Notes from Taped Lecture [redacted] 8/1/16 (afternoon)

Last Lecture for EE597

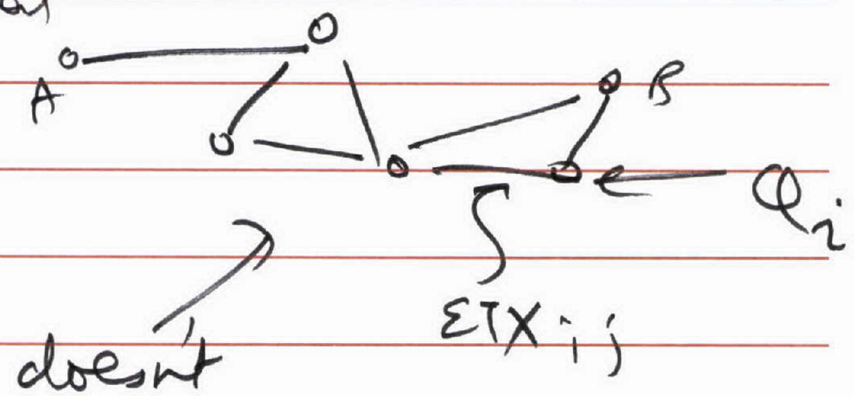
- No class on Wednesday
- Exam is on Monday 8/8 during class hours.

Intermittently connected Mobile Networks

- Wait Deliver
- Epidemic Flooding
- Spray & Wait

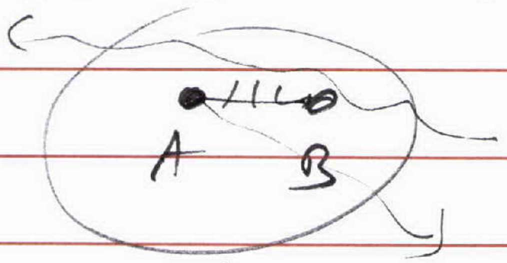
Methods to analyze ICMN protocols.

traditional static wireless networks



this doesn't work for ICMN

ETX not a useful
metric for ICMN — can
focus on high quality
links during the encounter

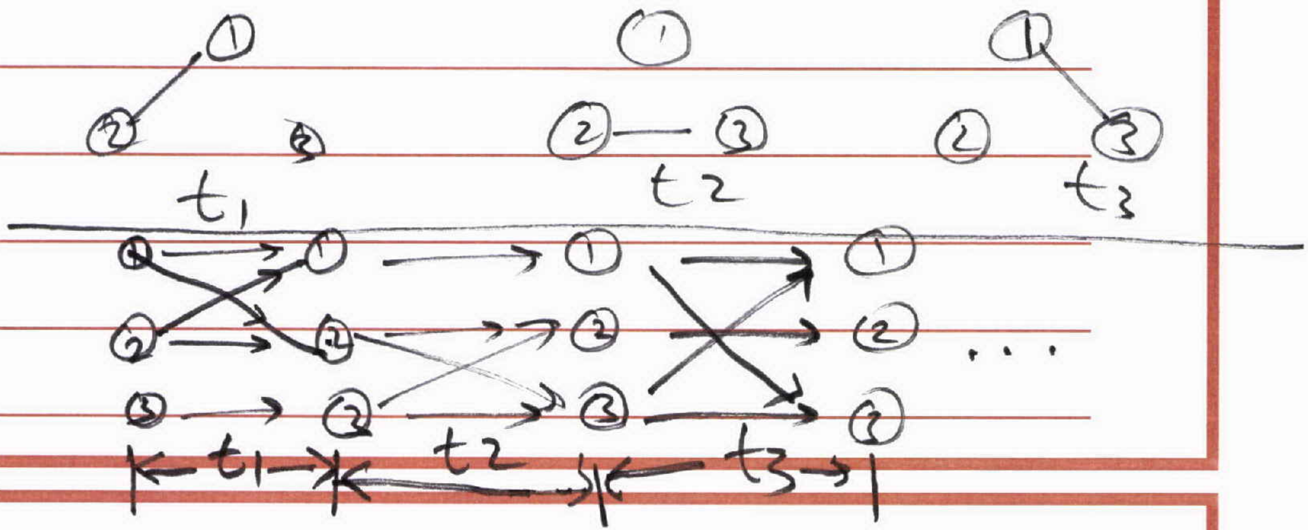


implicitly we assume encounters
are long enough to

communicate content between two
nodes in an ICMN.
(equivalently, the amount of
content is low).

In an ICMN,
latency is dominated not by
ETX, rather by the store
~~& forward~~ time between
receiving & forwarding a message.

If the mobility is predictable, can construct a trellis representation of all encounters:

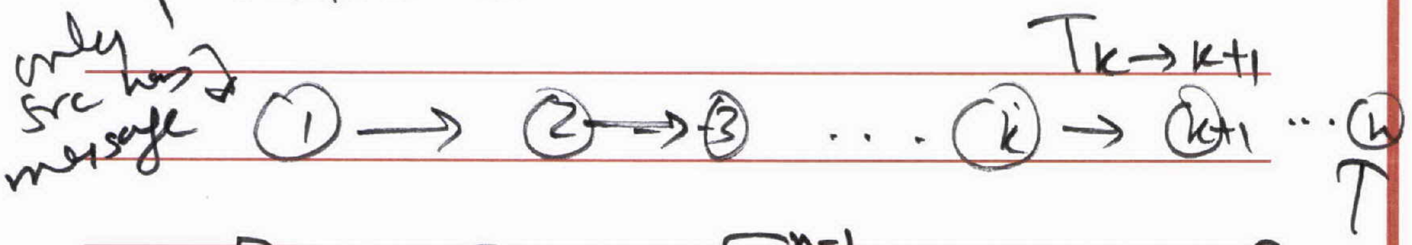


Other approach: IID encounter model (random encounters).

Illustration: flooding: how long to flood a message to all nodes?

IID model:
at each time pick at random any 2 nodes out of the $\binom{n}{2}$ possible pairs to encounter each other

Basic idea: # of "infected" nodes is the state of the network

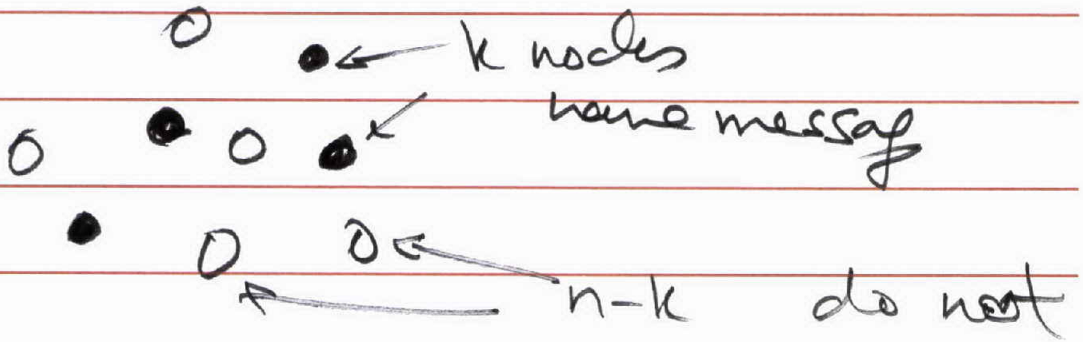


everyone has message

$$E\left[T_{flood}\right] = E\left[\sum_{k=1}^{n-1} T_{k\to k+1}\right]$$

$$= \sum_{k=1}^{n-1} E\left[T_{k\to k+1}\right].$$

can model this as expected time to success for a geometric r.v.



Prob. of success = Prob. of picking one node w/ data & one w/o.

$$P_{k\to k+1} = \frac{k\cdot(n-k)}{\binom{n}{2}}$$

$$\boxed{\frac{k\cdot(n-k)}{\frac{n\cdot(n-1)}{2}}}$$

$$E\left[T_{k\to k+1}\right] = \frac{1}{P_{k\to k+1}} = \frac{\binom{n}{2}}{(k)(n-k)}$$

$$E\left[T_f\right] = \sum_{k=1}^{n-1}\frac{\binom{n}{2}}{(k)(n-k)}$$

$$= \binom{n}{2}\sum_{k=1}^{n-1}\frac{1}{(k)(n-k)}$$

$$\frac{1}{k} + \frac{1}{n-k} = \frac{n-\cancel{k}+\cancel{k}}{(k)(n-k)}$$

$$\frac{1}{n}\left(\frac{1}{k}+\frac{1}{n-k}\right) = \boxed{\frac{1}{(k)\cdot(n-k)}}$$

$$E[T_f] = \binom{n}{2} \sum_{k=1}^{n-1} \frac{1}{n} \cdot \left(\frac{1}{k} + \frac{1}{n-k} \right)$$

$$\frac{\cancel{n} \cdot n-1}{2 \cdot \cancel{n}} \sum_{k=1}^{n-1} \left(\frac{1}{k} + \frac{1}{n-k} \right)$$

$$= \frac{n-1}{2} \cdot \left[\sum_{k=1}^{n-1} \frac{1}{k} + \sum_{k=1}^{n-1} \frac{1}{n-k} \right]$$

$$\frac{1}{1} + \frac{1}{2} + \dots \frac{1}{n-1} = \frac{1}{n-1} + \frac{1}{n-2} \dots \frac{1}{1}$$

$$= \frac{n-1}{\cancel{2}} \cdot \cancel{2} \boxed{\sum_{k=1}^{n-1} \frac{1}{k}} \sim H(n)$$

$$\sim \log(n) + c$$

$$E[T_{flood}] \sim O(n \cdot \log n) \text{ encounters}$$

↖ big-Oh notation

in order, it grows as $\boxed{n \cdot \log n}$

In practice, # of encounters per unit time will increase with the number of nodes.

e.g. if each pair has a constant rate of encounter, then # of encounters grows as $O(n^2)$.
If this is true (e.g. fixed area).

Then time to flood will scale as:

$$\underset{\text{to flood}}{\#\text{ of encounters}} \times \frac{\text{seconds}}{\text{encounter}}$$

$$\simeq n \cdot \log n \times \frac{1}{n^2}$$

$$\propto \frac{\log n}{n}$$

as $n \to \infty$ $E[T_{flood}] \to 0$!

i.e. flooding time decreases w/ population size in a fixed area

In a wireless link, the losses due to fading are not affected by send-rate & if treated the same as congestion, will result in very low throughput.

Need to separate losses due to fading from losses due

to buffer overflow/congestion.

<u>Solution</u>: use explicit congestion notification: intermediate routers mark pkts when congested. Src can then differentiate congestion from channel loss & only cut data rate due to congestion events.

In the short term this is bad for flow f_2.

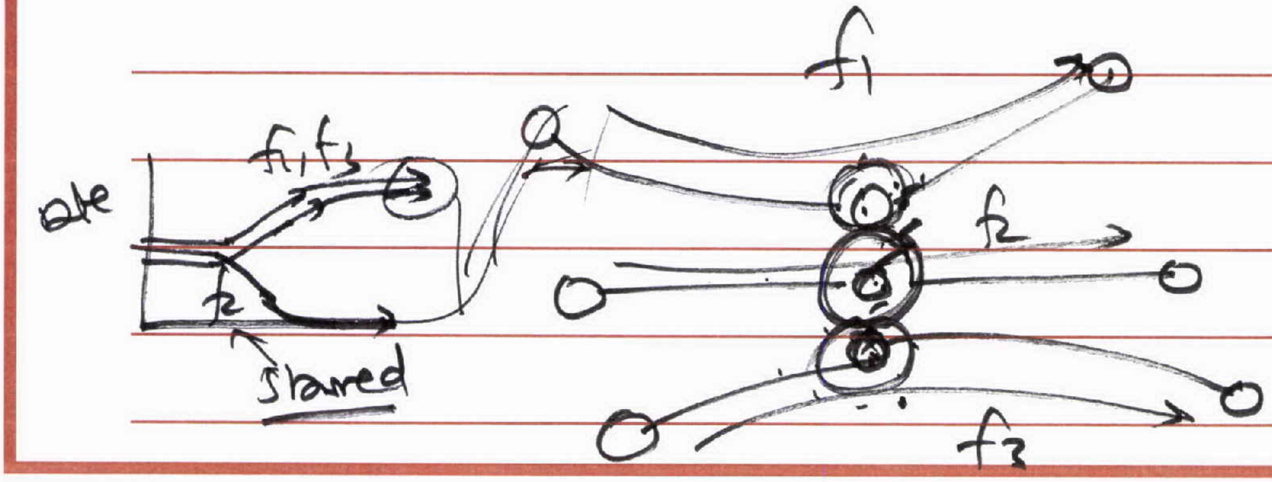
On the other hand, while f_2 is starved, f_1 & f_3 get more capacity than they would have & can finish quicker, allowing f_2 to then start later & finish.

In the long run, total time to complete all flows might actually be shorter w/ TCP in wireless setting. → not always of interest.

This recovers fairness.

(will post some pointers to papers addressing TCP fairness)

2nd challenge: fairness & flow starvation for flows-in-the-middle.



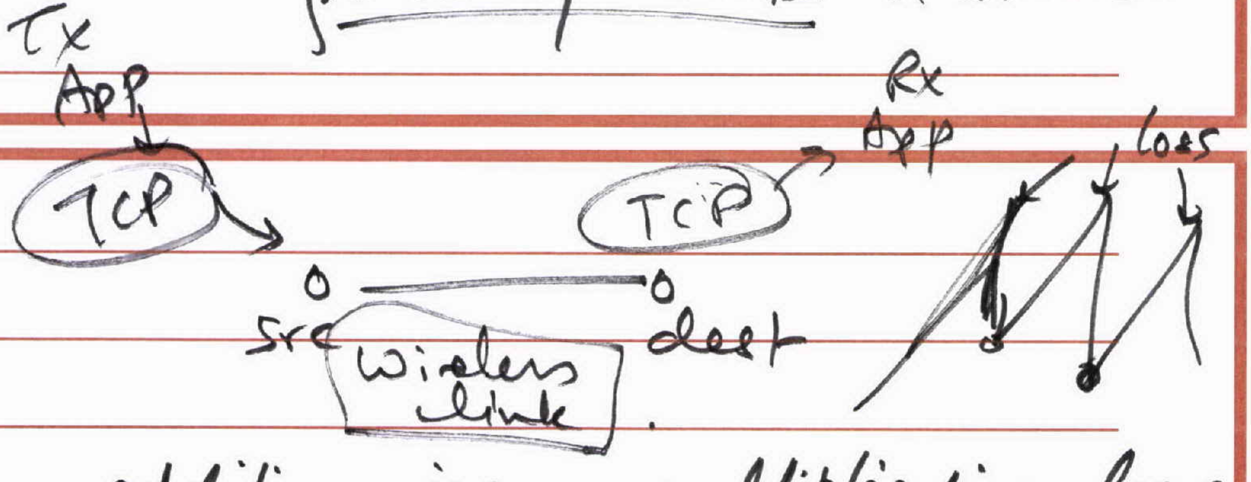
f_2's relay router sees the most interference, lowest capacity (e.g. due to higher backoffs when using CSMA), tends to cut back its rate more often, causing outer flows to increase their rates, in turn causing more congestion for middle flow... this escalates until f_2 has ~0 rate.

Transport Layer: Is TCP good enough?

UDP ← no change needed (best effort anyway)

TCP: two challenges:

rate control over dynamic quality links in wireless



additive inc, multiplicative decrease of the congestion window / rate

decrease is triggered by pkt loss causing timeouts [illegible] when awaiting the end to end ack

Fine if losses are primarily due to congestion

Key idea in solutions to
TCP unfairness:

f2 suffers congestion form
f1 & f3 but only f2
cuts back its rate.

if f1 & f3 could both also
cut back when f2 cuts back,
it would improve fairness.

---

e.g. WCP implements this idea.

TCP challenges in multi-hop wireless

1. Loss differentiation
→ solved using ECN pkts

2. Fairness (short term)
→ solved using shared
congestion notification