# CS118 Discussion 1B, Week 8

Boyan Ding

# Outline

- Lecture review:
    - Internet routing: OSPF; BGP
    - Link layer

# Quiz 2 - logistics

- Time: 4-10pm (PST), Friday, May 28
  - Choose 2.5h within to finish the exam
- Covered material: TCP (From 3.4 on), Chapter 4 and Chapter 5 (up to link state/distance vector routing)
- Format: similar to quiz 1

# 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

# OSPF

- Link state algorithm
- Main functions
  - Broadcast link state info
  - Link failure detection: Neighbor nodes send HELLO msg to each other periodically

# OSPF

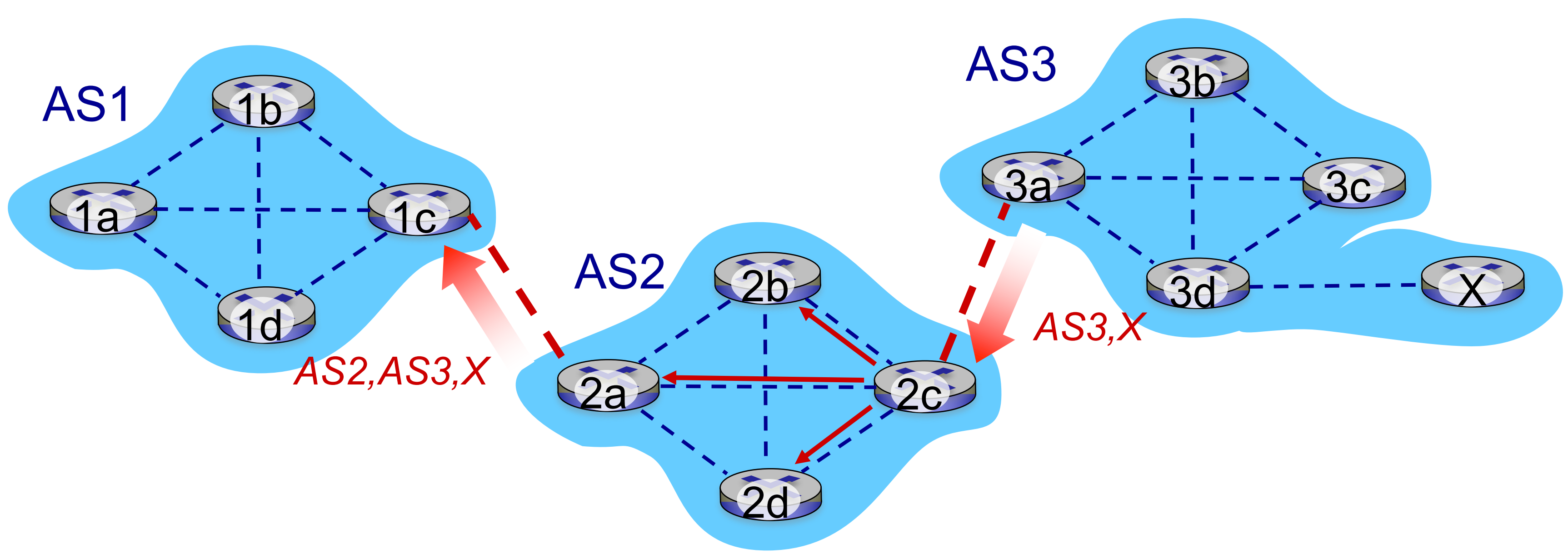
- Message:
  - HELLO message: used as heartbeat to detect failure
  - LSP: information of the node, the list of direct neighbors and link costs
    - Generated periodically or upon failure
    - Flooding of LSP
      - How to avoid loop? Check the message ID

# 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.
- Why do we need iBGP (when there is intra-AS routing)?

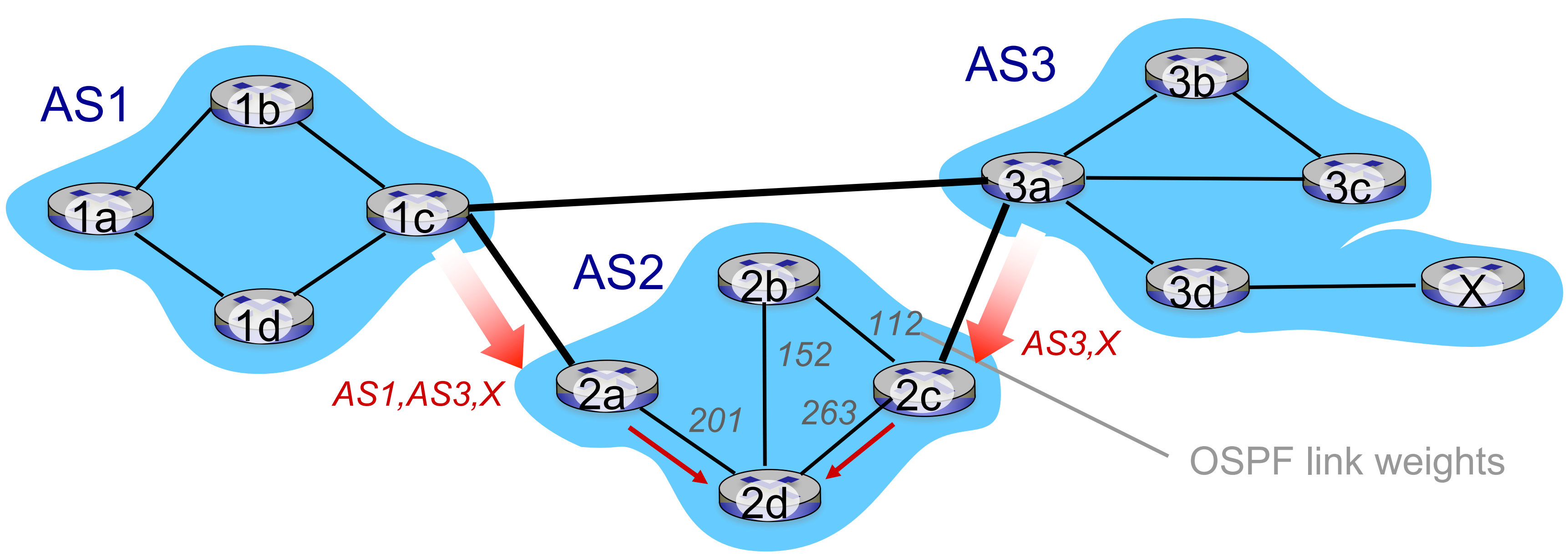
# BGP: iBGP and eBGP

- A single AS may have multiple gateways, and they may not even be adjacent to each other
- But they need to have a consistent view of the network
- Just consider how router 2a in AS2 know the path to AS3 and advertise it to AS1



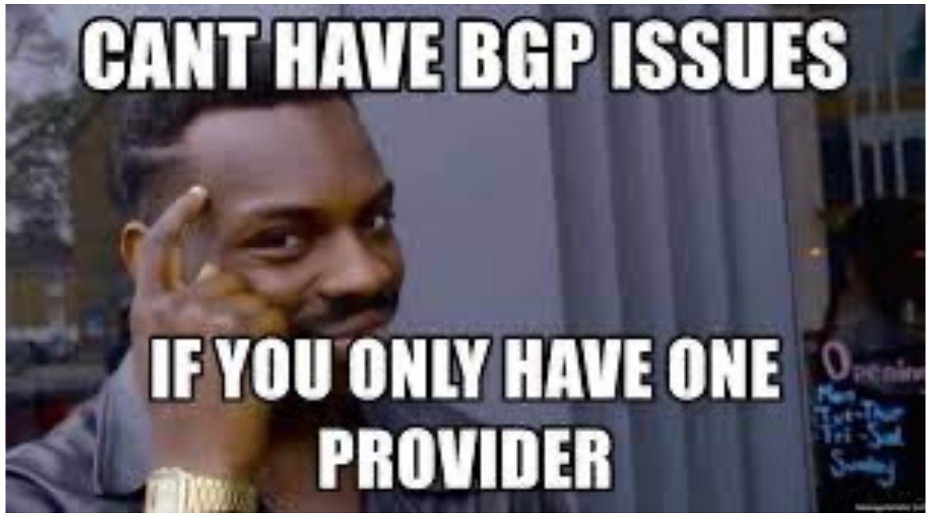
# BGP: Hot potato routing

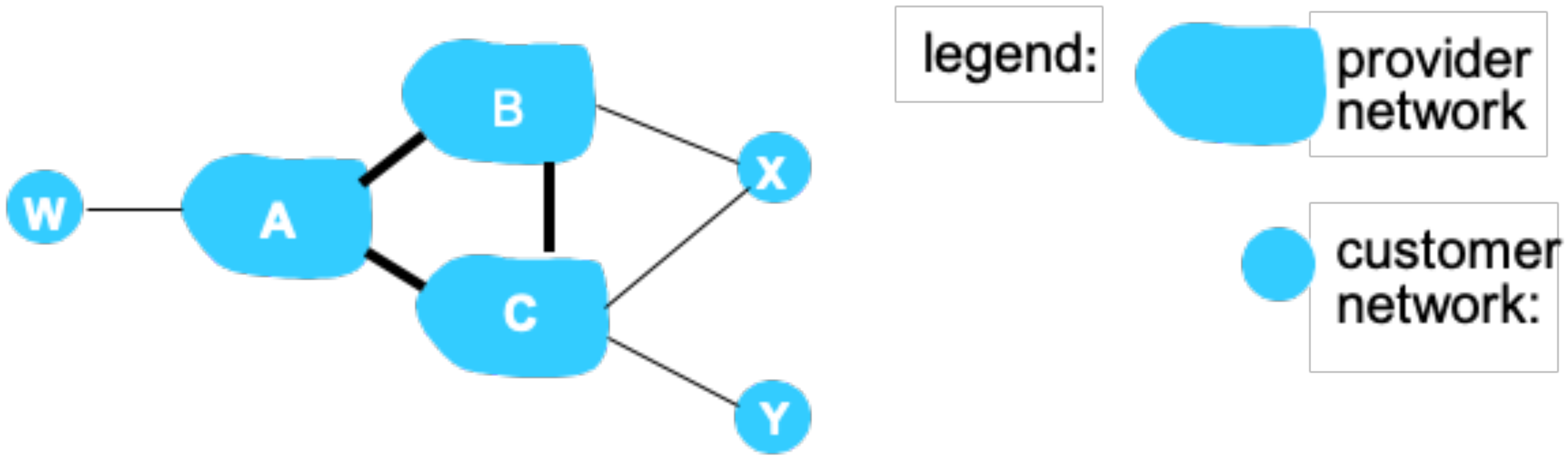
- Hot potato routing: choose local gateway that has least intra-domain cost (e.g., 2d chooses 2a, even though more AS hops to X): don't worry about inter-domain cost!



# BGP: routing policy

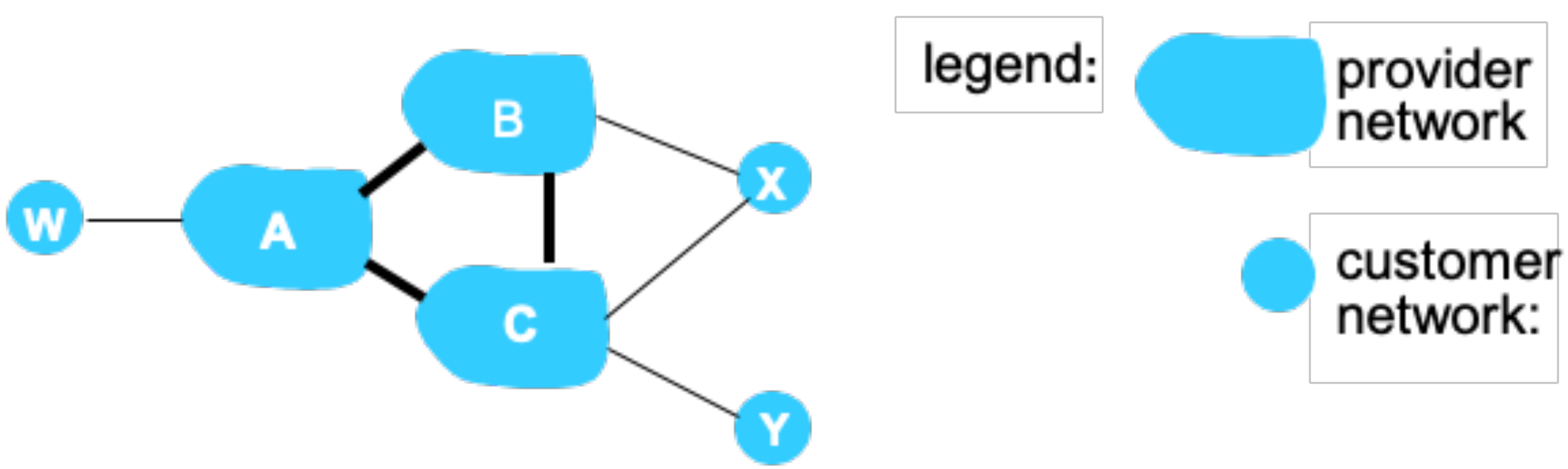
- A,B,C are provider networks
- X,W,Y are customer (of provider networks)
- X is attached to two networks.
    - It does not want to route from B via X to C
    - … so X will not advertise to B a route to C



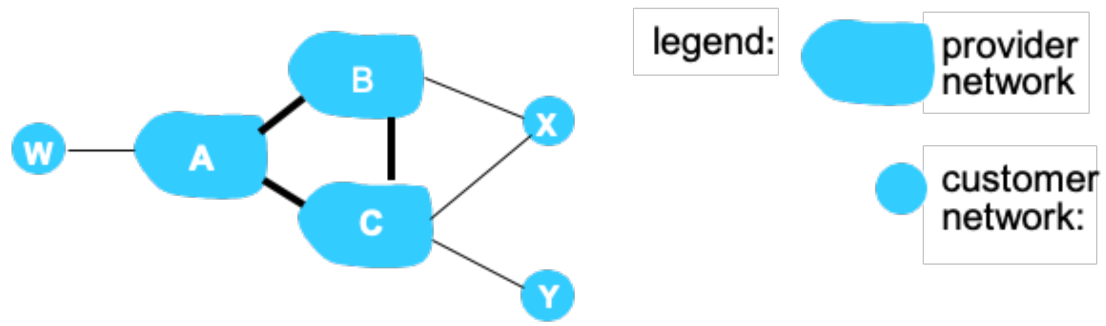



# BGP: routing policy

- 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.

# 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

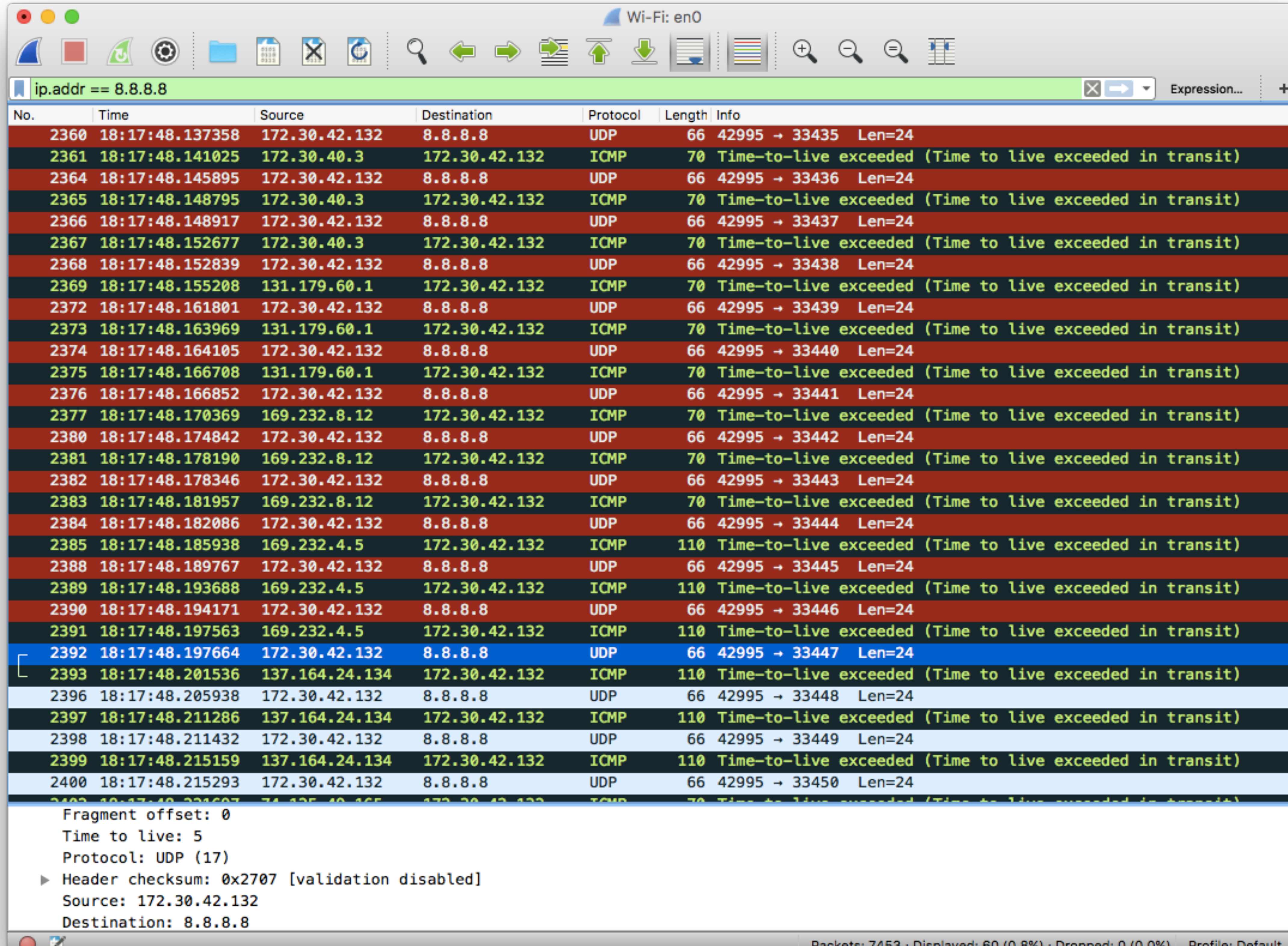
# ICMP: Internet Control Message Protocol

- 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: example

```
$ 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



Wi-Fi: en0

ip.addr == 8.8.8.8

Expression… +

| No. | Time | Source | Destination | Protocol | Length | Info |
|---|---|---|---|---|---|---|
| 2360 | 18:17:48.137358 | 172.30.42.132 | 8.8.8.8 | UDP | 66 | 42995 → 33435 Len=24 |
| 2361 | 18:17:48.141025 | 172.30.40.3 | 172.30.42.132 | ICMP | 70 | Time-to-live exceeded (Time to live exceeded in transit) |
| 2364 | 18:17:48.145895 | 172.30.42.132 | 8.8.8.8 | UDP | 66 | 42995 → 33436 Len=24 |
| 2365 | 18:17:48.148795 | 172.30.40.3 | 172.30.42.132 | ICMP | 70 | Time-to-live exceeded (Time to live exceeded in transit) |
| 2366 | 18:17:48.148917 | 172.30.42.132 | 8.8.8.8 | UDP | 66 | 42995 → 33437 Len=24 |
| 2367 | 18:17:48.152677 | 172.30.40.3 | 172.30.42.132 | ICMP | 70 | Time-to-live exceeded (Time to live exceeded in transit) |
| 2368 | 18:17:48.152839 | 172.30.42.132 | 8.8.8.8 | UDP | 66 | 42995 → 33438 Len=24 |
| 2369 | 18:17:48.155208 | 131.179.60.1 | 172.30.42.132 | ICMP | 70 | Time-to-live exceeded (Time to live exceeded in transit) |
| 2372 | 18:17:48.161801 | 172.30.42.132 | 8.8.8.8 | UDP | 66 | 42995 → 33439 Len=24 |
| 2373 | 18:17:48.163969 | 131.179.60.1 | 172.30.42.132 | ICMP | 70 | Time-to-live exceeded (Time to live exceeded in transit) |
| 2374 | 18:17:48.164105 | 172.30.42.132 | 8.8.8.8 | UDP | 66 | 42995 → 33440 Len=24 |
| 2375 | 18:17:48.166708 | 131.179.60.1 | 172.30.42.132 | ICMP | 70 | Time-to-live exceeded (Time to live exceeded in transit) |
| 2376 | 18:17:48.166852 | 172.30.42.132 | 8.8.8.8 | UDP | 66 | 42995 → 33441 Len=24 |
| 2377 | 18:17:48.170369 | 169.232.8.12 | 172.30.42.132 | ICMP | 70 | Time-to-live exceeded (Time to live exceeded in transit) |
| 2380 | 18:17:48.174842 | 172.30.42.132 | 8.8.8.8 | UDP | 66 | 42995 → 33442 Len=24 |
| 2381 | 18:17:48.178190 | 169.232.8.12 | 172.30.42.132 | ICMP | 70 | Time-to-live exceeded (Time to live exceeded in transit) |
| 2382 | 18:17:48.178346 | 172.30.42.132 | 8.8.8.8 | UDP | 66 | 42995 → 33443 Len=24 |
| 2383 | 18:17:48.181957 | 169.232.8.12 | 172.30.42.132 | ICMP | 70 | Time-to-live exceeded (Time to live exceeded in transit) |
| 2384 | 18:17:48.182086 | 172.30.42.132 | 8.8.8.8 | UDP | 66 | 42995 → 33444 Len=24 |
| 2385 | 18:17:48.185938 | 169.232.4.5 | 172.30.42.132 | ICMP | 110 | Time-to-live exceeded (Time to live exceeded in transit) |
| 2388 | 18:17:48.189767 | 172.30.42.132 | 8.8.8.8 | UDP | 66 | 42995 → 33445 Len=24 |
| 2389 | 18:17:48.193688 | 169.232.4.5 | 172.30.42.132 | ICMP | 110 | Time-to-live exceeded (Time to live exceeded in transit) |
| 2390 | 18:17:48.194171 | 172.30.42.132 | 8.8.8.8 | UDP | 66 | 42995 → 33446 Len=24 |
| 2391 | 18:17:48.197563 | 169.232.4.5 | 172.30.42.132 | ICMP | 110 | Time-to-live exceeded (Time to live exceeded in transit) |
| 2392 | 18:17:48.197664 | 172.30.42.132 | 8.8.8.8 | UDP | 66 | 42995 → 33447 Len=24 |
| 2393 | 18:17:48.201536 | 137.164.24.134 | 172.30.42.132 | ICMP | 110 | Time-to-live exceeded (Time to live exceeded in transit) |
| 2396 | 18:17:48.205938 | 172.30.42.132 | 8.8.8.8 | UDP | 66 | 42995 → 33448 Len=24 |
| 2397 | 18:17:48.211286 | 137.164.24.134 | 172.30.42.132 | ICMP | 110 | Time-to-live exceeded (Time to live exceeded in transit) |
| 2398 | 18:17:48.211432 | 172.30.42.132 | 8.8.8.8 | UDP | 66 | 42995 → 33449 Len=24 |
| 2399 | 18:17:48.215159 | 137.164.24.134 | 172.30.42.132 | ICMP | 110 | Time-to-live exceeded (Time to live exceeded in transit) |
| 2400 | 18:17:48.215293 | 172.30.42.132 | 8.8.8.8 | UDP | 66 | 42995 → 33450 Len=24 |

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

Packets: 7453 · Displayed: 60 (0.8%) · Dropped: 0 (0.0%) Profile: Default

# Link layer: introduction

- Link Layer: Provides data transport between adjacent nodes
  - The end-to-end Network Layer is built upon the hop-by-hop Link Layer
  - A single datagram may go through different link technologies along the way
  - Different link layer protocols may provide different kinds of service

# 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

# 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 (CSMA/CA)
  - Taking turns: Token ring/passing
  - **Pros and cons for each class of protocol?**

# 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(given node has success in a slot) = p(1-p)^(N-1)
- Pr(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 = 1/e = .37

# Random access: ALOHA efficiency

- Slotted ALOHA max efficiency = 1/e = .37
- Unslotted ALOHA max efficiency = 1/2e = .18