Administivia

• IP Project
  – You should get an email today with your group assignment and a repo link
  – Milestone: meet with me/a TA on/before October 11 to discuss your design (signups soon)

• Look for a post soon about debugging/workflow

• HW2: Announcement soon
  – Stuff we’ve covered + warmup for IP!
Today

Start of network layer

• IP forwarding mechanics
• About the IP project
Recall: IP Addressing

What’s in an IP address?

138.16.161.209

- **Host part**: Identifies your individual host within the network.
- **Network part**: What network you’re on (Brown).
Recall: IP addressing

A typical IP configuration looks like this:

Configure IPv4: Using DHCP
IPv4 Address: 138.16.161.209
Subnet Mask: 255.255.255.0
Router: 138.16.161.1

Network: 138.16.161.209/24
/24 => first 24 bits are network part
Which part is network part?
Subnet mask: always know those two parts!
IP 1.2.3.4
MASK 255.255.255.0

0000 0000 0000 0010 0000 0011 0000 0100
1111 1111 1111 1111 1111 1111 0000 0000

127 \rightarrow \text{FIRST 24 BIT ARE}

\text{CIDR NOTATION}

255.0.0.0 \Rightarrow 18 \text{ 16M HOSTS}
255.255.0.0 \Rightarrow 16 \text{ 65K HOSTS}
255.255.255.0 \Rightarrow 124 \text{ 256 HOSTS}

/128 1 HOST
Identifying the network

• Prefix or Subnet mask => identifies what part of the address is the network part
Identifying the network

- Prefix or Subnet mask => identifies what part of the address is the network part

Why do we care?
Routers forward to networks, not individual hosts!
How do we move packets between networks?
### IP forwarding

Given a packet, decide where to send it

**How? Forwarding Table**

**Key:** Prefix

Maps to:
- Some local interface
- Another IP ("next hop")

<table>
<thead>
<tr>
<th>Prefix</th>
<th>Interface/Next hop</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.168.1.0/24</td>
<td>WIFI</td>
</tr>
<tr>
<td>10.1.5.0/24</td>
<td>ETHERNET</td>
</tr>
<tr>
<td>*</td>
<td>192.168.1.1</td>
</tr>
</tbody>
</table>
A forwarding table (my laptop)

deemer@ceres ~ % ip route

10.3.128.1 dev wlp2s0
10.3.128.0/18 dev wlp2s0 proto dhcp scope link src 10.3.135.44 metric 3003
172.18.0.0/16 dev docker0 proto kernel scope link src 172.18.0.1
192.168.1.0/24 dev enp0s31f6 proto kernel scope link src 192.168.1.1

Let's break this down further...
Example: Different networks on my work

Each interface is connected to a different network, i.e., its own IP address.

- Ethernet
  - IP address
  - Subnet mask
  - Default gateway
  - Example 1:
    - IP: 192.168.0.2
    - Ethernet

- Wi-Fi
  - IP address
  - Subnet mask
  - Default gateway
  - Example 2:
    - IP: 10.3.123.1

Routing based on networks

The forwarding table makes decisions based on the network part of the address.

Could we possibly have one entry per host? No!

Only the largest routers will have big tables—others will only know about a few networks.

⇒ Scalability
A routing table

R6#sh ip ro

Gateway of last resort is 108.34.215.1 to network 0.0.0.0

S* 0.0.0.0/0 [1/0] via 108.34.215.1
10.0.0.0/8 is variably subnetted, 7 subnets, 3 masks
C 10.1.0.0/24 is directly connected, wlan-ap0
L 10.1.0.2/32 is directly connected, wlan-ap0
O IA 10.1.44.0/24 [110/1001] via 10.20.30.33, 3w4d, Tunnel0
C 10.1.48.0/24 is directly connected, Loopback0
L 10.1.48.1/32 is directly connected, Loopback0
C 10.20.30.0/24 is directly connected, Tunnel0
L 10.20.30.32/32 is directly connected, Tunnel0
108.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C 108.34.215.0/24 is directly connected, GigabitEthernet0/0
L 108.34.215.208/32 is directly connected, GigabitEthernet0/0
172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
C 172.16.98.0/24 is directly connected, Vlan98
L 172.16.98.1/32 is directly connected, Vlan98
172.17.0.0/16 is variably subnetted, 6 subnets, 3 masks
O IA 172.17.44.0/24 [110/1001] via 10.20.30.33, 3w4d, Tunnel0
C 172.17.48.0/24 is directly connected, Vlan20

This table also has entries for this router's own IPs.
### A routing table

```
R6#sh ip ro
Codes:  L - local,  C - connected,  S - static,  R - RIP,  M - mobile,  B - BGP
       D - EIGRP,  EX - EIGRP external,  O - OSPF,  IA - OSPF inter area
       N1 - OSPF NSSA external type 1,  N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1,  E2 - OSPF external type 2
       i - IS-IS,  su - IS-IS summary,  L1 - IS-IS level-1,  L2 - IS-IS level-2
       ia - IS-IS inter area,  * - candidate default,  U - per-user static route
       o - ODR,  P - periodic downloaded static route,  H - NHRP,  l - LISP
       + - replicated route,  % - next hop override

Gateway of last resort is 108.34.215.1 to network 0.0.0.0

S*  0.0.0.0/0 [1/0] via 108.34.215.1
    10.0.0.0/8 is variably subnetted, 7 subnets, 3 masks
C  10.1.0.0/24 is directly connected, wlan-ap0
L  10.1.0.2/32 is directly connected, wlan-ap0
O IA 10.1.44.1/32 [110/1001] via 10.20.30.33, 3w4d, Tunnel0
C  10.1.48.0/24 is directly connected, Loopback0
L  10.1.48.1/32 is directly connected, Loopback0
```
A large table

rviews@route-server.ip.att.net>show route table inet.0 active-path

inet.0: 866991 destinations, 13870153 routes (866991 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

0.0.0.0/0  *[Static/5] 5w0d 19:43:09
> to 12.0.1.1 via em0.0

1.0.0.0/24  *[BGP/170] 1d 10:24:47, localpref 100, from 12.122.83.23
    AS path: 7018 3356 13335 I, validation-state: valid
> to 12.0.1.1 via em0.0

1.0.4.0/22  *[BGP/170] 1d 10:24:47, localpref 100, from 12.122.83.23
    AS path: 7018 3356 4826 38803 I, validation-state: valid
> to 12.0.1.1 via em0.0

1.0.4.0/24  *[BGP/170] 1d 10:24:47, localpref 100, from 12.122.83.23
    AS path: 7018 3356 4826 38803 I, validation-state: valid
> to 12.0.1.1 via em0.0

1.0.5.0/24  *[BGP/170] 1d 10:24:47, localpref 100, from 12.122.83.23
    AS path: 7018 3356 4826 38803 I, validation-state: valid
> to 12.0.1.1 via em0.0

1.0.6.0/24  *[BGP/170] 1d 10:24:47, localpref 100, from 12.122.83.23
    AS path: 7018 3356 4826 38803 I, validation-state: valid
> to 12.0.1.1 via em0.0
How does forwarding actually work?
The IPv4 Header

<table>
<thead>
<tr>
<th>0</th>
<th>4</th>
<th>8</th>
<th>16</th>
<th>31 bit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version</td>
<td>IHL</td>
<td>TOS</td>
<td>Total length</td>
<td>Identification</td>
</tr>
<tr>
<td>TTL</td>
<td>Protocol</td>
<td>Header checksum</td>
<td>Source address</td>
<td>Destination address</td>
</tr>
<tr>
<td>Options</td>
<td></td>
<td></td>
<td>Up to 65536 bytes</td>
<td></td>
</tr>
<tr>
<td>Data</td>
<td></td>
<td></td>
<td>0-40 bytes</td>
<td>20 bytes</td>
</tr>
</tbody>
</table>
The IPv4 Header

Defined by RFC 791
RFC (Request for Comment): defines network standard
Most Important fields

- **Version**: 4 for IPv4 packets, 6 for IPv6
- **Source address**: where the packet came from
- **Destination address**: where the packet is going

(use this to decide how to forward)

(continued…)
More important fields

- **TTL (time-to-live):** decremented each hop
  - Can prevent forwarding loops (and do other stuff…)
- **Checksum:** computed over header (very weak!)
- **Protocol identifier:** describes what’s in the packet
  - 6: TCP, 17: UDP, 1: ICMP, …
  - Defines the type of the payload
Less important fields

- Header length: in 32-bit units
  - >5 implies use of IP options
  - Almost all routers ignore IP options
Less important fields

- **Header length**: in 32-bit units
  - >5 implies use of IP options
  - Almost all routers ignore IP options
- **Fragmentation**
  - Network can fragment a packet if next link requires a small frame
  - Most routers don’t fragment (or reassemble fragments)

> **Most Internet packets** <1500 bytes
Less important fields

- **Header length:** in 32-bit units
  - >5 implies use of IP options
  - Almost all routers ignore IP options
- **Fragmentation**
  - Network can fragment a packet if next link requires a small frame
  - Most routers don’t fragment (or reassemble fragments)
- **We won’t talk about…**
  - **Type of Service (TOS):** basic traffic classification
  - **Identifier:** might have special meaning on some networks
Forwarding steps

What does a router do when it receives a packet?
IP Forwarding: Steps

What do you do when you get a packet?

1. Is the packet valid?
   - Is check sum valid? => If no, drop
   - Is IP < 0? => Drop

2. Is it for me? => If dest IP == (any local IP)
   => Send it to OS
   - Check if dest addr is for this host.
   - If yes, send "upstream" to OS. => OS considers protocol field: TCP, UDP ...

3. Is it for a local network? => Consider dest IP
   - Does packet match any networks in forwarding table?
   - If yes, send on that interface.

4. Do I have a next hop? => If so, lookup interface for next hop IP
Looking at it in a different way.

1. Is it valid?
   - No -> Drop
   - Yes -> Check forwarding table

2. My IP?
   - Send to OS

3. Local interface?
   - Send to that interface

4. Next hop?
   - Next hop?

To send:
- Decrement TTL
- Recompute checksum
- Send on interface
Forwarding mechanics

When an IP packet arrives at a host/router:

- **Is it valid?** Verify checksum over header
- **Is it for me?** If dest IP == your address, send to OS
- **If not, where should it go?**
  - Consult forwarding table => find next hop
  - Decrement TTL
  - Send packet to next hop
Traceroute

• When TTL reaches 0, router may send back an error
  – ICMP TTL exceeded
• If it does, we can identify a path used by a packet!

\[ A \xrightarrow{1} B \xrightarrow{2} \text{error} \xrightarrow{3} \text{error} \]

\[ B \xrightarrow{1} \text{error} \xrightarrow{2} \text{error} \xrightarrow{3} \text{error} \]
Traceroute example

[deemer@Warsprite ~]$ traceroute -q 1 google.com
traceroute to google.com (142.251.40.174), 30 hops max, 60 byte packets
 1  router1-nac.linode.com (207.99.1.13)  0.621 ms
 2  if-0-1-0-0-0.gw1.cjj1.us.linode.com (173.255.239.26)  0.499 ms
 3  72.14.222.136 (72.14.222.136)  0.949 ms
 4  72.14.222.136 (72.14.222.136)  0.919 ms
 5  108.170.248.65 (108.170.248.65)  1.842 ms
 6  lga25s81-in-f14.1e100.net (142.251.40.174)  1.812 ms
Trace route example

[deemer@Warsprite ~]$ traceroute -q 1 amazon.co.uk
traceroute to amazon.co.uk (178.236.7.220), 30 hops max, 60 byte packets
1 router2-nac.linode.com (207.99.1.14) 0.577 ms
2 if-11-1-0-1-0.gw2.cjj1.us.linode.com (173.255.239.16) 0.461 ms
3 ix-et-2-0-2-0.tcore3.njy-newark.as6453.net (66.198.70.104) 1.025 ms
4 be3294.ccr41.jfk02.atlas.cogentco.com (154.54.47.217) 2.938 ms
5 be2317.ccr41.lon13.atlas.cogentco.com (154.54.30.186) 69.725 ms
6 be2350.rcr21.b023101-0.lon13.atlas.cogentco.com (130.117.51.138) 69.947 ms
7 a100-row.demarc.cogentco.com (149.11.173.122) 71.639 ms
8 150.222.15.28 (150.222.15.28) 78.217 ms
9 150.222.15.21 (150.222.15.21) 84.383 ms
10 *
11 150.222.15.4 (150.222.15.4) 74.529 ms
12 * ...
Coming up…

• ARP: Mapping IPs to MAC addresses
• How are addresses assigned?
• NAT: When it gets complicated
• Routing algorithms: how to build forwarding tables