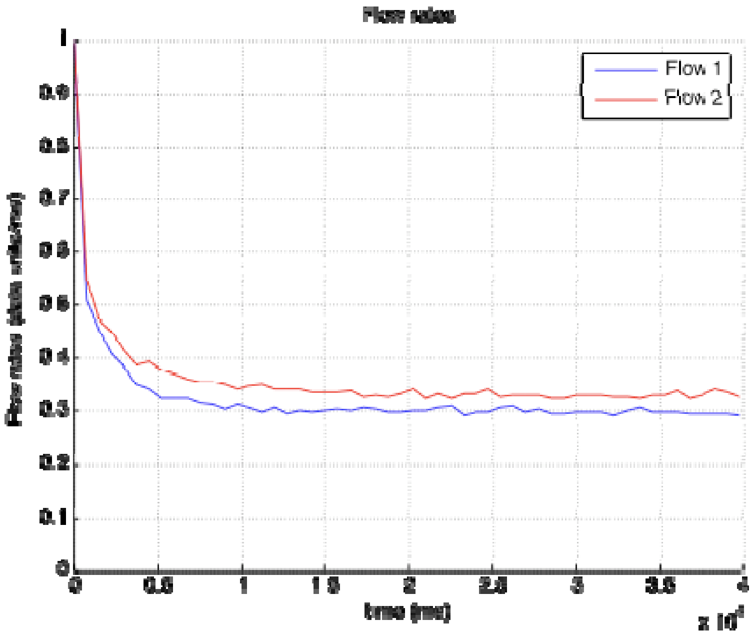
Also, A chooses x that maximizes $U_A(x) - \beta 10x$;

B chooses y s.t....

Fact: (Essentially) achieves maximum utility

Congestion Control + Scheduling + Routing





Multipath routing allowed

Unicast S2 -> D2
Anycast S1 to any D1

Summary

- Each node:
- Adjusts incoming rate: maximize $u(\lambda) - \lambda \beta X$
- Calculates, for every flow
  backpressure = link rate * (trans. Q - receiver Q)
- Chooses flow with max. backpressure B
- Generates backoff with mean = $1/\exp\{\alpha B\}$
- Then uses CSMA
- Fact: (Essentially) maximizes sum of utilities

Status

- Compatibility with TCP

  B: Using max Σ(log(1 + x)) [Srikant]
- Reduce Delays: B+: Placeholder packets, virtual arrivals
- Routing: C+: Avoid long paths
- Collisions

  Without hidden nodes: A: [Srikant et al., LJ-JW]

  With hidden nodes: B: Using RTS/CTS; [Kim et al.]
- Unreliable links: A
- Changing links: C
- Multicast, Unicast: A
- Implementations: Promising examples (Rhee, Chiang, …)

Summary

Goals:

Distributed algorithms for network control

Efficiency and fairness: Utility maximization

Tools:

Dual program

Lyapunov method

Entropy relaxation for randomized matching

Reference:

J. Walrand and A. Parekh, “Congestion control, routing and scheduling in communication networks: a tutorial,” *IEICE Trans.*, 2013.


thank you!