CSE/EE 461 Lecture 7
Routing

Tom Anderson
tom@cs.washington.edu
Peterson, Chapter 4.2

Routing Alternatives

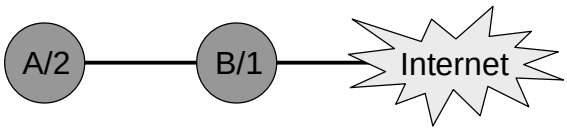
- Spanning Tree (Ethernet)
  - Convert graph into a tree; route only along tree
- *Distance vector (RIP, BGP)*
  - *exchange routing tables with neighbors*
  - *no one knows complete topology*
- *Link state (OSPF)*
  - *send everyone your neighbors*
  - *everyone computes shortest path*

Distance Vector Routing

- Each router periodically exchanges messages with neighbors
  - best known distance to each destination (“distance vector”)
- Initially, can get to self with 0 cost
- On receipt of update from neighbor, for each destination
  - switch forwarding tables to neighbor if it has cheaper route
  - update best known distance
  - tell neighbors of any changes
- Absent topology changes, will converge to shortest path

Count To Infinity Problem

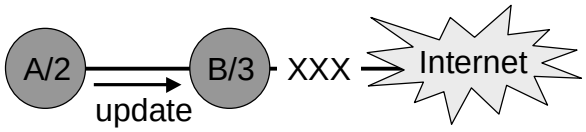
- Simple example
  - Costs in nodes are to reach Internet



- Now link between B and Internet fails …

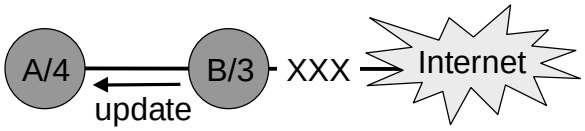
Count To Infinity Problem

- B hears of a route to the Internet via A with cost 2
- So B switches to the “better” (but wrong!) route



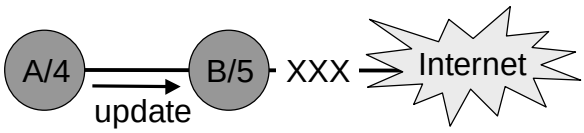
Count To Infinity Problem

- A hears from B and increases its cost



Count To Infinity Problem

- B hears from A and (surprise) increases its cost
- Cycle continues and we “count to infinity”



- Packets caught in the crossfire loop between A and B

Solutions

- Split horizon
  - Router never advertises the cost of a destination back to its next hop – that’s where it learned it from!
  - Solves trivial count-to-infinity problem
- Poison reverse (RIP)
  - go farther: advertise infinity back to source
  - vulnerable to more complex topology changes
- Path vector (BGP)
  - announce entire path to each destination
  - easy to check for loops

Routing Information Protocol (RIP)

- DV protocol with hop count as metric
  - Infinity value is 16 hops; limits network size
  - Includes split horizon with poison reverse
- Routers send vectors every 30 seconds
  - With triggered updates for link failures
  - Time-out in 180 seconds to detect failures
- RIPv1 specified in RFC1058
  - www.ietf.org/rfc/rfc1058.txt
- RIPv2 (adds authentication etc.) in RFC1388
  - www.ietf.org/rfc/rfc1388.txt

Link State

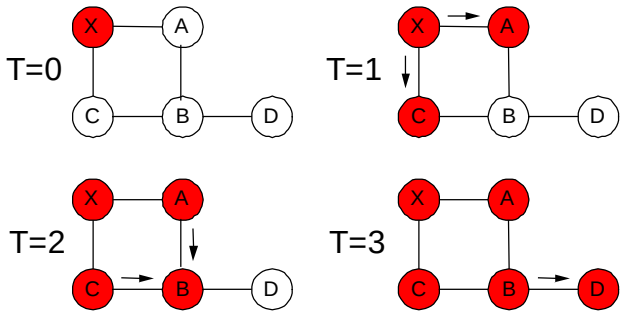
- Every router learns complete topology and then runs shortest-path
- Two phases:
  - Topology dissemination -- each node gets complete topology via reliable flooding
  - Shortest-path calculation (Dijkstra's algorithm)
- As long as every router uses the same information, will reach consistent tables

Flooding

- Each router identifies direct neighbors; put in numbered link state packets (LSPs) and periodically send to neighbors
  - LSPs contain [router, neighbors, costs]
- If get a link state packet from neighbor Q
  - drop if seen before
  - else add to database and forward everywhere but Q
- Each LSP will travel over the same link at most once in each direction

Example

- LSP generated by X at T=0
- Nodes become red as they receive it



Complications

- What happens when a link is added or fails?
  - LSPs are numbered; only forward LSP if its new
  - Use cost infinity to signal a link is down
- What happens when a router fails and restarts?
  - How do the other nodes know it has failed?
  - What sequence number should it use?
- What happens if the network is partitioned and heals?
  - Different LS databases must be synchronized

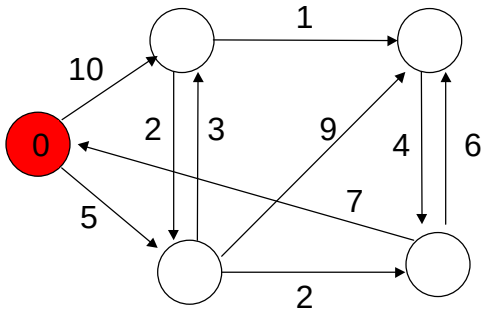
Shortest Paths: Dijkstra's Algorithm

- Graph algorithm for single-source shortest path

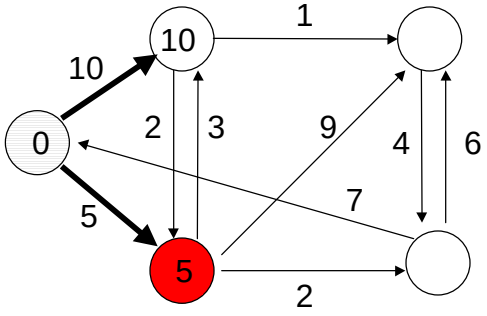
```
S ← {}
Q ← <all nodes keyed by distance>
While Q != {}
        u ← extract-min(Q)
        S ← S plus {u}
        for each node v adjacent to u
                "relax" the cost of v
```

←u is done, add to shortest paths

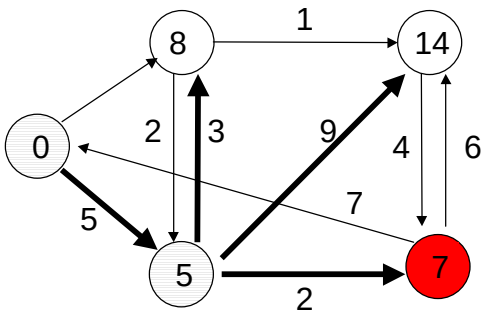
Dijkstra Example – Step 1



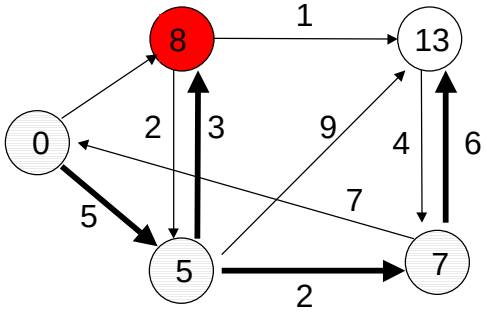
Dijkstra Example – Step 2



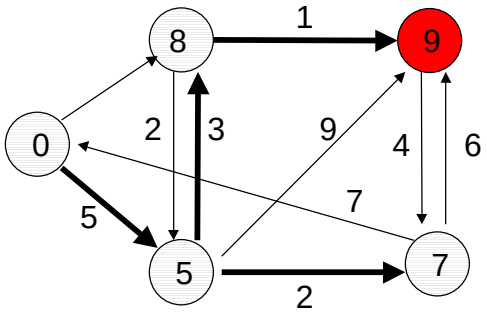
Dijkstra Example – Step 3



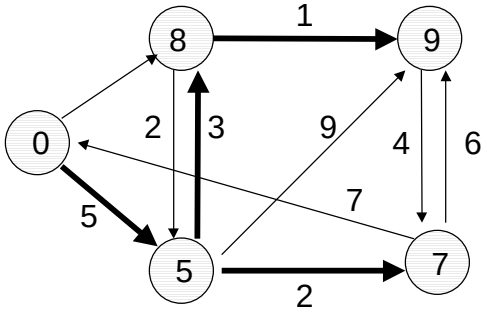
Dijkstra Example – Step 4



Dijkstra Example – Step 5



Dijkstra Example – Done



Open Shortest Path First (OSPF)

- Most widely-used link state protocol today
- Reliable flooding and shortest path calculation, plus many features:
  - Authentication of routing messages
  - Extra hierarchy: partition into routing areas
  - Load balancing: multiple equal cost routes

Link State vs. Distance Vector

- Link State
  - tell everyone about your neighbors
  - only say what you know for sure
- Distance Vector
  - tell neighbors about everyone
  - say what your neighbors told you
- Harder to track original source of information in distance vector

Question

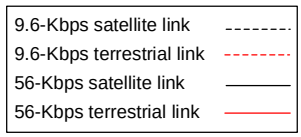
- Does link state algorithm guarantee routing tables are loop free?
  - Yes if everyone has the same information
  - No if updates are propagating
- Is path-based distance vector loop free?
  - Same problem
- Solution: IP header has “time-to-live”
  - at each router, decrement and discard if zero

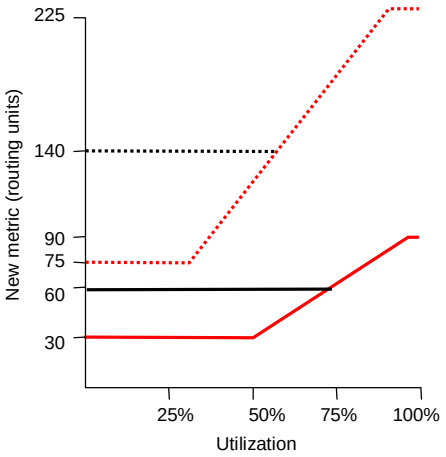
Cost Metrics

- How should we choose cost?
  - To get high bandwidth, low delay or low loss?
  - Do costs depend on the load?
- Static Metrics
  - Unit cost? Treats OC48 same as ISDN
  - Inverse bandwidth? Typical default
  - Manually tweak to yield desired goal?
- Dynamic Metrics
  - Depend on load; try to avoid hotspots (congestion)
  - But can lead to oscillations (damping needed)

Revised ARPANET Cost Metric

- Based on load and link
- Variation limited (3:1) and change damped
- Capacity dominates at low load; only try to reroute traffic if high load





Scalability Concerns

- Size of routing tables ~ # of hosts
- IP packet header limits # of hosts to 4 billion
  - Use bigger IP packet headers?
- Is it feasible for entire Internet to use the same distance vector or link state routing protocol?
  - Different organizations own different routers, want different policies
- Routing overhead grows with size of Internet
  - Volume of routing messages = O(# of routers^2)
  - Routing computation = O(# of routers^2)

⇒ RIP/OSPF do not scale to the size of the Internet

Routing Scalability

- Hierarchical IP addressing
  - Allocate addresses to match network structure
  - Aggregate addresses dynamically
    - IP prefix lookup
  - Virtual IP addresses
    - NATs and load balancers
- Interdomain routing (BGP)
  - Intradomain flexibility
  - Explicit policy knobs for cross-organizational resource allocation
- Wild card: multihoming

Structure of the Internet

Inter-domain versus intra-domain routing