CS118 Discussion 1A, Week 7

Zengwen Yuan
Dodd Hall 78, Friday 10:00—11:50 a.m.
Outline

• Network control plane
  • Routing
    • Link state routing (OSPF)
    • Distance vector routing (RIP)
  • BGP
  • ICMP
• Midterm/Project 2
Distance vector routing

- Bellman-Ford equation (dynamic programming)

- let

- \( dx(y) := \) cost of least-cost path from \( x \) to \( y \)

- then

- \( dx(y) = ? \)
Distance vector routing

- Bellman-Ford equation (dynamic programming)

- let

- \( dx(y) := \text{cost of least-cost path from } x \text{ to } y \)

- then

- \( dx(y) = \min_v \{ c(x,v) + dv(y) \} \)
Distance vector routing: example

• What’s the cost of least-cost path for $u \rightarrow z$?
Let’s work it out

- clearly:
  - $dv(z) = \_\_\_, dx(z) = \_\_\_, dw(z) = \_\_\_$
Let’s work it out

• clearly:
  
  • $d\nu(z) = 5$, $dx(z) = 3$, $dw(z) = 3$

• According to B-F equation:
  
  • $du(z) = \min \{ \ ? \}$
Let’s work it out

• clearly:
  • $dv(z) = 5$, $dx(z) = 3$, $dw(z) = 3$

• According to B-F equation:
  • $du(z) = \min \{c(u,v) + dv(z), c(u,x) + dx(z), c(u,w) + dw(z)\}$
Let’s work it out

- clearly:
  - $dv(z) = 5$, $dx(z) = 3$, $dw(z) = 3$

- According to B-F equation:
  - $du(z) = \min \{c(u,v) + dv(z), c(u,x) + dx(z), c(u,w) + dw(z)\}$
    - $= \min \{2 + 5, 1 + 3, 5 + 3\} = 4$
Distance vector routing: key idea

- from time-to-time, each node sends its own distance vector estimate to neighbors
- when x receives new DV estimate from neighbor, it updates its own DV using B-F equation.
Distance vector routing: caveat

- Count-to-infinity problem.
- Can you work out an example?
Distance vector routing: caveat

• Count-to-infinity problem.

• Can you work out an example?
Distance vector routing: caveat

- Count-to-infinity problem.
- Can you work out an example?

![Diagram: A/3 to B/2, then C/1 with an error to D]

- Can you propose a solution?
  - basic idea?
Distance vector routing: split horizon

• Previous solution idea:
  • split horizon
    • if B reaches D through C, B should not tell C that B can reach D
    • Then C will not attempt to go through B to reach D
  • Are we good?
Distance vector routing: split horizon

- Previous solution idea:
  - split horizon
    - if B reaches D through C, B should not tell C that B can reach D
    - Then C will not attempt to go through B to reach D
  - Are we good?
Distance vector routing: poison reverse

- Split horizon + poison reverse

  - if A reaches D through C:
    - A tells C that A’s distance to D is infinite
    - Then C will not attempt to go through A to reach D
    - In practice, infinite == 16 hops
## Link State v.s. Distance Vector

<table>
<thead>
<tr>
<th></th>
<th>Link state</th>
<th>Distance vector</th>
</tr>
</thead>
<tbody>
<tr>
<td>message complexity</td>
<td>with n nodes, E links, $O(nE)$ msgs sent</td>
<td>exchange between neighbors only (convergence time varies)</td>
</tr>
<tr>
<td>convergence speed</td>
<td>$O(n^2)$ algorithm requires $O(nE)$ msgs</td>
<td>convergence time varies (may be routing loops)</td>
</tr>
<tr>
<td>robustness</td>
<td>node can advertise incorrect link cost; each node computes only its own table</td>
<td>DV node can advertise incorrect path cost; error propagate thru network</td>
</tr>
<tr>
<td>implementation</td>
<td>OSPF</td>
<td>RIP</td>
</tr>
</tbody>
</table>
Summary

• Link-state routing (Dijkstra) algorithm:
  • each node computes the shortest paths to all the other nodes based on the complete topology map

• Distance Vector (Bellman-Ford) routing algorithm:
  • each node computes the shortest paths to all the other nodes based on its neighbors distance to all destinations
Inter-domain routing

- aggregate routers into regions
- AS: autonomous systems
- routers in same AS run same routing protocol
- “intra-AS” routing protocol
- routers in different AS can run different intra-AS routing protocol
BGP (Border Gateway Protocol)

• An inter-domain routing protocol; allows subnet to advertise its existence to rest of Internet: “I am here”

• BGP provides each AS a means to:
  
  • eBGP: obtain subnet reachability information from neighboring ASs.
  
  • iBGP: propagate reachability information to all AS-internal routers.

• How BGP works with intra-domain routing (e.g. OSPF)

Important BGP path advertisement example on Chapter 5 slides 49—50
BGP: routing policy

- A, B, C are provider networks
- X, W, Y are customer (of provider networks)
- X is attached to two networks.
  - It does not want to route from B via X to C
  - ... so X will not advertise to B a route to C

Legend:
- Blue rectangle: provider network
- Blue circle: customer network
BGP: routing policy

- A advertises path AW to B
- B advertises path BAW to X
- Should B advertise path BAW to C?
BGP: routing policy

- A advertises path AW to B
- B advertises path BAW to X

Should B advertise path BAW to C?

- No! B gets no “revenue” for routing CBAW since neither W nor C are B’s customers

- B wants to force C to route to w via A
- B wants to route only to/from its customers!
BGP: practice problems

• Explain how loops in paths can be detected in BGP.

• BGP advertisements contain complete paths showing the AS’s the path passes through, and so a router can easily identify a loop because an AS will appear two or more times.
BGP: practice problems

• Suppose that there is another stub network V that is a customer of ISP A. Suppose that B and C have a peering relationship, and A is a customer of both B and C. Suppose that A would like to have the traffic destined to W to come from B only, and the traffic destined to V from either B or C. How should A advertise its routes to B and C? What AS routes does C receive?

• A should advertise to B two routes: A-W and A-V

• A should advertise to C only one route: A-V

Routing: summary

- Intra-domain routing V.S. inter-domain routing
  - Performance V.S. policy
- Scalability: hierarchical routing
- Distance-vector routing V.S. link-state routing
  - Fully-distributed algorithm V.S. decentralized algorithm
- Unicast V.S. multicast
SDN: software defined networking

- A logically centralized control plane
  - easier network management
  - programmable forwarding table (OpenFlow API)
  - open (non-proprietary) implementation of control plane
- Components
  - data plane switches
  - SDN controller
  - network-control apps

Hot Internet research topic; P4 programmable switch
ICMP: Internet Control Message Protocol

- Used for feedback, status checking, error reporting at IP layer
- ICMP msgs are carried in IP packets
- `ping`: echo request/reply
- `traceroute`: nth packet has TTL = n
$ traceroute 8.8.8.8
traceroute to 8.8.8.8 (8.8.8.8), 64 hops max, 52 byte packets
1 172.30.40.3 (172.30.40.3)  4.055 ms  3.017 ms  3.871 ms
2 wifi-131-179-60-1.host.ucla.edu (131.179.60.1)  2.545 ms  2.288 ms  2.714 ms
3 ra00f1.anderson--cr00f2.csb1.ucla.net (169.232.8.12)  3.653 ms  3.506 ms  3.724 ms
4 cr00f2.csb1--bd11f1.anderson.ucla.net (169.232.4.5)  3.959 ms  4.383 ms  3.483 ms
5 lax-agg6--ucla-10g.cenic.net (137.164.24.134)  3.951 ms  5.480 ms  3.840 ms
6 74.125.49.165 (74.125.49.165)  6.558 ms  3.882 ms  3.890 ms
7 108.170.247.129 (108.170.247.129)  3.192 ms
108.170.247.193 (108.170.247.193)  93.964 ms
108.170.247.161 (108.170.247.161)  3.297 ms
8 108.177.3.127 (108.177.3.127)  3.657 ms
209.85.255.73 (209.85.255.73)  3.571 ms
108.177.3.129 (108.177.3.129)  3.261 ms
9 google-public-dns-a.google.com (8.8.8.8)  5.315 ms  3.770 ms  12.165 ms
Traceroute: example