CSCI1680
Network Layer: IP & Forwarding

Nick DeMarinis

Based partly on lecture notes by Rodrigo Fonseca, David Mazières, Phil Levis, John Jannotti
Administivia

• Snowcast: due Wednesday (9/27)

• IP project: out Thursday, fill out group preference form by Thursday 11:59pm

• HW1: out later today, due next Thurs
  – Some practice for IP!
Start of network layer
• Network layer: Internet Protocol (IP) (v4)
• Mechanics of IP forwarding
• Intro to IP project
Last time: link-layer

- Hosts connect to network via interfaces
- Every interface has a link-layer address
  Ethernet/Wifi: MAC address (0c:45:22:c1:be:03)
- Switches, Wifi APs: in-network devices that forward packets between nodes

Mental model for the link layer
=> How to connect hosts on a “small” network
=> Given link-layer address, know how to reach host on your network
Layers, Services, Protocols

Application
Service: user-facing application. Application-defined messages

Transport
Service: multiplexing applications
Reliable byte stream to other node (TCP),
Unreliable datagram (UDP)

Network
Service: move packets to any other node in the network
Internet Protocol (IP)

Link
Service: move frames to other node across link.
May add reliability, medium access control

Physical
Service: move bits to other node across link
Internet Protocol (IP) Goals

How to connect everyone?
• Glue lower-level networks together
• A network of networks!
• Router: device that forwards packets between networks

=> Doesn’t that sound like switching?
Map of the Internet, 2021 (via BGP)
OPTE project
New Challenges

- Networks are heterogeneous (e.g., Wifi vs. Ethernet)
  - Different frame formats
  - Some are more reliable than others
  - Different packet sizes/bandwidths
- Scaling: link-layer strategies don’t work!
What came before the Internet?
At the time the internet was forming (~1960s), landline phones were a big network that connected lots of end-users (160M worldwide)

Why should we care?
⇒ Provides a useful comparison
⇒ Understand some of the design goals of the time
Early telephone networks

Used circuit switching: set up whole path for call beforehand
• Does it scale?
A Bit of History

Early Packet switched networks: Arpanet’s IMPs
- Late 1960’s => RFC 1, 1969!
- Reliable network with many features we know today

• Build apps with Network Control Program (NCP)
  - Built on reliable IMPs
  - Used by programs like telnet, mail, file transfer

What about when network isn’t reliable?
THE ARPA NETWORK
DEC 1969
Abb. 4 ARPA NETwork, topologische Karte. Stand Juni 1974.
How to make such a protocol?

- How to deal with heterogeneous networks?
- How to find hosts?
- Should messages be reliable or unreliable?
- What to do when a device joins/leaves?
- ...

Big concerns
- Not every application needs all features
- Can’t assume much functionality from (heterogeneous link layer)
1974: TCP/IP Introduced

• Vint Cerf, Robert Kahn build protocol to replace NCP
• Initial design: single protocol providing a reliable pipe

• Eventually, separated into different protocols we know today
  – IP: basic datagram service among hosts
  – TCP: reliable transport
  – UDP: unreliable *multiplexed* datagram service
IP’s Decisions

• Connectionless, packet-switched network

• “Best-effort” service: other layers add reliability if you need it

How to reach hosts?
• Common message format: IP header
• Every host identified by an IP address
IP’s Decisions

• Connectionless, packet-switched network
  => Routers are “simple” => no connection state

• “Best-effort” service: other layers add reliability if you need it
  => Packets might be dropped, reordered, delayed, …

How to reach hosts?

• Common message format: IP header
• Every host identified by an IP address
David D. Clark, “The design Philosophy of the DARPA Internet Protocols”, 1988

• Primary goal: multiplexed utilization of existing interconnected networks

• Other goals:
  – Communication continues despite loss of networks or gateways
  – Support a variety of communication services
  – Accommodate a variety of networks
  – Permit distributed management of its resources
  – Be cost effective
  – Low effort for host attachment
  – Resources must be accountable
The Internet Protocol

IP runs on all hosts and routers

- **Addressing**: how we name nodes in an IP network
- **Provides forwarding**: how routers move packets based on the destination address
- **(later) Routing**: how routers build forwarding rules
IP Addressing
Overview

• Unique number to identify “all” hosts on the Internet

• A number with structure => the number tells the network where the host is
Analogy: back to phones

Telephone numbers have a structure to them

+ 1 401 863 1000
Analogy: back to phones

Telephone numbers have a structure to them

+1 401 863 1000

+1 212 555 4253

Part of the number tells where you are! (or at least it did before cell phones)
... and not all numbers are the same length!
IP Addressing

IP Version 4: Each address is a 32-bit number:

128.148.16.7

10000000 10010100 00010000 00000111

Notation
• Write each byte ("octet") as a decimal number
• This is called "dotted decimal" or "dotted quad" notation

32 bits => $2^{32}$ possible addresses… problem?
IP Addressing

An IP address identifies...

- *Who* a host is: A unique number
- *Where* it is on the Internet
- Networks are allocated ranges of IPs by global authority (ICANN)
  - Further subdivided by regions, ISPs, ...  
- Some IPs have special uses (eg. 127.0.0.1)


*ICANN (Internet Corporation for Assigned Names and Numbers)*
IP Addressing

Brown owns the range:

128.148.xxx.xxx

10000000 10010100 xxxxxxxxx xxxxxxxxx

Network part
Identifies Brown (to the Internet)

Host part
Denotes individual hosts within the Brown Network
A typical configuration

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

A network can designate IP addresses for its own hosts within its address range

For 128.148.xxx.xxx:

10000000 10010100 xxxxxxxxxx xxxxxxxxxx

Brown uses the prefix 128.148.0.0/16

Some other ways to write this:

128.148/16
128.148.0.0 + subnet mask 255.255.0.0
Common prefixes

<table>
<thead>
<tr>
<th>Prefix</th>
<th>Binary</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.2.0.0/16</td>
<td>00000001 00000010 xxxxxxxxxx xxxxxxxxxx</td>
</tr>
<tr>
<td>8.0.0.0/8</td>
<td>00001000 xxxxxxxxxx xxxxxxxxxx xxxxxxxxxx</td>
</tr>
<tr>
<td>123.10.1.0/24</td>
<td>01111011 00001010 00000001 xxxxxxxxxx</td>
</tr>
<tr>
<td>201.112.10.200/30</td>
<td>11001001 01110000 00001010 110010xx</td>
</tr>
</tbody>
</table>
Example

How many addresses are in the network 1.1.0.0/20?
How do we move packets between networks?
How IP forwarding works

Assume:

• Communicating on same network is easy—this is the link-layer’s job!
• Can map IP addresses to MAC addresses (more on this later)

How to reach an address outside this network?

Send packets to a router, which forwards IP packets to other networks
Forwarding IP packets

To more networks (i.e., Internet)

Src: 1.2.1.3
Dst: 1.2.2.100
. . .
Forwarding IP packets

Src: 1.2.1.3  
Dst: 1.2.2.100  
. . .  

To more networks (ie, Internet)
Forwarding IP packets

<table>
<thead>
<tr>
<th>Prefix</th>
<th>Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.2.1.2</td>
<td>IF1</td>
</tr>
<tr>
<td>1.2.1.3</td>
<td>IF0</td>
</tr>
<tr>
<td>1.2.1.200</td>
<td>IF2</td>
</tr>
<tr>
<td>1.2.2.100</td>
<td></td>
</tr>
<tr>
<td>1.2.2.105</td>
<td></td>
</tr>
</tbody>
</table>
Forwarding IP packets

Src: 1.2.1.3
Dst: 1.2.2.100

To more networks (ie, Internet)

Prefix | Interface
--- | ---
1.2.1.0/24 | IF1
1.2.2.0/24 | IF2
<everything else> | (IF0)
Wait, what happens at the link layer?
What about the rest?

How to reach networks that aren’t directly connected?

<table>
<thead>
<tr>
<th>Prefix</th>
<th>Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.2.1.0/24</td>
<td>IF1</td>
</tr>
<tr>
<td>1.2.2.0/24</td>
<td>IF2</td>
</tr>
<tr>
<td>&lt;everything else&gt;</td>
<td>IF0</td>
</tr>
</tbody>
</table>

To more networks (ie, Internet)
What about the rest?

- Need “next hop” IP: another router that knows about other networks
  - How to reach it? Check table again!
- “Default gateway”: where to send to reach anything not in the table

<table>
<thead>
<tr>
<th>Prefix</th>
<th>IF/Next hop</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.2.1.0/24</td>
<td>IF1</td>
</tr>
<tr>
<td>1.2.2.0/24</td>
<td>IF2</td>
</tr>
<tr>
<td>8.0.0.0/30</td>
<td>IF0</td>
</tr>
<tr>
<td>128.148.0.0/16</td>
<td>1.2.1.5</td>
</tr>
<tr>
<td>Default</td>
<td>8.0.0.2</td>
</tr>
</tbody>
</table>
The forwarding table

Exploits hierarchical structure of addresses: know how to reach networks, not individual hosts

- Table is keyed is a network prefix, not a whole address
- Select best prefix with *longest prefix matching* (more on this later)

<table>
<thead>
<tr>
<th>Prefix</th>
<th>IF/Next hop</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.2.1.0/24</td>
<td>IF1</td>
</tr>
<tr>
<td>1.2.2.0/24</td>
<td>IF2</td>
</tr>
<tr>
<td>8.0.0.0/30</td>
<td>IF0</td>
</tr>
<tr>
<td>128.148.0.0/16</td>
<td>1.2.1.5</td>
</tr>
<tr>
<td>Default</td>
<td>8.0.0.2</td>
</tr>
</tbody>
</table>
A forwarding table

```
# ip route
127.0.0.0/8 via 127.0.0.1 dev lo
172.17.44.0/24 dev enp7s0 proto kernel scope link src 172.17.44.22 metric 204
default via 172.17.44.1 dev eth0 src 172.17.44.22 metric 204
```
How do we move packets between networks?
IP forwarding

Given a packet, decide where to send it

<table>
<thead>
<tr>
<th>Prefix</th>
<th>Interface/Next hop</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

50
A forwarding table (my laptop)

deeemer@ceres ~ % ip route default via 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
Forwarding: examples

<table>
<thead>
<tr>
<th>Prefix</th>
<th>Interface/Next hop</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Routing based on networks
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 10.1.44.1/32 [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
C  10.20.30.32/31 is directly connected, Tunnel0
L  10.20.30.32/32 is directly connected, Tunnel0
C  108.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
L  108.34.215.0/24 is directly connected, GigabitEthernet0/0
C  108.34.215.208/32 is directly connected, GigabitEthernet0/0
C  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
C  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
L  172.17.48.1/32 is directly connected, Vlan20
C  172.17.49.0/25 is directly connected, Vlan50
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

<table>
<thead>
<tr>
<th>Prefix</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>S* 0.0.0.0/0</td>
<td>[1/0]</td>
<td>via 108.34.215.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10.0.0.0/8 is</td>
<td>variably subnetted, 7 subnets, 3 masks</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C 10.1.0.0/24 is directly connected, wlan-ap0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>L 10.1.0.2/32 is directly connected, wlan-ap0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>O IA 10.1.44.1/32 [110/1001] via 10.20.30.33, 3w4d, Tunnel0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C 10.1.48.0/24 is directly connected, Loopback0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>L 10.1.48.1/32 is directly connected, Loopback0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C 10.20.30.32/31 is directly connected, Tunnel0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>L 10.20.30.32/32 is directly connected, Tunnel0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IP Address</td>
<td>Route Details</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------------</td>
<td>-------------------------------------------------------------------------------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.0.0.0/0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0.0.0/24</td>
<td>*[BGP/170] 1d 10:24:47, localpref 100, from 12.122.83.238, AS path: 7018 3356 4826 38803 I, validation-state: valid</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0.4.0/22</td>
<td>*[BGP/170] 1d 10:24:47, localpref 100, from 12.122.83.238, AS path: 7018 3356 4826 38803 I, validation-state: valid</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0.4.0/24</td>
<td>*[BGP/170] 1d 10:24:47, localpref 100, from 12.122.83.238, AS path: 7018 3356 4826 38803 I, validation-state: valid</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0.5.0/24</td>
<td>*[BGP/170] 1d 10:24:47, localpref 100, from 12.122.83.238, AS path: 7018 3356 4826 38803 I, validation-state: valid</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0.6.0/24</td>
<td>*[BGP/170] 1d 10:24:47, localpref 100, from 12.122.83.238, AS path: 7018 3356 4826 38803 I, validation-state: valid</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
How does forwarding actually work?
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

(continued…)

59
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

• 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?
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!
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
Traceroute 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
  ...  
Demo: IP project
The IPv4 Header

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

- **Version**: 4 for IPv4 packets, 6 for IPv6
- **Destination address**: used for forwarding
- **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

• 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 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!
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
Traceroute 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
  ...  
Demo: IP project
Coming up...

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

Fill out the group preference survey for the IP project (announcement soon) by tomorrow (Sep 30) by 11:59PM
Putting it all together...

- The more connected a router becomes, the more complex its forwarding table... and the more it may change!

- Routing algorithms: routers exchange path information to their forwarding tables (more on this later)
Goal: find the most specific (ie, longest) prefix matching the destination

How to reach 1.2.2.100?

1.2.2.100  00000001.00000010.00000010.01101000
1.2.1.0/24  00000001.00000010.00000001.xxxxxxxx
1.2.2.0/24  00000001.00000010.00000010.xxxxxxxx
0.0.0.0/0  x.x.x.x.x.x.x.x.x.x.x.x.x.x

Output: IF2

Longest Prefix Matching (LPM): can represent entire IP space in (small) table!
Some ISP

<table>
<thead>
<tr>
<th>Prefix</th>
<th>Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>128.148.0.0/16</td>
<td>IF1</td>
</tr>
<tr>
<td>1.3.0.0/16</td>
<td>IF2</td>
</tr>
<tr>
<td>5.6.128.0/20</td>
<td>IF3</td>
</tr>
<tr>
<td>128.148.100.0/24</td>
<td>IF4</td>
</tr>
<tr>
<td>0.0.0.0/0</td>
<td>8.0.0.2</td>
</tr>
</tbody>
</table>

Brown
128.148.0.0/16

Customer 2
1.3.0.0/16

Customer 3
5.6.128.0/20

Brown’
128.148.100.0/24

Dst: 128.148.105.207

Dst: 128.148.100.104

8.0.0.0/30
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 10.1.44.1/32 [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
C   10.20.30.32/31 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
C   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
C   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
L   172.17.48.1/32 is directly connected, Vlan20
C   172.17.49.0/25 is directly connected, Vlan50
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
C  10.20.30.32/31 is directly connected, Tunnel0
L  10.20.30.32/32 is directly connected, Tunnel0
## show route table inet.0 active-path

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

<table>
<thead>
<tr>
<th>Route</th>
<th>Prefix</th>
<th>Path Description</th>
<th>Validation State</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0.0.0/0</td>
<td>*[Static/5] 5w0d 19:43:09</td>
<td>&gt; to 12.0.1.1 via em0.0</td>
<td>valid</td>
</tr>
<tr>
<td>1.0.0.0/24</td>
<td>*[BGP/170] 1d 10:24:47, localpref 100, from 12.122.83.238</td>
<td>AS path: 7018 3356 13335 I</td>
<td>valid</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&gt; to 12.0.1.1 via em0.0</td>
<td></td>
</tr>
<tr>
<td>1.0.4.0/22</td>
<td>*[BGP/170] 1d 10:24:47, localpref 100, from 12.122.83.238</td>
<td>AS path: 7018 3356 4826 38803 I</td>
<td>valid</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&gt; to 12.0.1.1 via em0.0</td>
<td></td>
</tr>
<tr>
<td>1.0.4.0/24</td>
<td>*[BGP/170] 1d 10:24:47, localpref 100, from 12.122.83.238</td>
<td>AS path: 7018 3356 4826 38803 I</td>
<td>valid</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&gt; to 12.0.1.1 via em0.0</td>
<td></td>
</tr>
<tr>
<td>1.0.5.0/24</td>
<td>*[BGP/170] 1d 10:24:47, localpref 100, from 12.122.83.238</td>
<td>AS path: 7018 3356 4826 38803 I</td>
<td>valid</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&gt; to 12.0.1.1 via em0.0</td>
<td></td>
</tr>
<tr>
<td>1.0.6.0/24</td>
<td>*[BGP/170] 1d 10:24:47, localpref 100, from 12.122.83.238</td>
<td>AS path: 7018 3356 4826 38803 I</td>
<td>valid</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&gt; to 12.0.1.1 via em0.0</td>
<td></td>
</tr>
</tbody>
</table>
How to avoid loops?

**TTL (Time to Live):** Decrement by 1 at each hop, send back error at 0

**traceroute:** tool to send packets with increasing TTLs

=> can learn about network paths!