4/3/12 EE 597

Routing in Multihop Networks.

- ETX metric
- Anypath Routing

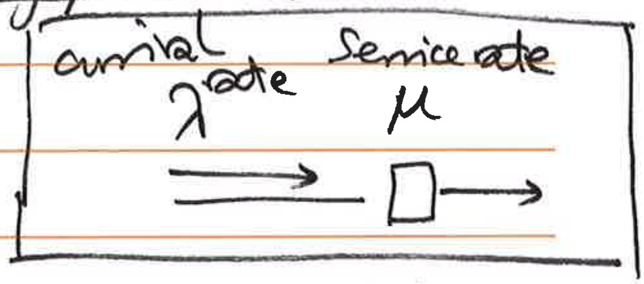
<u>Backpressure Scheduling</u>

'Dynamic Backpressure Routing
BCP- Backpressure Collection Protocol.
"Queue - Aware Routing"

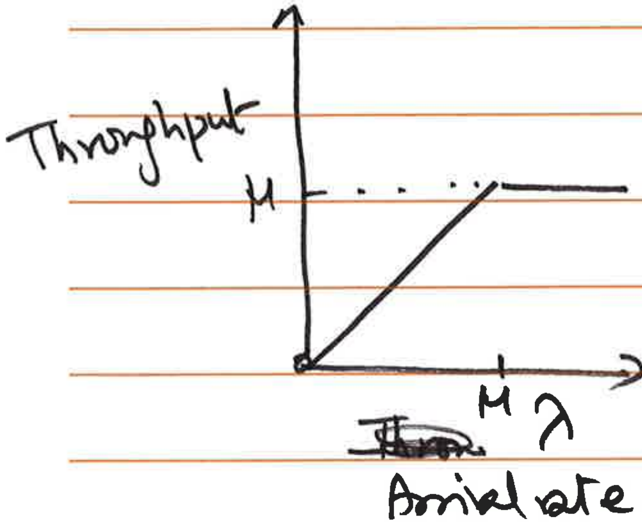
1993 Tassiulas & Ephremides
proved that "Backpressure Scheduling"
is <u>Throughput</u> Optimal.

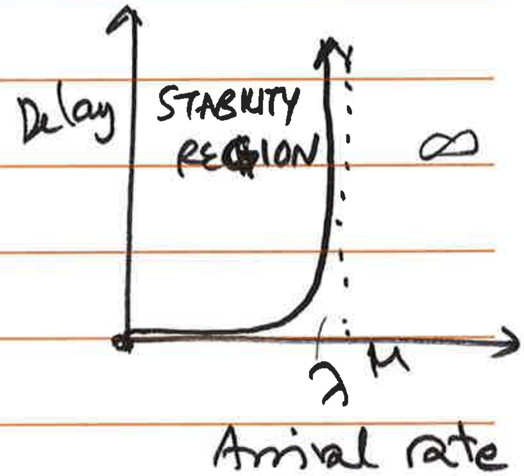
EE694

M/M/1 Queue

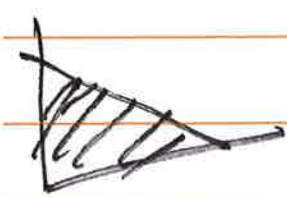


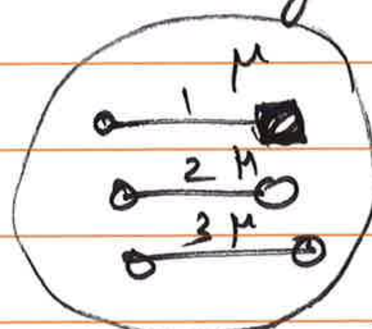
$\lambda \leq \mu$
defines the
stability region





Consider an arbitrary Network.

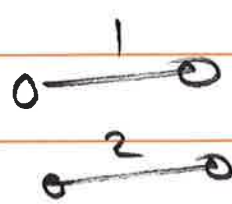




$\lambda_i \leq \mu.$

$\sum \lambda_i \leq \mu.$

Arrival rate vector $\vec{\lambda} = [\lambda_1, \lambda_2, \lambda_3]$



$\sqrt{R_1} + \sqrt{R_2} = 1$

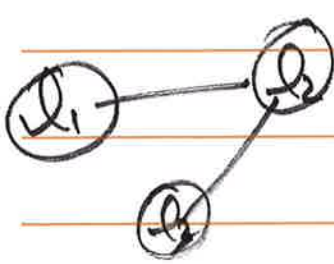
$\sqrt{\lambda_1} + \sqrt{\lambda_2} \leq \sqrt{\mu}$

commodity c: a pair of sets S_c, D_c
s.t. all pkts from sources
$s \in S_c$ must be sent to
at least one of the
destinations $d \in D_c$

e.g. Unicast flow: (s, d)
Anycast flow (s, D)
D: it's ok for any
$d \in D$ to get
s's pkts.

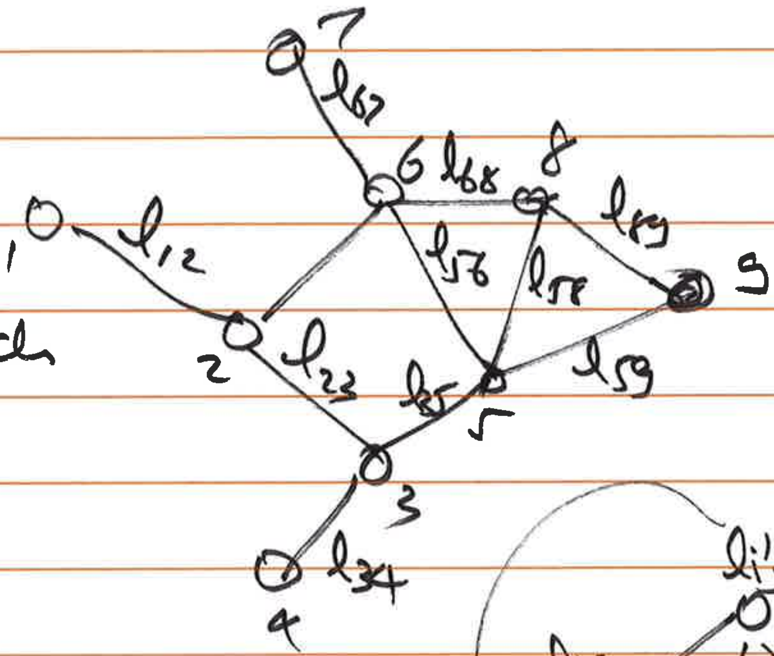
Convergecast flow: (S, d)
data from multiple sources
to get to one destination.

Network: a collection of nodes;
links; a
conflict graph to represent
interference; a set of
commodities c: 1 . . . . m;
Rates R_{ij}



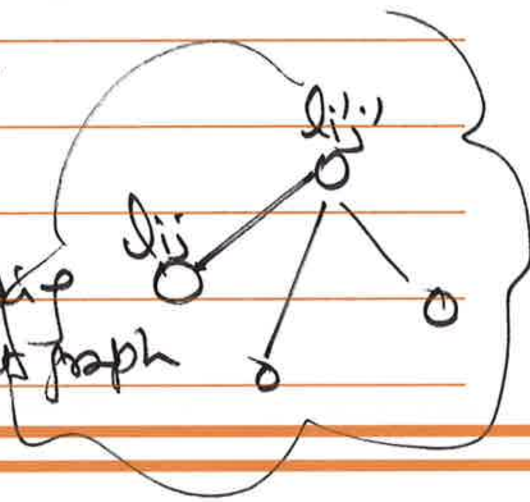
assume Buffer sizes are infinite.

max rate R_{ij} on each link.



({1}, {9})

({2}, {5, 9})



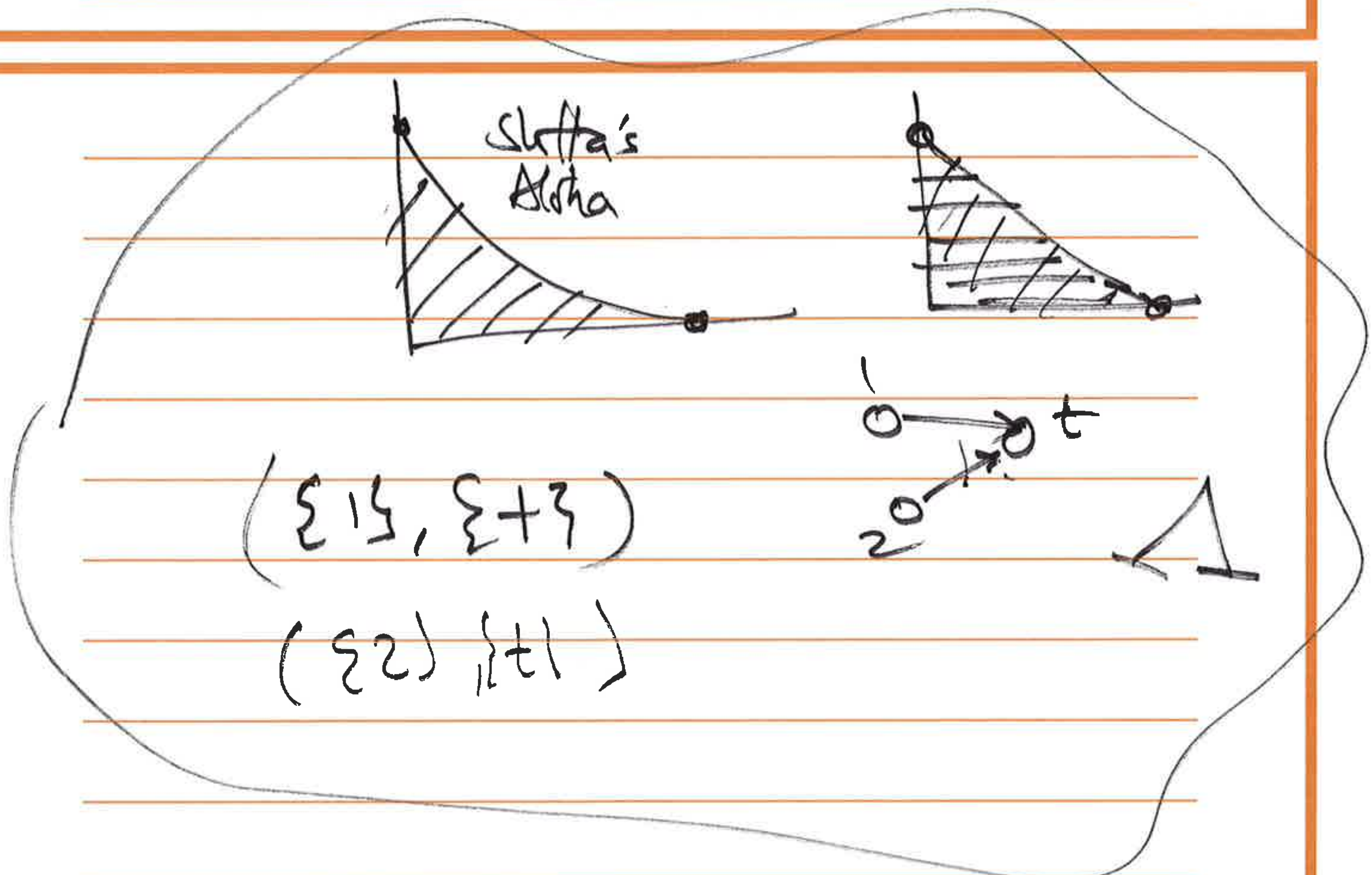
λ_c ← aggregate arrival rate for packets belonging to commodity c.

Arrival rate vector $\vec{\lambda} = \{\lambda_c\}_{c:1 \text{ to } m}$

Λ ← stability region: defined as the set of $\vec{\lambda}$, arrival rates, s.t. it is possible *somehow* to support this traffic in the network without letting any queue become unboundedly large.

i.e. In principle so long as $\vec{\lambda} \in \Lambda$, $\exists$ some way to make sure the network is stable.

A scheduling mechanism would be called "Throughput optimal" if it can stably support any $\vec{\lambda} \in \Lambda$.



time is discretized, each node maintains m queues

Max Weight Algorithm for Backpressure Scheduling

$$w_{ij}^c(t) = \left(Q_i^c(t) - Q_j^c(t) \right) \cdot R_{ij}$$

- $w_{ij}^c(t)$: weight for link ij & commodity c at time t.
- $Q_i^c(t)$: # of pkts in queue i for commodity c at time t
- $Q_j^c(t)$: " j
- R_{ij}: rate of link ij

Independent set: A set of links that can be scheduled at the same time.
(Given a conflict graph, a set of

nodes on the conflict graph that do not have any edge between them)

MaxWeight Algorithm:

At each time, schedule transmissions on links & commodities such that

a) the links form an independent set

& b) the following weight is maximized:

$$\sum_{i,j,c} w_{ij}^{c}(t)$$

Tassiulas & Ephremides showed in their 93 paper that this algorithm can stabilize any network so long as the arrival rates are in the stability region of that network.

Issues:

① The MaxWeight algorithm requires the computation of a maximum weight independent set ~~(…)~~ (MWIS).
This problem is unfortunately NP-hard!

(CS 570 / 670)

② All queue state at each node of the network needs to be known to the centralized controller at each time.

③ Also requires synchronization for scheduling across entire network.

④ ~~It~~ Analysis assumes infinite-sized queues (?)

BCP : Backpressure Collection Protocol.

Since '93 a number of researchers have exted the T-E result to allow for cross-layer utility-based network optimization, including Neely & Modiano, Stolyar, Srikant, Shroff and others

BCP 2010 Scott Moeller & others

for collection / convergecast in wireless sensor networks

~~joint~~ tries to minimize ETX cost while ensuring (stability) high throughput performance

WSN
Convergecast: all sensor are trying to get their data to a common sink (set of sinks).

· Single commodity system
∴ only 1 queue at each node.

· BCP implemented as a distributed ~~routing~~ protocol at the network layer & uses the underlying CSMA MAC (IEEE 802.15.4-based) radios

NOT a centralized scheduling solution!

BCP weight calculation:

$$w_{ij}(t) = (Q_i(t) - Q_j(t) - V \cdot ETX_{ij}) \cdot R_{ij}$$

node i computes this quantity for all nbrs j, continuously.

~~Routing &~~
Forwarding decisions:

send the next packet to the neighbor j^*: $\underset{j \in N(i)}{\arg\max}\ w_{ij}$

so long as $w_{ij^*} > 0$.

Features of BCP:

* pkt by pkt forwarding decisions
* purely local computations that are tractable

* takes into account both link & queue states.
* packets from each source may traverse multiple routes to get to sink
* no maintenance of static routing tables!