CSSE2310/7231 — B.1

Networking continued

IP Headers

Things from IP header

- Packet length
  - 2^{16} bytes including header
- Protocol
  - Which transport layer protocol should get this message?
- TTL
  - Reduced each time the packet reaches an interface
  - Packet is dropped if TTL reaches 0

ping

1. Send a message to a device
2. (hopefully) it sends a copy back
3. Calculate travel time

traceroute

Demo

- Send out a packet with TTL=1 from N_0
- Packet reaches N_1
- Packet is dropped
- N_1 sends an ICMP message back to N_0
  - N_1: "Just thought you should know ..."
  - N_0: "Oh how surprising"
  - N_0 now knows N_1's address
- Send out a packet with TTL=2 from N_0
- Packet travels through N_1
- Packet reaches N_2
- Packet is dropped
- ...

IPv4 structure

- v4 Addresses = 32 bits
- Divided into network and host parts
  - Network comes part starts with the most significant bit
  - eg: moss is `130.102.72.9`
  - `130.102 | 72.9`
  - `10000010.01100100 | 01001000.00001001`
- UQ public addresses look like `130.102.?.?`

“subnet size”

- Increasing the number of bits in the network part means a “smaller” network.
- eg 16 bits for the network part means 16 bits for host addresses = 65536 possible host addresses.
- 18 bit network part would allow more networks but each network has “only” 16384 possible host addresses.

Routing?

When sending a message, the network layer needs to make a decision:

1. Send direct to the destination
   - Find the MAC of the destination
2. Send via another machine
   - Find the MAC of the intermediary

Option #1 will only work if the destination is directly reachable at Layer 2.

subnets

- An organisation's network can be divided into subnets.
- A host can directly communicate with everything on the same subnet.
- Broadcasts will reach all hosts in the subnet.

For the rest of this discussion, we'll use network and subnet interchangably.

To communicate, a host needs to know both its IP address and which (sub)network it belongs to.
Can describe the subnet in two ways:

- CIDR[1] notation
- subnet mask

[1] Classless Inter-Domain Routing

Method 1 — CIDR

eg 130.102.0.0 / 16

- Set all host bits to 0
- The value after / is how many bits are in the network part

130.102.12.0 / 24

- Subnet of all addresses starting with 130.102.12.

CIDR

/x networks, x does not need to fall on a byte boundary (/8, /16, /24)
These describe different networks

- 130.102.12.0 / 24 = "roughly" 254 host addresses
- 130.102.12.0 / 23 = "roughly" 510 host addresses

/24 $\Rightarrow$ 130.102.00001100.????????
/23 $\Rightarrow$ 130.102.0000110?.????????

So 130.102.00001101.00000110 ==
130.102.13.6 belongs to 130.102.12.0 / 23
but not 130.102.12.0 / 24

“Roughly?”

Each subnet will have two addresses reserved:

- All host bits = zero (minimum host address)
  - “network address”
- All host bits = one (maximum host address)
  - “broadcast address”

So subnet `A.B.C.D / x` has $32 - x$ host bits and $2^{32-x} - 2$ usable host addresses[2].

---

[2] /31 is a special case

Method 2 — netmask

A netmask = a bit pattern which will map[3] any IP address to the corresponding network address.

1. Set all network bits to 1.
2. Set all host bits to 0.

For example: / 24
Mask would be 255.255.255.0

- 130.102.24.17 → 130.102.24.0
- 130.102.24.250 → 130.102.24.0
- 130.102.21.16 → 130.102.21.0

[3] under bitwise AND

Example

130.102.160.0 / 20 ($160 = 128 + 32$)
130.102.10100000.00000000
Network bits are:
130.102.1010 0000.00000000
netmask:
255.255.1111 0000.00000000
So netmask is 255.255.240.0

- 130.102.163.19 $\rightarrow$130.102.160.0 — yes
- 130.102.171.99 $\rightarrow$130.102.160.0 — yes
- 130.102.176.14 $\rightarrow$130.102.176.0 — no

Valid?

Which of the following are[4] valid netmasks?

- 255.255.255.192? $192 = 128 + 64$
- 255.208.0.0? $208 = 128 + 64 + 0 + 16$
- 224.0.0.0? $224 = 128 + 64 + 32$

[4] Theoretically

Exercise

What is the broadcast address for use by:
117.98.141.19 netmask=255.254.0.0?

Netmask tells us that the network is:
117.98.0.0/15
01110101.01100010.00000000.00000000/15

Setting the $32 - 15 = 17$ least significant bits to 1 gives:
01110101.0110001 0.00000000.00000000/15
01110101.0110001 1.11111111.11111111/15
= 117.99.255.255

Exercise

Give the CIDR form and netmask for the largest network which

- Includes:
  - 100.89.19.80
  - 100.89.19.82
- Does not include:
  - 100.89.19.97

yes	100.89.19.80	01100100.01011001.00010011.01 0 10000
yes	100.89.19.82	01100100.01011001.00010011.01 0 10010
no	100.89.19.97	01100100.01011001.00010011.01 1 00001

- So 100.89.19.80 / 27 is as big as possible without including 97.
- Netmask = 255.255.255.224

Special networks

(From RFC 6890).

non-routable / "link local" addresses

Addresses from the following networks should not be used on the public internet:

- 10.0.0.0/8
- 172.16.0.0/12
- 192.168.0.0/16
- 169.254.0.0/16
  - For auto config when you can't get a real address

Special networks

All addresses in 127.0.0.0/8 are "loopback" addresses:

- Including but not limited to 127.0.0.1
- Yes, that's $2^{24} - 2$ addresses
- ... what?

Request For Comment

RFCs describe critical protocols for the internet and are publically available.

eg: `http://tools.ietf.org/html/rfc1178`

- SSH #4253 (and others)
- HTTP/1.1 #7230 (and others)

Request For Comment

Not all are of uniform importance:

- Chosing a name for your computer #1178
- IP over Avian carriers #1149
- ... with Quality of Service #2549
- ... for IPv6 #6214

NAT — overview

- Host X=10.0.20.15 wants to connect to address G (on the public internet).
  - Address information will be: {src-ip=X, src-port=sp, dest-ip=G, dest-port=80}
- Packet arrives at G.
- G tries to reply, with:
  - {src-ip=G, src-port=80, dest-ip=X, dest-port=sp}
- Reply doesn't go anyhere because nobody knows where X is.

NAT

NAT = Network Address Translation

1. X → ... →R → ... → G
   {src-ip=X, src-port=sp, dest-ip=G, dest-port=80}
2. Packet arrives at *R*.
3. *R* modifies address information
   {src-ip=R, src-port=np, dest-ip=G, dest-port=80}
4. ...
5. *G* recieves packet and replies {src-ip=G, src-port=80, dest-ip=R, dest-port=np}
6. ...
7. *R* recieves packet and modifies info: {src-ip=G, src-port=80, dest-ip=X, dest-port=sp}
8. *X* recieves the message

NAT

- This only works because R remembers that port np corresponds to port sp on X
- R does not need to be directly connected to X or G.
  - It needs to be somewhere before the packets with local addresses leaks onto the public internet.