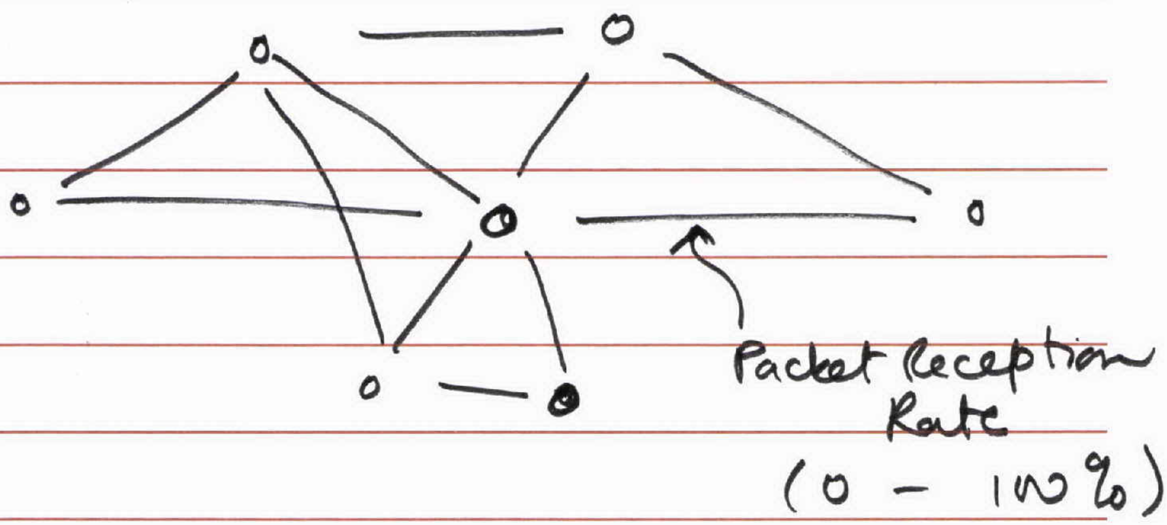
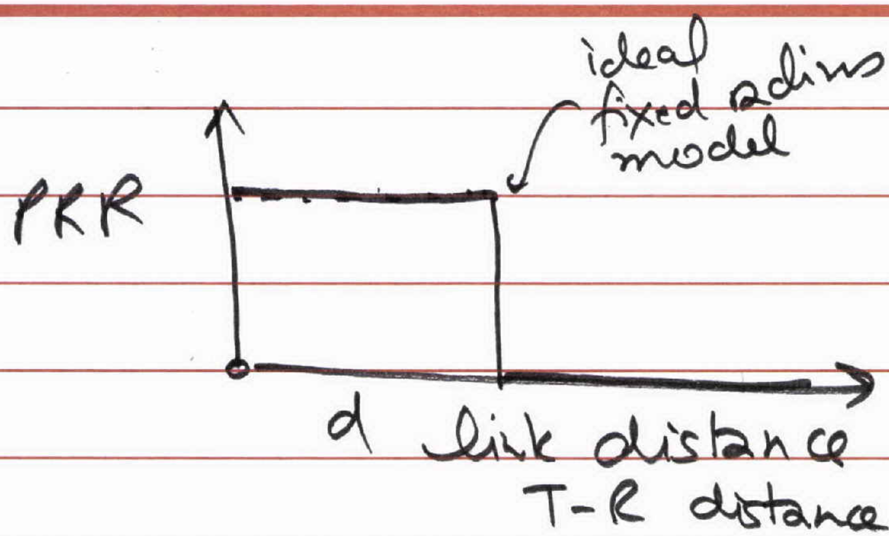
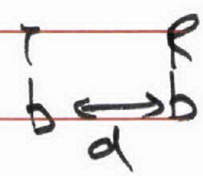
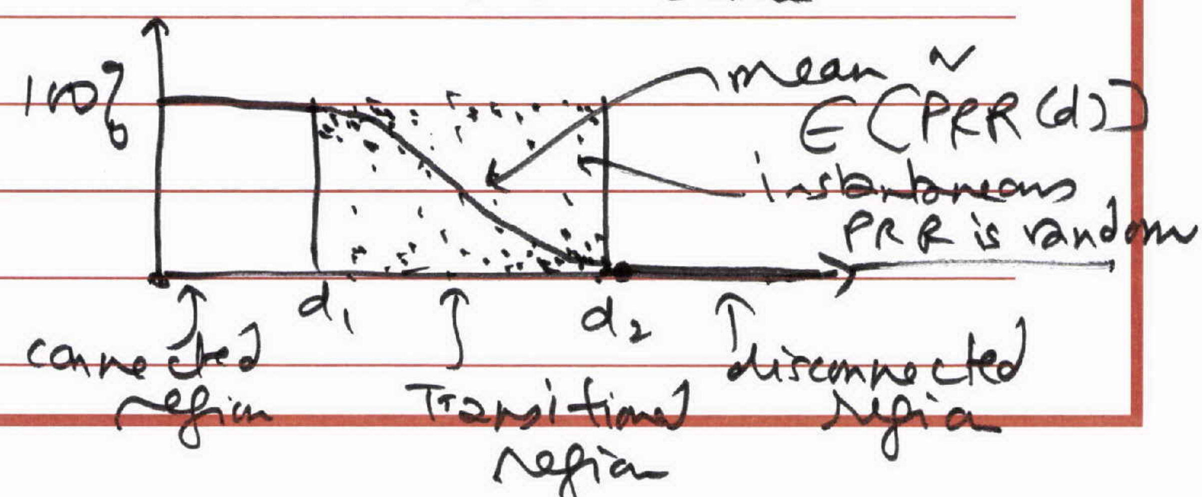
EE [redacted] 597 7/20/2016

Network Layer

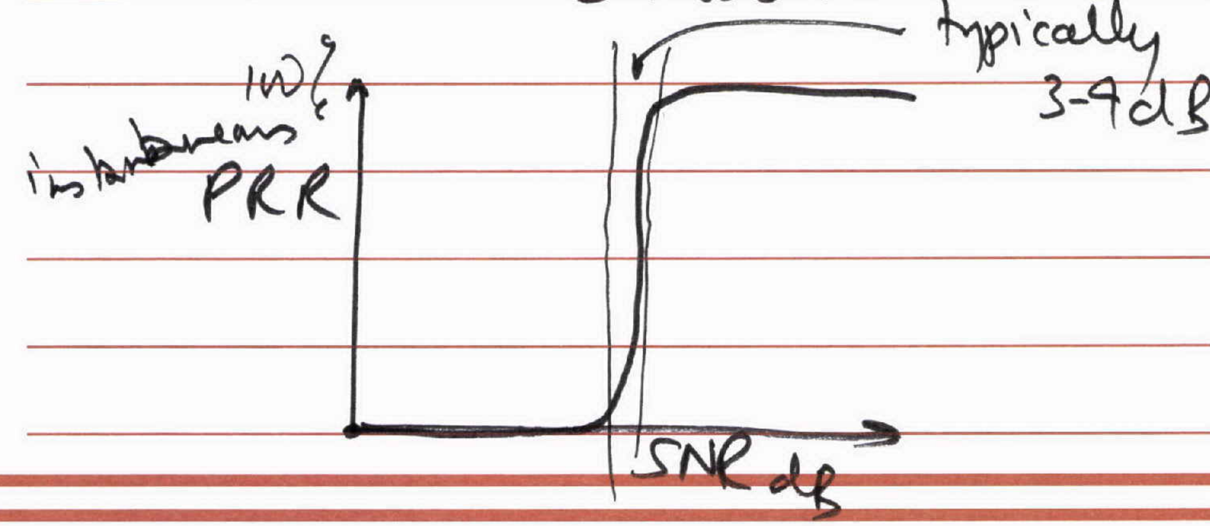


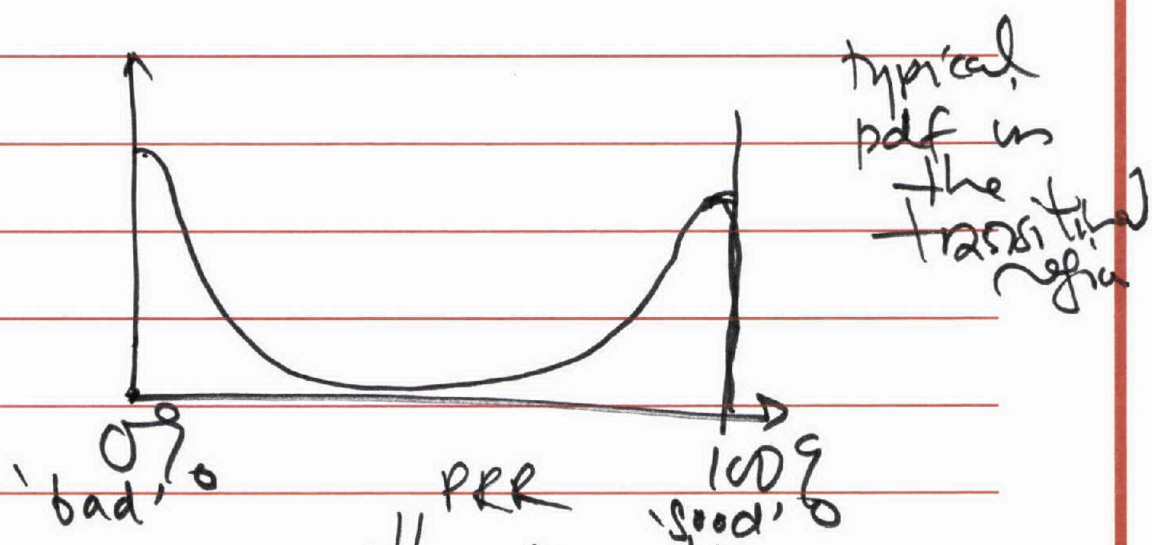




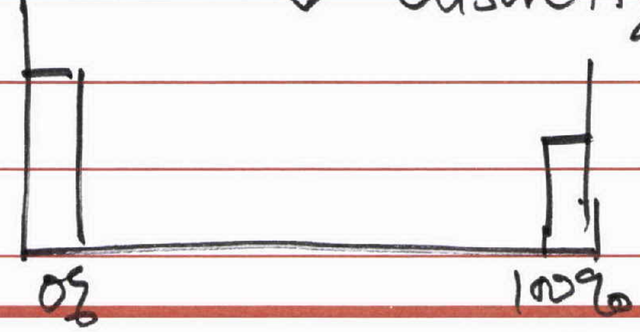


Typical distribution of PRR at a given distance in the transitional region is Bimodal.

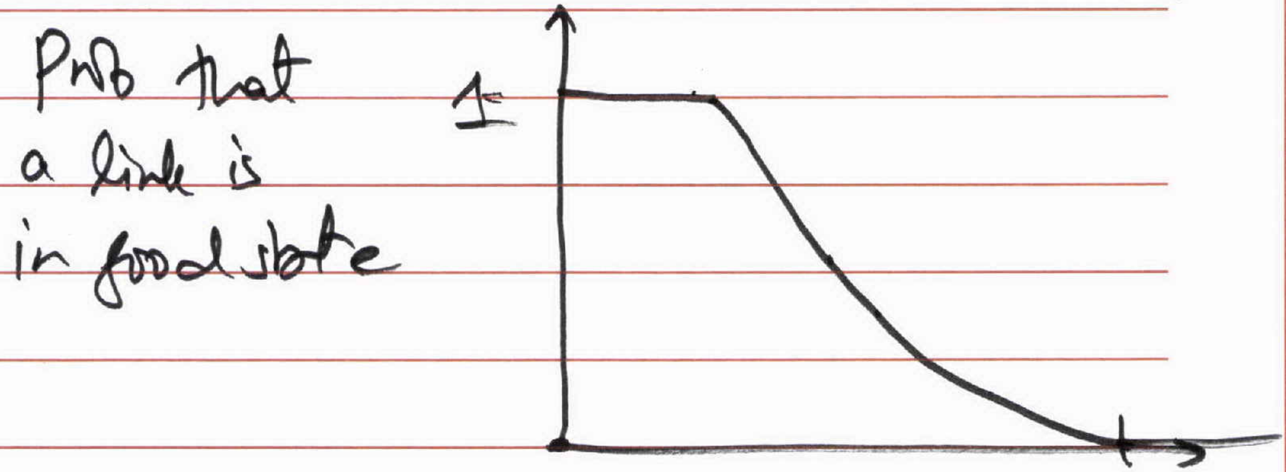




⇓ discretize version



An idealized 2-state model of link quality.



Temporal variation may be modeled via a

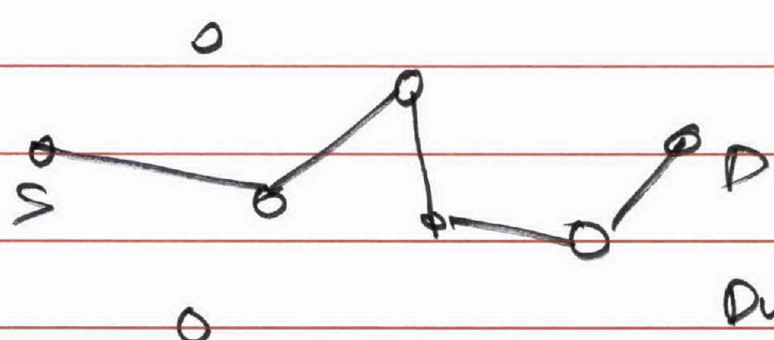
Markov model

Blacklisting: avoid using links w/ $\boxed{PRR < \theta}$

ETX - A metric for use in routing

Simplest Approach to routing:
use Link State / Distance vector routing to compute shortest paths from end to end, using ETX as the metric.

This produces paths such that $\sum_{ij \in P} ETX_{ij}$ is minimized.



Two innovations in Wireless Routing: Dynamic Routing Schemes

Diversity → 1. Anypath Routing / Opportunistic Routing

Queue-Aware Routing → 2. Backpressure Routing

In this scheme, the probability of success at the first hop =
prob. that either A or B or both get the pkt
= 1 − prob. that neither gets it
if independent
$$= 1 - (1 - P_{SA}) \cdot (1 - P_{SB})$$

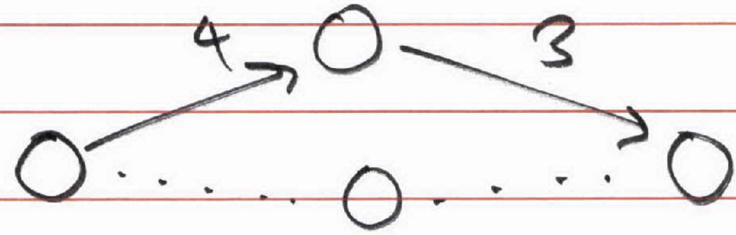
$$= 1 - \left(1 - \frac{1}{4}\right) \cdot \left(1 - \frac{1}{5}\right)$$

$$= 1 - \frac{3}{4} \cdot \frac{4}{5} = 1 - \frac{3}{5} = \frac{2}{5}$$

The total ETX of the Anypath Scheme

$$= \frac{5}{2} + 3 = 5.5$$

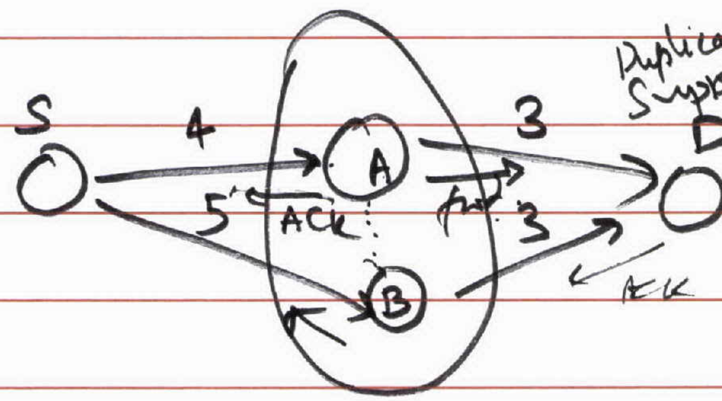
Single path min-ETX solution:



ETX = 7

Duplicate suppression

Anypath Solution



ETX = ?

5.5

Simplifying assumption: $ETX = \frac{1}{PRR}$

$\therefore$ $PRR_{S \to A}$: $\frac{1}{4}$ (Prob. of success)

$PRR_{S \to B}$: $\frac{1}{5}$

$PRR_{A \to D}$: $\frac{1}{3}$

$PRR_{B \to D}$: $\frac{1}{3}$

A	B	
X	X	
X	1	B fwd
1	X	A fwd
1	1	

If neither A or B get it, no Ack, retransmit the pkt

If only A or only B get it, only they fwd.

If A & B both receive only A fwds.

How to coordinate nodes A & B?

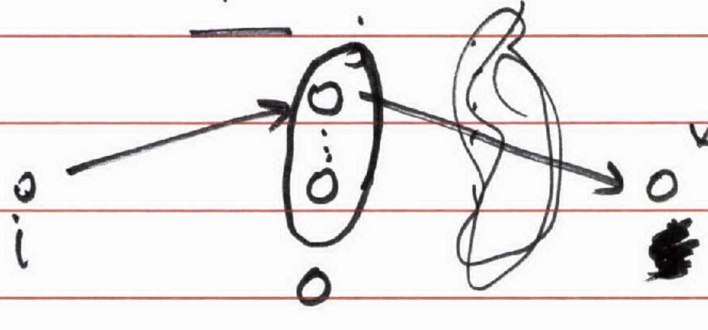
In particular, when both receive the pkt how to ensure that only A fwds the pkt.

Naive implementation would make B forward all the time regardless of whether A gets the message.

—

have separate timers / latencies associated w/ each forwarder as a function of their priority.

Can generalize Dijkstra's algorithm to Anypath Routing.



destination node.

total cost $D_i = d_{ij} + D_j$

local cost to nbr j

cost to go from nbr j

say the forwarding set be called J

$$D_i = d_{iJ} + D_J$$

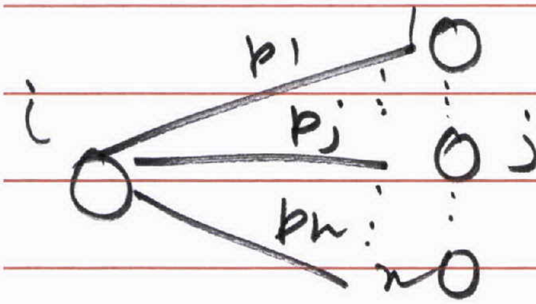
local cost to fwd set J

cost to go from set J

say $J = \{1, 2 \dots n\}$ ← n nodes

w/ success probabilities $p_{i1}, p_{i2} \dots p_{in}$

for simplicity $p_1, p_2 \dots p_n$



p_{iJ} ← success prob. to get pkt at at least one node in J.

$$d_{iJ} = \frac{1}{p_{iJ}}$$

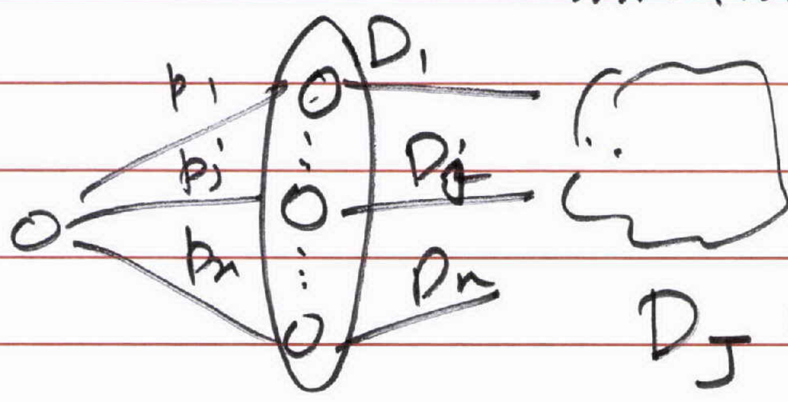
(assuming iid success)

$$p_{iJ} = 1 - \prod_{j=1}^{n} (1 - p_j)$$

← assuming independent link success

$$\therefore d_{iJ} = \frac{1}{1 - \prod_{j=1}^{n} (1 - p_j)}$$

D_J ← expected cost to go from set J to destination.

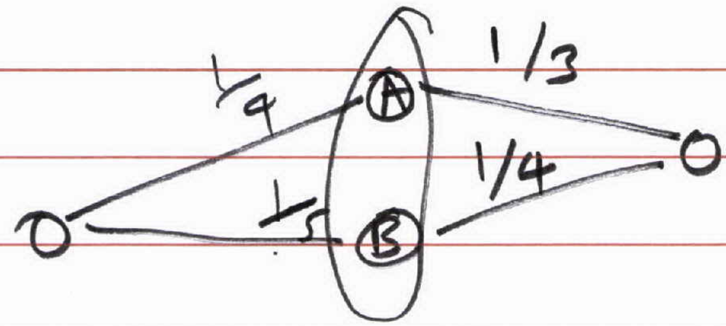


O destination

$$D_J = \sum_{j=1}^{n} \omega_j \cdot D_j$$

↑ prob. that node j is the forwarder

$$w_B = \frac{Pr(B \text{ got it}) \cdot Pr(A \text{ did not got it})}{Pr(A \text{ or } B \text{ got it})}$$



$$w_A = \frac{Pr(A \text{ gets it})}{Pr(A \text{ or } B \text{ get it})}$$

$$Pr(X \mid Y) = \frac{Pr(X \;\&\; Y)}{Pr(Y)}$$

$Pr(A \text{ is a forwarder})$

$$= Pr(A \text{ got it} \mid A \text{ or } B \text{ or both got it})$$

$$= \frac{Pr(A \text{ gets it} \;\&\; A \text{ or } B \text{ or both got it})}{Pr(A \text{ or } B \text{ or both got it})}$$

$$= \frac{Pr(A \text{ gets it})}{Pr(A \text{ or } B \text{ or both})}$$

$$\omega_j = \frac{p_j \prod_{k=1}^{j-1} (1 - p_k)}{1 - \prod_{j=1}^{n} (1 - p_j)}$$

general expression for probability that node $j \in J = \{1 \ldots n\}$ is the forwarder of a pkt from node i given Pr success $i \to j = p_j$

$$D_i = d_{iJ} + D_J$$

$$= \frac{1}{p_{iJ}} + \sum_{j=1}^{n} \omega_j D_j$$

$$D_i = \frac{1}{1 - \prod_{j=1}^{n} (1 - p_j)} + \sum \frac{p_j \prod_{k=1}^{j-1} (1 - p_k)}{1 - \prod_{j=1} (1 - p_j)} \cdot D_j$$

given a forwarding set J, & $D_j \; \forall j \in J$

& $p_{ij} = p_j \; \forall j \in J$

computes D_i

$G=(V,E)$ network graph, dest. node

Shortest Anypath First Algorithm (G, d)

for each node $i \in V$ //initialization

do $D_i \leftarrow \infty$, $F_i \leftarrow \phi$

$D_d \leftarrow 0$ $S \leftarrow \phi$ $Q \leftarrow V$ // ϕ: empty set.

while $Q \neq \phi$

{ do $j \leftarrow$ extract-Min(Q) // pulls out vertex w/ smallest D_i

$S \leftarrow S \cup \{j\}$

for each incoming-edge (i, j) in E

do $J \leftarrow F_i \cup \{j\}$

if $D_i > D_j$

then

$D_i \leftarrow d_{iJ} + D_J$

$F_i \leftarrow J$