CS144: An Introduction to Computer Networks

Routing: How do packets know the way?

Today:
Bellman Ford and Dijkstra, and
How does routing work in the Internet?
(Part 2 of 3)
Videos and Lectures this week

Lectures: Mostly the “why” we do it this way
Videos: Mostly the “what” and the “how”
I assume you watched three videos: Basics, Bellman-Ford & Dijkstra

Today’s lecture and discussion:
1. Recap what we learned in Lecture 1
2. Lowest-cost spanning tree: What if each link has a different cost?
3. Dijkstra vs. Bellman Ford
4. How routing works in the Internet

Before Friday, watch Videos Internet & BGP
Recap of Lecture 1

Methods used for routing in the Internet:
1. Flooding: When we don’t know the topology
2. Source routing: When the end host wants to pick the route
3. A distributed algorithm such as Bellman-Ford or Dijkstra: Routers deliver packet once to the correct destination along the lowest-cost path.

Bellman-Ford (distance-vector algorithm):
- Fully distributed
- Routers do not need to know the topology
- Can be hard to stabilize when links, weights or routers change
- Was the basis of the original RIP routing protocol

Dijkstra (link-state algorithm):
- Two phases: determine topology, then runs Dijkstra’s algorithm
- Each router calculates lowest cost spanning tree to every other router
- Is the basis of the widely used OSPF and IS-IS protocols
What if each link has a “cost”?
“Expensive link”:
It might be very long, e.g. a link from Europe to USA.
Or it might be very busy, e.g. it connects to Google or CNN.
Or it may be very slow, e.g. 1Mb/s instead of 100Mb/s.
Find lowest cost path to H
Find lowest cost path to H
Find the lowest cost spanning tree rooted on 1
Find the lowest cost spanning tree rooted on 1

Cost from router 2 to router 1 = 22

...and so on!

Router 21 tells Router 16: “You can reach router 1 via me with a cost of 17”

Router 21 hears a lower cost path to Router 1. It tells Router 16: “You can reach router 1 via me with a cost of 13”
The Distributed Bellman-Ford Algorithm

Example: Find min-cost spanning tree to router $R$

- Assume routers know cost of link to each neighbor.
- Router $R_i$ maintains value of cost $C_i$ to reach $R$, and the next hop.
- Vector $\mathbf{C}=(C_1, C_2,\ldots)$ is the distance vector to $R$.
- Initially, set $\mathbf{C} = (\infty, \infty, \ldots \infty)$
  1. After $T$ seconds, $R_i$ sends $C_i$ to its neighbors.
  2. If $R_i$ learns of a lower cost path, update $C_i$. Remember next hop.
  3. Repeat.
The Distributed Bellman-Ford Algorithm

Questions:
1. What is the maximum run time of the algorithm?
2. Will the algorithm always converge?
3. What happens when routers/links fail?
Observations

Bellman Ford: Doesn’t need to know topology.
Distributed. Hard to stabilize.

Dijkstra: more work, but easier to stabilize, particularly with different link costs.

The routers in most enterprises, including Stanford, use IETF standard protocols based on Dijkstra: IS-IS or OSPF.
How routing works in the Internet
How routing works in the Internet

Internet routing is hierarchical.
The Internet is divided into Autonomous Systems (AS’s).
Each AS owner can decide how to route packets within each AS.
Everyone agrees to use the same protocol (BGP) to route packets between AS’s.
In the Internet, Autonomous Systems (AS’s) have Border Routers (orange). The border routers route packets to each other using the Border Gateway Protocol (BGP).
nickm@yuba.Stanford.EDU > traceroute -q1 www.cam.ac.uk
traceroute to www.cam.ac.uk (128.232.132.8), 30 hops max, 40 byte packets
1  csee-west-rtr-vl3874.SUNet (171.64.74.2)  0.229 ms
2  he-rtr-vlan12.SUNet (171.66.0.209)  1.531 ms
3  100ge5-1.core1.pao1.he.net (184.105.177.237)  0.638 ms
4  10ge7-5.core1.sjc2.he.net (72.52.92.70)  1.311 ms
5  100ge10-2.core1.nyc4.he.net (184.105.81.217)  62.771 ms
6  100ge16-2.core1.lon2.he.net (72.52.92.165)  145.243 ms
7  linx-gw1.ja.net (195.66.224.15)  136.102 ms
8  ae23.londtt-sbr1.ja.net (146.97.35.169)  135.434 ms
9  [...]  
10 uoc.ja.net (146.97.41.38)  169.232 ms
11 d-dw.s-dw.net.cam.ac.uk (193.60.88.2)  142.827 ms
12 [...]
nickm@yuba.Stanford.EDU > traceroute -q1 www.cam.ac.uk
tracert to www.cam.ac.uk (128.232.132.8), 30 hops max, 40 byte packets
1 csee-west-rtr-vl3874.SUNet (171.64.74.2) 0.229 ms
2 he-rtr-vlan12.SUNet (171.66.0.209) 1.531 ms
3 100ge5-1.core1.pao1.he.net (184.105.177.237) 0.638 ms
4 10ge7-5.core1.sjc2.he.net (72.52.92.70) 1.311 ms
5 100ge10-2.core1.nyc4.he.net (184.105.81.217) 62.771 ms
6 100ge16-2.core1.lon2.he.net (72.52.92.165) 145.243 ms
7 linx-gw1.ja.net (195.66.224.15) 136.102 ms
8 ae23.londtt-sbr1.ja.net (146.97.35.169) 135.434 ms
9 [...] 12 uoc.ja.net (146.97.41.38) 169.232 ms
13 d-dw.s-dw.net.cam.ac.uk (193.60.88.2) 142.827 ms
14 [...]
AS (Autonomous System) numbers

nickm@yuba.Stanford.EDU > traceroute -q1 www.cam.ac.uk
traceroute to www.cam.ac.uk (128.232.132.8), 30 hops max, 40 byte packets
1 csee-west-rtr-vl3874.SUNet (171.64.74.2) 0.229 ms
2 he-rtr-vlan12.SUNet (171.66.0.209) 1.531 ms
3 100ge5-1.core1.pao1.he.net (184.105.177.237) 0.638 ms
4 10ge7-5.core1.sjc2.he.net (72.52.92.70) 1.311 ms
5 100ge10-2.core1.nyc4.he.net (184.105.81.217) 62.771 ms
6 100ge16-2.core1.lon2.he.net (72.52.92.165) 145.243 ms
7 linx-gw1.ja.net (195.66.224.15) 136.102 ms
8 ae23.londtt-sbr1.ja.net (146.97.35.169) 135.434 ms
9 [...]
12 uoc.ja.net (146.97.41.38) 169.232 ms
13 d-dw.s-dw.net.cam.ac.uk (193.60.88.2) 142.827 ms
14 [...]

e.g. yuba.Stanford.edu

nickm> whois -h whois.cymru.com 146.97.35.169

<table>
<thead>
<tr>
<th>AS</th>
<th>IP</th>
<th>AS Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>786</td>
<td>146.97.35.169</td>
<td>JANET Jisc Services Limited, GB</td>
</tr>
</tbody>
</table>

JANET is AS 786

nickm> whois -h whois.cymru.com 171.64.74.155

<table>
<thead>
<tr>
<th>AS</th>
<th>IP</th>
<th>AS Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>32</td>
<td>171.64.74.155</td>
<td>STANFORD, US</td>
</tr>
</tbody>
</table>

Stanford is AS 32
Autonomous Systems (AS’s) usually connect to each other in an Internet eXchange Point (IXP)

Stanford University

yuba.stanford.edu

Cambridge University (UK)

Hurricane Electric (he.net)

JANET (ja.net)

(transatlantic cable)

There are hundreds of IXPs worldwide
Hurricane Electric (he.net)
Stanford University
yuba.stanford.edu
Stanford
AS 32
he.net
AS 6939
CENIC
AS 2152
Level3
AS 3356

Inside the IXP

Other AS’s at same IXP
Hurricane Electric (he.net)