Routing – the big picture

- Internet divided into Autonomous Systems (ASes)
  - corresponds to an administrative domain
  - examples: University, company, backbone network
  - assign each AS a 16-bit number
- Two-level route propagation hierarchy
  - interior gateway protocol (RIP, OSPF)
  - exterior gateway protocol (Internet-wide standard)

Overview

- Forwarding vs Routing
  - forwarding: to select an output port based on destination address and routing table
  - routing: process by which routing table is built
- Network as a Graph
- Problem: Find best path between two nodes
- Factors
  - static: topology
  - dynamic: load
Families of routing algorithms

- Distance vector
  - Tell your neighbors about everybody you know of
  - Lower memory
  - RIP: Route Information Protocol
    - based on hop-count
- Link state
  - Tell everybody about your neighbors
  - Most used today
  - OSPF: Open Shortest Path First

Distance Vector

- Each node maintains a set of triples
  - \((\text{Destination, Cost, NextHop})\)
- Neighbors exchange updates
  - periodically (on the order of several seconds)
  - whenever table changes (called triggered update)
- Each update is a list of pairs: \((\text{Dest, Cost})\)
- Update local table if receive a “better” route
  - smaller cost
  - came from next-hop
- Refresh existing routes; delete if they time out

Example

Find the 4 bugs in this picture!!!
Link failure example

Count to infinity example

Count to infinity example
Loop-Breaking Heuristics

- Set infinity to 16
- Split horizon
  - Don’t advertise route to neighbor you heard it from
- Split horizon with poison reverse
  - Advertise it with $\infty$ cost
Link State

- **Strategy**
  - send to all nodes (not just neighbors) information about directly connected links (not entire routing table)
- **Link State Packet (LSP)**
  - id of the node that created the LSP
  - cost of link to each immediate neighbor
  - sequence number (SEQNO)
  - time-to-live (TTL) for this packet

--

Link State (cont)

- **Reliable flooding**
  - store most recent LSP from each node
  - forward new LSPs to all neighbors (except the one that sent it)
  - generate new LSP periodically
    - increment SEQNO
  - start SEQNO at 0 when reboot
  - decrement TTL of each stored LSP
    - discard when TTL=0

--

Route Calculation

- **Dijkstra’s shortest path algorithm**
  - $s$ denotes node performing calculation
  - $(i,j)$ denotes non-negative cost (weight) for edge $(i,j)$
  - $C(n)$ denotes cost of the path from $s$ to node $n$
  - $N$ denotes set of all nodes in the graph
  - $M$ denotes set of nodes incorporated so far
  - $M = \{s\}$
    - for each $n$ in $N - \{s\}$
      - $C(n) = l(s,n)$
    - while $|M| < |N|$
      - $w(\{w\})$ such that $C(w)$ is the min for all $w$ in $N - M$
      - for each $n$ in $N - M$
        - $C(n) = \min(C(n), C(w) + l(w,n))$
  - **Invariant of Dijkstra’s algorithm**
    - We have shortest path for nodes from $M$ to $s$
    - For nodes outside $M$ we have shortest path that goes to $s$ only using nodes in $M$ as next hop