Routing Algorithms: Dealing With Failures

6.02 Fall 2013 Lecture 21

INTRODUCTION TO EECS II
DIGITAL COMMUNICATION SYSTEMS
Today’s Plan

- Network Dynamics and Failures
- Analysis of Dijkstra Algorithm (Static/Ideal)
- Consequences of Incomplete LSA Flooding
- Dealing With Loopy Forwarding
- Analysis of Distance-Vector Algorithms (Static/Ideal)
- Count-To-Infinity
- Split-Horizon and Path-Vector Routing
Network Dynamics And Failures

Link Failure:

\[ \text{Probabilistic Nature of Packet Delivery:} \]

\[ P(\text{delivery}) = p \]
Dijkstra’s Shortest Path Algorithm: Complexity

Parameters:

- \( N \): number of nodes
- \( L \): number of links

Complexity:

- **finding \( u \) (minimal cost node):**
  - \( N \) times: \( O(\log N) \) each time, total \( O(N \log N) \)
- **updating costs:**
  - \( O(L) \), since each link appears twice
Correctness of Shortest Path Routing (Ideal)

Shortest Path Routing:
forward to the neighbor with minimal total cost to destination

Distance to destination decreases monotonically at every step.
Consequences of Incomplete LSA Flooding

Node G does not get the LSA from A and E
Node D does not get the LSA from A and B
Dealing With Forwarding Loops

Eventually, in the LSA framework, we expect all nodes to have complete info about the network.

In the meanwhile, hop limit eliminates infinite loops:

- Passing every node, a package reduces its hop limit by 1
- A package with hop limit of zero is dropped
Distance-Vector Algorithms

\begin{align*}
D^+(A, X) &= \min_{B \text{ is neighbor}(A)} \{ C(A, B) + D(B, X) \} \\
R(A, X) &= \arg \min_{B \text{ is neighbor}(A)} \{ C(A, B) + D(B, X) \}
\end{align*}
Counting To Infinity

A ghost of C is lurking in the links...
The "Split-Horizon" Fix

Do not advertise (advertize $\infty$) down the forward route

![Diagram showing network with nodes A, B, and C, and routes L1 and L2.]

- A to B: L1 (cost=1)
- B to C: L2 (cost=1)
- C to A: L1 (cost=2)
- C to B: C: 1, L2
- C to C: $\infty$, None

...
The "Split-Horizon" Fix Does Not Always Work

When Advertisements Are Not In Sync

```
A -- L1 -- B -- L2 -- C x
```

- A: D: 2,L3
- B: D: 2,L2
- C: D: 1,L4
- D: D: 2,L3
- D: 3,L1
- D: inf,None
- D: 4,L2
The "Path-Vector" Fix

- Report not just the first forward, but the whole path
- Do not advertise (advertize $\infty$) down the path

![Diagram of network with nodes A, B, C, D and edges L1, L2, L3 with path vectors]

- D: 2, L3, L4
- D: 2, L2, L4
- D: 1, L4
- D: 2, L3, L4
- D: 2, L2, L4
- D: inf, None
- D: 3, L1, L3, L4
- D: inf, None
- D: inf, None
- D: inf, None
- ...
The network layer implements the glue that achieves connectivity.

Does addressing, forwarding, and routing.

Forwarding entails a routing table lookup; the table is built using routing protocol.

DV protocol: distributes route computation; each node advertises its best routes to neighbors.

Path-vector: include path, not just cost, in advertisement to avoid count-to-infinity.