Outline

• Lecture review:
  • BGP routing
  • Link layer
• Project 2 questions
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
Path attributes and BGP routes

• advertised prefix includes BGP attributes
  • prefix + attributes = “route”

• two important attributes:
  • AS-PATH: list of ASes through which prefix advertisement has passed
  • NEXT-HOP: indicates specific internal-AS router to next-hop AS

• Policy-based routing:
  • gateway receiving route advertisement uses import policy to accept/decline path (e.g., never route through AS Y).
  • AS policy also determines whether to advertise path to other other neighboring ASes using the local preference value
BGP path advertisement

- AS2 router 2c receives path advertisement AS3, X (via eBGP) from AS3 router 3a
- Based on AS2 policy, AS2 router 2c accepts path AS3,X, propagates (via iBGP) to all AS2 routers
- Based on AS2 policy, AS2 router 2a advertises (via eBGP) path AS2, AS3, X to AS1 router 1c
BGP path advertisement

- Gateway router may learn about multiple paths to destination:
  - AS1 gateway router 1c learns path AS2,AS3,X from 2a
  - AS1 gateway router 1c learns path AS3,X from 3a
  - Based on policy, AS1 gateway router 1c chooses path AS3,X, and advertises path within AS1 via iBGP
Hot Potato Routing

• 2d learns (via iBGP) it can route to X via 2a or 2c

• hot potato routing: choose local gateway that has least intra-domain cost (e.g., 2d chooses 2a, even though more AS hops to X): don’t worry about inter-domain cost!
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
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

<table>
<thead>
<tr>
<th>No.</th>
<th>Time</th>
<th>Source</th>
<th>Destination</th>
<th>Protocol</th>
<th>Length</th>
<th>Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>2360</td>
<td>18:17:48.137358</td>
<td>172.30.42.132</td>
<td>8.8.8.8</td>
<td>UDP</td>
<td>66</td>
<td>42995 ➔ 33435 Len=24</td>
</tr>
<tr>
<td>2361</td>
<td>18:17:48.1411025</td>
<td>172.30.42.132</td>
<td>8.8.8.8</td>
<td>ICMP</td>
<td>70</td>
<td>Time-to-live exceeded (Time to live exceeded in transit)</td>
</tr>
<tr>
<td>2364</td>
<td>18:17:48.145895</td>
<td>172.30.42.132</td>
<td>8.8.8.8</td>
<td>UDP</td>
<td>66</td>
<td>42995 ➔ 33436 Len=24</td>
</tr>
<tr>
<td>2365</td>
<td>18:17:48.148795</td>
<td>172.30.42.132</td>
<td>8.8.8.8</td>
<td>ICMP</td>
<td>70</td>
<td>Time-to-live exceeded (Time to live exceeded in transit)</td>
</tr>
<tr>
<td>2366</td>
<td>18:17:48.148917</td>
<td>172.30.42.132</td>
<td>8.8.8.8</td>
<td>UDP</td>
<td>66</td>
<td>42995 ➔ 33437 Len=24</td>
</tr>
<tr>
<td>2367</td>
<td>18:17:48.152677</td>
<td>172.30.42.132</td>
<td>8.8.8.8</td>
<td>ICMP</td>
<td>70</td>
<td>Time-to-live exceeded (Time to live exceeded in transit)</td>
</tr>
<tr>
<td>2368</td>
<td>18:17:48.152839</td>
<td>172.30.42.132</td>
<td>8.8.8.8</td>
<td>UDP</td>
<td>66</td>
<td>42995 ➔ 33438 Len=24</td>
</tr>
<tr>
<td>2369</td>
<td>18:17:48.155208</td>
<td>172.30.42.132</td>
<td>8.8.8.8</td>
<td>ICMP</td>
<td>70</td>
<td>Time-to-live exceeded (Time to live exceeded in transit)</td>
</tr>
<tr>
<td>2372</td>
<td>18:17:48.161801</td>
<td>172.30.42.132</td>
<td>8.8.8.8</td>
<td>UDP</td>
<td>66</td>
<td>42995 ➔ 33439 Len=24</td>
</tr>
<tr>
<td>2373</td>
<td>18:17:48.163969</td>
<td>172.30.42.132</td>
<td>8.8.8.8</td>
<td>ICMP</td>
<td>70</td>
<td>Time-to-live exceeded (Time to live exceeded in transit)</td>
</tr>
<tr>
<td>2374</td>
<td>18:17:48.164015</td>
<td>172.30.42.132</td>
<td>8.8.8.8</td>
<td>UDP</td>
<td>66</td>
<td>42995 ➔ 33440 Len=24</td>
</tr>
<tr>
<td>2375</td>
<td>18:17:48.166708</td>
<td>172.30.42.132</td>
<td>8.8.8.8</td>
<td>ICMP</td>
<td>70</td>
<td>Time-to-live exceeded (Time to live exceeded in transit)</td>
</tr>
<tr>
<td>2376</td>
<td>18:17:48.166852</td>
<td>172.30.42.132</td>
<td>8.8.8.8</td>
<td>UDP</td>
<td>66</td>
<td>42995 ➔ 33441 Len=24</td>
</tr>
<tr>
<td>2377</td>
<td>18:17:48.178369</td>
<td>169.232.8.12</td>
<td>172.30.42.132</td>
<td>ICMP</td>
<td>70</td>
<td>Time-to-live exceeded (Time to live exceeded in transit)</td>
</tr>
<tr>
<td>2380</td>
<td>18:17:48.174842</td>
<td>172.30.42.132</td>
<td>8.8.8.8</td>
<td>UDP</td>
<td>66</td>
<td>42995 ➔ 33442 Len=24</td>
</tr>
<tr>
<td>2381</td>
<td>18:17:48.178190</td>
<td>169.232.8.12</td>
<td>172.30.42.132</td>
<td>ICMP</td>
<td>70</td>
<td>Time-to-live exceeded (Time to live exceeded in transit)</td>
</tr>
<tr>
<td>2382</td>
<td>18:17:48.178346</td>
<td>172.30.42.132</td>
<td>8.8.8.8</td>
<td>UDP</td>
<td>66</td>
<td>42995 ➔ 33443 Len=24</td>
</tr>
<tr>
<td>2383</td>
<td>18:17:48.181957</td>
<td>169.232.8.12</td>
<td>172.30.42.132</td>
<td>ICMP</td>
<td>70</td>
<td>Time-to-live exceeded (Time to live exceeded in transit)</td>
</tr>
<tr>
<td>2384</td>
<td>18:17:48.182086</td>
<td>172.30.42.132</td>
<td>8.8.8.8</td>
<td>UDP</td>
<td>66</td>
<td>42995 ➔ 33444 Len=24</td>
</tr>
<tr>
<td>2385</td>
<td>18:17:48.189938</td>
<td>169.232.8.12</td>
<td>172.30.42.132</td>
<td>ICMP</td>
<td>110</td>
<td>Time-to-live exceeded (Time to live exceeded in transit)</td>
</tr>
<tr>
<td>2388</td>
<td>18:17:48.189767</td>
<td>172.30.42.132</td>
<td>8.8.8.8</td>
<td>UDP</td>
<td>66</td>
<td>42995 ➔ 33445 Len=24</td>
</tr>
<tr>
<td>2389</td>
<td>18:17:48.193689</td>
<td>169.232.8.12</td>
<td>172.30.42.132</td>
<td>ICMP</td>
<td>110</td>
<td>Time-to-live exceeded (Time to live exceeded in transit)</td>
</tr>
<tr>
<td>2390</td>
<td>18:17:48.194171</td>
<td>172.30.42.132</td>
<td>8.8.8.8</td>
<td>UDP</td>
<td>66</td>
<td>42995 ➔ 33446 Len=24</td>
</tr>
<tr>
<td>2391</td>
<td>18:17:48.197563</td>
<td>169.232.8.12</td>
<td>172.30.42.132</td>
<td>ICMP</td>
<td>110</td>
<td>Time-to-live exceeded (Time to live exceeded in transit)</td>
</tr>
<tr>
<td>2392</td>
<td>18:17:48.197664</td>
<td>172.30.42.132</td>
<td>8.8.8.8</td>
<td>UDP</td>
<td>66</td>
<td>42995 ➔ 33447 Len=24</td>
</tr>
<tr>
<td>2393</td>
<td>18:17:48.201536</td>
<td>172.30.42.132</td>
<td>172.30.42.132</td>
<td>ICMP</td>
<td>110</td>
<td>Time-to-live exceeded (Time to live exceeded in transit)</td>
</tr>
<tr>
<td>2396</td>
<td>18:17:48.205938</td>
<td>172.30.42.132</td>
<td>8.8.8.8</td>
<td>UDP</td>
<td>66</td>
<td>42995 ➔ 33448 Len=24</td>
</tr>
<tr>
<td>2397</td>
<td>18:17:48.211286</td>
<td>172.30.42.132</td>
<td>172.30.42.132</td>
<td>ICMP</td>
<td>110</td>
<td>Time-to-live exceeded (Time to live exceeded in transit)</td>
</tr>
<tr>
<td>2398</td>
<td>18:17:48.211432</td>
<td>172.30.42.132</td>
<td>8.8.8.8</td>
<td>UDP</td>
<td>66</td>
<td>42995 ➔ 33449 Len=24</td>
</tr>
<tr>
<td>2399</td>
<td>18:17:48.215159</td>
<td>172.30.42.132</td>
<td>172.30.42.132</td>
<td>ICMP</td>
<td>110</td>
<td>Time-to-live exceeded (Time to live exceeded in transit)</td>
</tr>
<tr>
<td>2400</td>
<td>18:17:48.215293</td>
<td>172.30.42.132</td>
<td>8.8.8.8</td>
<td>UDP</td>
<td>66</td>
<td>42995 ➔ 33450 Len=24</td>
</tr>
</tbody>
</table>

Fragment offset: 0
Time to live: 5
Protocol: UDP (17)
Header checksum: 0x2707 [validation disabled]
Source: 172.30.42.132
Destination: 8.8.8.8
Link layer: introduction

- understand principles behind link layer services:
  - data framing
  - error detection, correction — CRC (cyclic redundancy check)
  - sharing a broadcast channel: multiple access
  - link layer addressing
- local area networks: Ethernet, VLANs
Delivering packet at link layer

• Basic communication models
  • Unicast
  • Broadcast
  • Multicast
• What, how, why?
Broadcast

• Replicate at source, or in-network

• Flooding (FYI)
  • Reverse path forwarding (RPF)
  • Spanning tree
Multicast (FYI)

- Address range
  - class-D IP: 224.0.0.0 — 239.255.255.255

- Protocols:
  - IGMP
  - DVMRP, PIM
Medium Access Links and Protocols

- Two types: point-to-point, broadcast

- **Broadcast** channel shared by multiple hosts
  - What if we only have unicast channel?
  - What’s the pros and cons for a broadcast channel?

- Three classes of Multiple Access Control (MAC) protocols
  - Channel partitioning: FDMA, TDMA, CDMA
  - Random access: Aloha, CSMA/CD, Ethernet
  - Taking turns: Token ring/passing

- **Pros and cons for each class of protocol?**
Cyclic Redundancy Check (CRC)

- More powerful error-detection coding
  - view data bits, D, as a binary number
  - choose r+1 bit pattern (generator), G
  - goal: choose r CRC bits, R, such that
    - \(<D,R>\) exactly divisible by G (modulo 2)
    - receiver knows G, divides \(<D,R>\) by G. If non-zero remainder: error detected!
    - can detect all burst errors less than r+1 bits
  - widely used in practice (Ethernet, 802.11 WiFi, ATM)

\[ D: \text{data bits to be sent} \quad R: \text{CRC bits} \]
Random access: slotted ALOHA

• Assumptions:
  • all frames same size
  • time divided into equal size slots (time to transmit 1 frame)
  • nodes start to transmit only slot beginning
  • nodes are synchronized
  • if 2 or more nodes transmit in slot, all nodes detect collision
Random access: slotted ALOHA

• suppose: N nodes with many frames to send, each transmits in slot with probability p

• \( \Pr(\text{given node has success in a slot}) = p(1-p)^{(N-1)} \)

• \( \Pr(\text{any node has a success}) = Np(1-p)^{(N-1)} \)

• max efficiency: find \( p^* \) that maximizes \( Np(1-p)^{(N-1)} \)

• Take the limit of \( Np^*(1-p^*)^{(N-1)} \) as \( N \) goes to infinity, yields:
  • max efficiency = \( \frac{1}{e} = .37 \)
Random access: ALOHA efficiency

- Slotted ALOHA max efficiency = $1/e = 0.37$
- Unslotted ALOHA max efficiency = $1/2e = 0.18$
Project 2 hints

• Timer:
  • `<sys/time.h>` time() — :))))
  • `<unistd.h>` alarm() — cause the system to generate a SIGALRM signal: https://linux.die.net/man/3/alarm
  • C++ std::chrono

• Signal handler
  • C++: https://en.cppreference.com/w/cpp/utility/program/signal