Routing
Network = $G(V,E)$

Routing: Find "optimal" path from node A to B.

Classical problem in Internet. 2 basic ideas

- Link State Routing (Dijkstra's)
- Distance Vector Routing (Bellman Ford)

Core idea:

- Recruit node $w$ not in already recruited set $S$ with minimum cost
- Include node $w$ in $S$
- Update costs to all nodes $v$ adjacent to $w$

Cost ($v$) = min \{ \text{cost}(v), \text{cost}(w) + \text{cost}(w,v) \}

$d_x(y) = \min \{ \text{cost}(x,v) + d_v(y) \}$

- No need of global view of network.
Why not apply to wireless.

* Channel is "Broadcast" medium.
  Take advantage of it.

* Channel errors much more than wired.
  Cope with it.

Question 1: How to compute path cost?

Internet:

\[
\text{when } C_i = \text{Delay, then } \\
\text{Cost}(A \rightarrow D) = \sum_i C_i
\]

\[
\text{when } C_i = \text{Congestion} \\
\text{Cost}(A \rightarrow D) = \max(C_i)
\]

Choose a path with minimum bottleneck congestion.

When \( C_i = \text{Delivery ratio}, \) then what?

Which path better?
If you pick $A \to B \to D$

Throughput = 33%  

$A \to C \to D$ : Throughput = 25%

But then, say

$A \to B \to D$ Throughput = 33%

$A \to D$ Throughput = 50%

So you cannot add up delivery ratios.

Proposed metric (ETX)

Minimize expected transmission count

\[
\text{link } \text{ETX} = \frac{1}{P(\text{packet success})}
\]

\[
\text{link throughput} \approx \frac{1}{\text{link } \text{ETX}}
\]

\[
\text{Route } \text{ETX} = \sum_{\text{Li} \in \text{R}} \text{ETX}(\text{Link Li})
\]
So we took care of high link errors. But didn't take advantage of wireless broadcast nature of channel.

Exploit the fact that some intermediate node (close to the destination) may have received the source's broadcast.

Utilize this node as the relay (or route).

i.e., Pick your route after transmission as opposed to choosing beforehand.

Example:
Static route ETX
\[
= \frac{1}{0.25} + \frac{1}{1} = 5
\]

But Exor ETX:
\[
= \frac{1}{P(\text{at least one pkt makes it to B, C, or E})} + \frac{1}{1}
\]
\[
= \frac{1}{1 - (1 - 0.25)^4} + \frac{1}{1} = 0.25
\]

Much better

**Big Question:**
How do you make the node closest to dest to transmit first (followed by next closest and so on)?

**Solution:**
- Tx sends batches of packets, say \([1, 100]\)
- Each node receives different subsets
- Closest node (i.e., best ETX to dest?) forwards first ... all packets it has received
- Also sends summary of all received packets as headers in each packet
- Those who overhear summaries do union of summaries with its own packets & inserts in packet header
- Source gets final summary & retransmits all packets not in summary
- Also nodes periodically their own ETX values to all other nodes
Of course, many issues need investigation.

1. What about data rates? If links transmit at best data rates, further away links won't overhear.

2. What about TCP issues.

3. What about overhead of summaries in EVERY packet header.

4. What about real-time applications (since packets will come in batches).