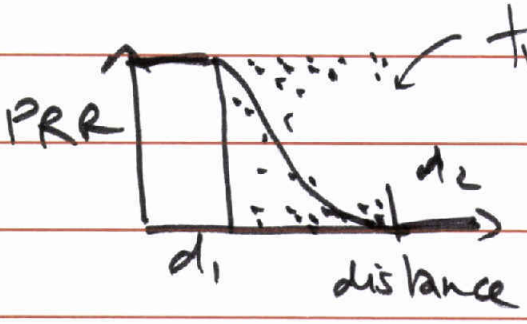
EES97 7/25/16

Network Layer for Wireless Networks

- Blacklisting
- ETX as a metric



first-order approximation:

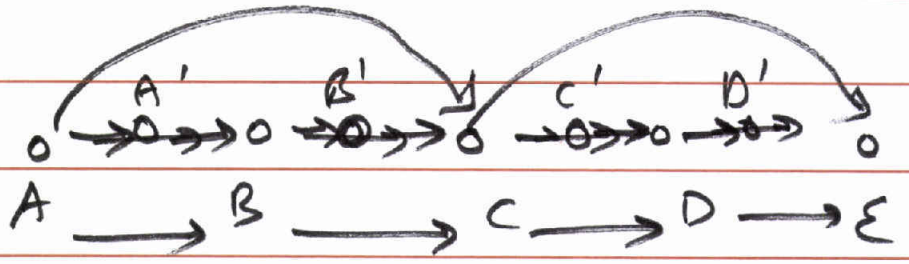
- if PRR is an iid r.v.
- if ACK pkts always received

$$ETX \sim \frac{1}{PRR}$$

in practice it is measured as a moving average of instantaneous ETX

[empirically ~0.8 - 0.9]

$$ETX(n+1) = \alpha ETX(n) + (1-\alpha) ETX_{inst.}$$



min-hop count routing may prefer longer hops

min-ETX routing balances both hop-distance & # of hops.

ETX favors:

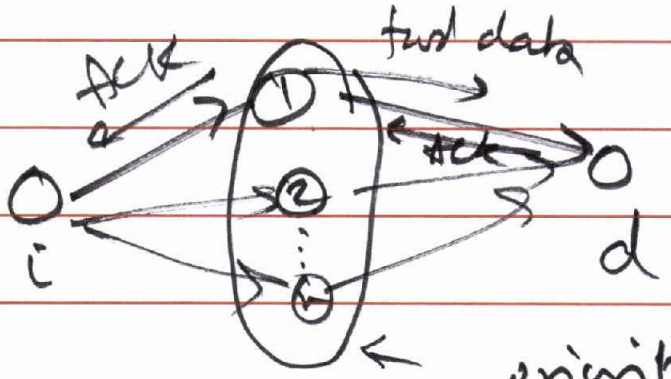
- Reliability
- Energy
- Latency
- Throughput

min ETX routing can be implemented w/ standard Dijkstra / Bellman Ford Algorithm (i.e. Link State or Distance Vector protocols)

—

Anypath routing can reduce total end to end ETX even further

- uses the diversity of receivers / exploits the broadcast advantage in wireless.



prioritized forwarding set-

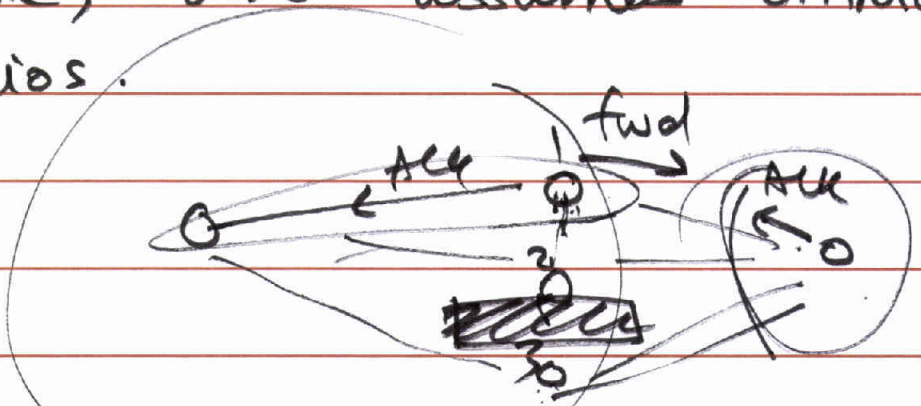
Requires some coordination problem → typically implemented via timer + ACK

Anypath routing needs modifications of Dijkstra/BF to work.

We discussed shortest-Anypath first algorithm which generalizes Dijkstra's algorithm.

—

Anypath routing assumes a high-density / non-sparse network, also assumes omnidirectional radios.



Anypath routing may introduce some additional latency.

Cooperative Caching / Barrage Relay

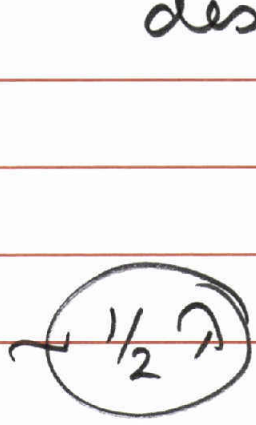
- Barrage relay networking
- Glossy
  - implemented in Trelliware Radios
  - used in Wireless Sensor Networks.

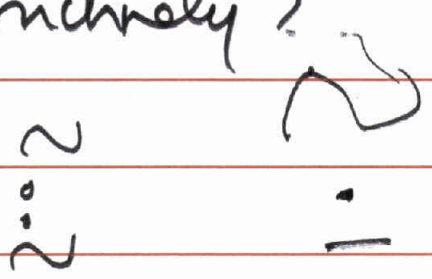
1. Time is Synchronized Globally
2. At any time, the entire network may be focused on helping just one packet be broadcasted throughout the network.
3. At each scheduled slot, all nodes with the pkt will transmit the pkt.

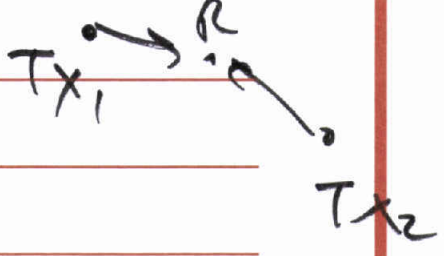
Key idea: Synchronized transmission of identical signals from multiple sources increases the effective-SNR, improving range & reliability

accurate!

1. How close does sync have to be?
2. Can the signals combine destructively?





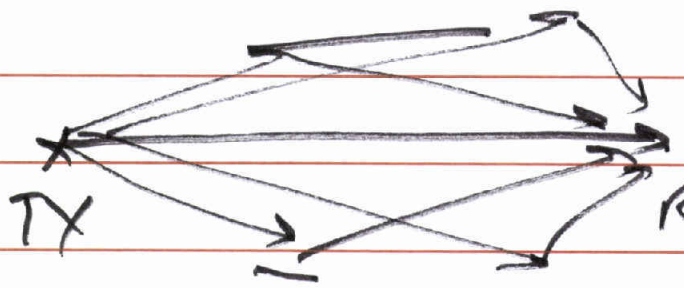


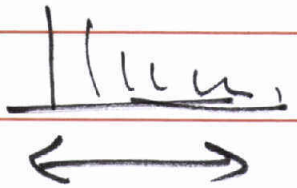
$\Delta T < \frac{1}{f_c}$ $\frac{1}{2 \times 10^9 \, s^{-1}}$

$\sim 10^{-9} \, s$

~ ns accuracy

At the layer, in fact, we already deal w/ longer delay spreads!





Phase Dithering: transmitters may intentionally introduce a random phase shift in the signal.

$$\tilde{A} + \tilde{B} = \tilde{C}$$

on average is louder than either in isolation

Encephally:

- Synchronization (typically only to symbol level, but not to $\frac{1}{f_c}$ level)

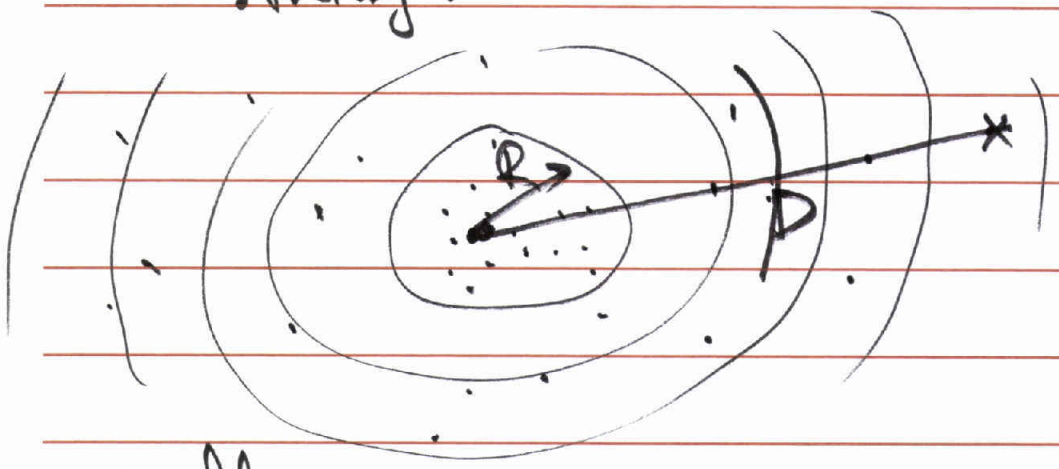
possible w/ GPS

- Can improve performance by

a) phase dithering
b) phy/link layer approaches to handle delay spreads.

The practical implementations: Trellis wave radios & Glossy have been shown to be very robust & highly reliable.

Can show in theory that the total time needed is much less for cooperative-broadcast routing.

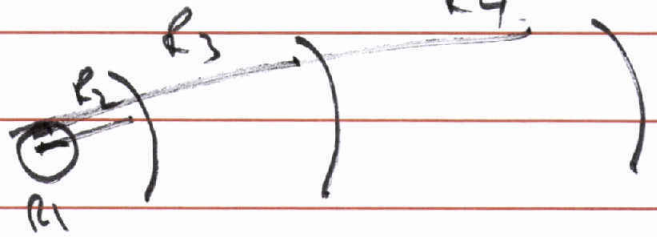


Traditionally

Time to broadcast ∝ hop-radius of the network.

$\frac{D}{R}$ ← hop radius

For cooperative broadcast the range R increases at each iteration, geometrically.



$R_{n+1} > R_n$ as more nodes transmit at each round.

Traditional broadcast: Time $\propto \frac{D}{R}$

w/ cooperative broadcast: Time $\propto \log\left(\frac{D}{R}\right)$

↑ exponentially faster.

Backpressure Scheduling and Routing.

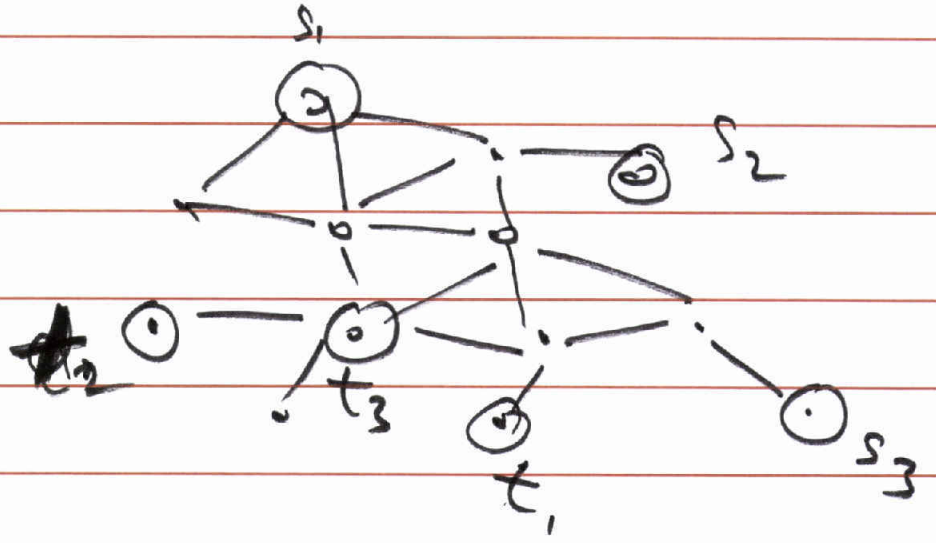
"Queue-aware scheduling/routing"

paper from ~1993 Tassiulas & Ephremides

Max Weight Algorithm

This algorithm can stably schedule any arrival throughput/rate vector that can be scheduled by any other algorithm in a stable manner.

Given a wireless network



Say we have a set of source-destination pairs : $(s_1, t_1) \ldots (s_n, t_n)$
each has a unicast flow associated with it: $f_1 \ldots f_n$.

Arrival rate vector: $(r_1 \ldots r_n)$
a set of data rates, one for each flow that describes arriving traffic at ~~not~~ the source nodes

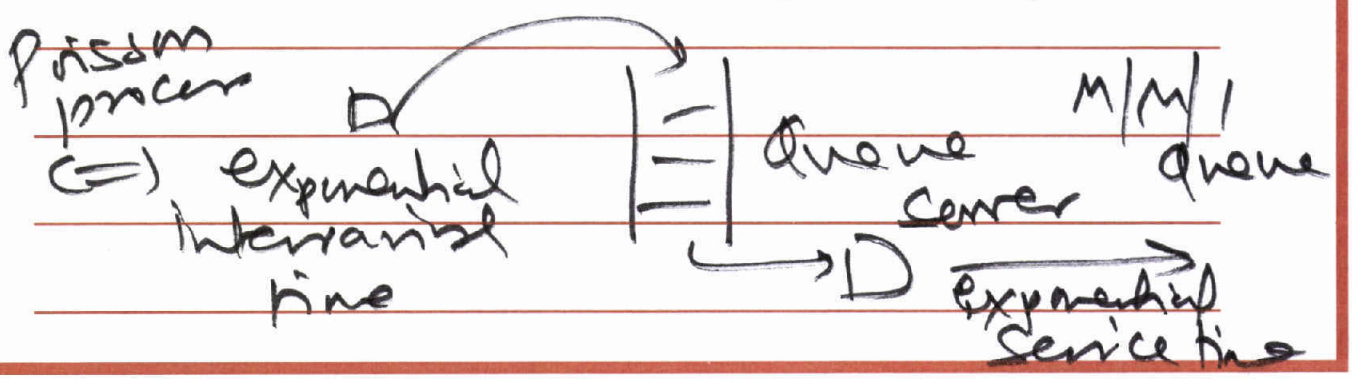
Each node i in the network maintains a set of queues, one for each flow:

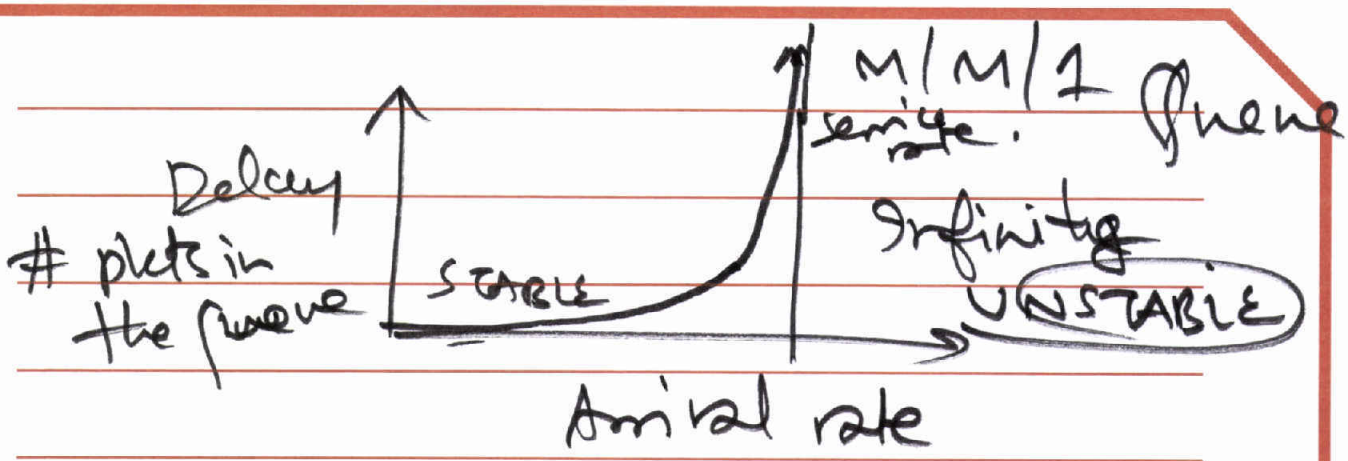
$[Q_i^1 \ldots Q_i^n]$

We assume no max-size for any queue, but want the queue occupancy to be bounded. i.e. $Q^{f(t)} \not\to \infty$

must be avoided — unstable queue

(Stability is a theoretical concept. In practice all queues are finite. Stability $\Rightarrow$ low drop rate/probability)



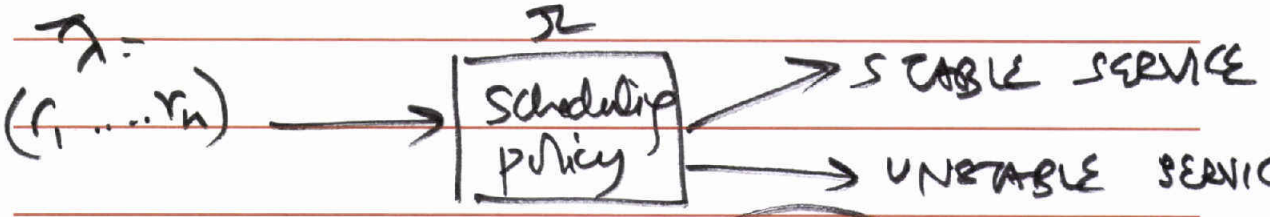


The network has an n-dimensional arrival rate & an n-dimentional service rate region.

The set of arrival rates that can be serviced without letting any queue become unstable ("blow up") is called the stability region.

Network chooses: (1) which links to schedule (subject to interference constraints) (2) which flows to schedule on each link.

A scheduling policy chooses link activation & flow-service decisions at each time.



for any $\underline{\lambda}$, if $\exists \pi$ s.t. $\underline{\lambda}$ can be serviced stably, then

$\underline{\lambda} \in \Lambda$

Λ — stability region for the network.

Tassiulas & Ephremides showed that just one scheduling policy/algorithm called Maxweight can guarantee stability $\forall \underline{\lambda} \in \Lambda$.

MAX weight policy:

$$w_{ij}^{f} = Q_i^{f} - Q_j^{f}$$

(Queue backpressure)

$$w_{ij}^{f^*} = \max_{f} w_{ij}^{f}$$

f^*

schedule links & flows such that the sum of weights: $\sum w_{ij}^{f^*}$ is maximized & interference constraints are considered.

Repeat the above at all times.

to be continued...