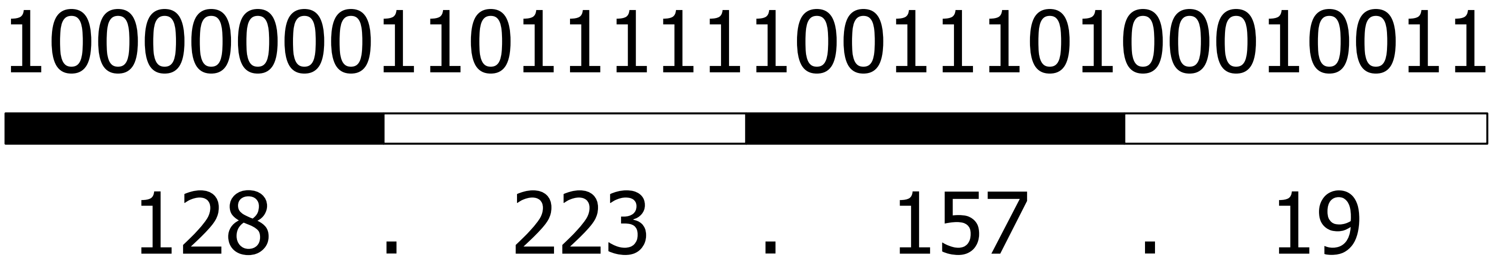
Campus Networking Workshop

IP Addressing and Routing

These materials are licensed under the Creative Commons *Attribution-Noncommercial 3.0 Unported* license
(http://creativecommons.org/licenses/by-nc/3.0/)

IPv4 addresses

- 32-bit binary number
  - How many unique addresses in total?
- Conventionally represented as four dotted decimal octets

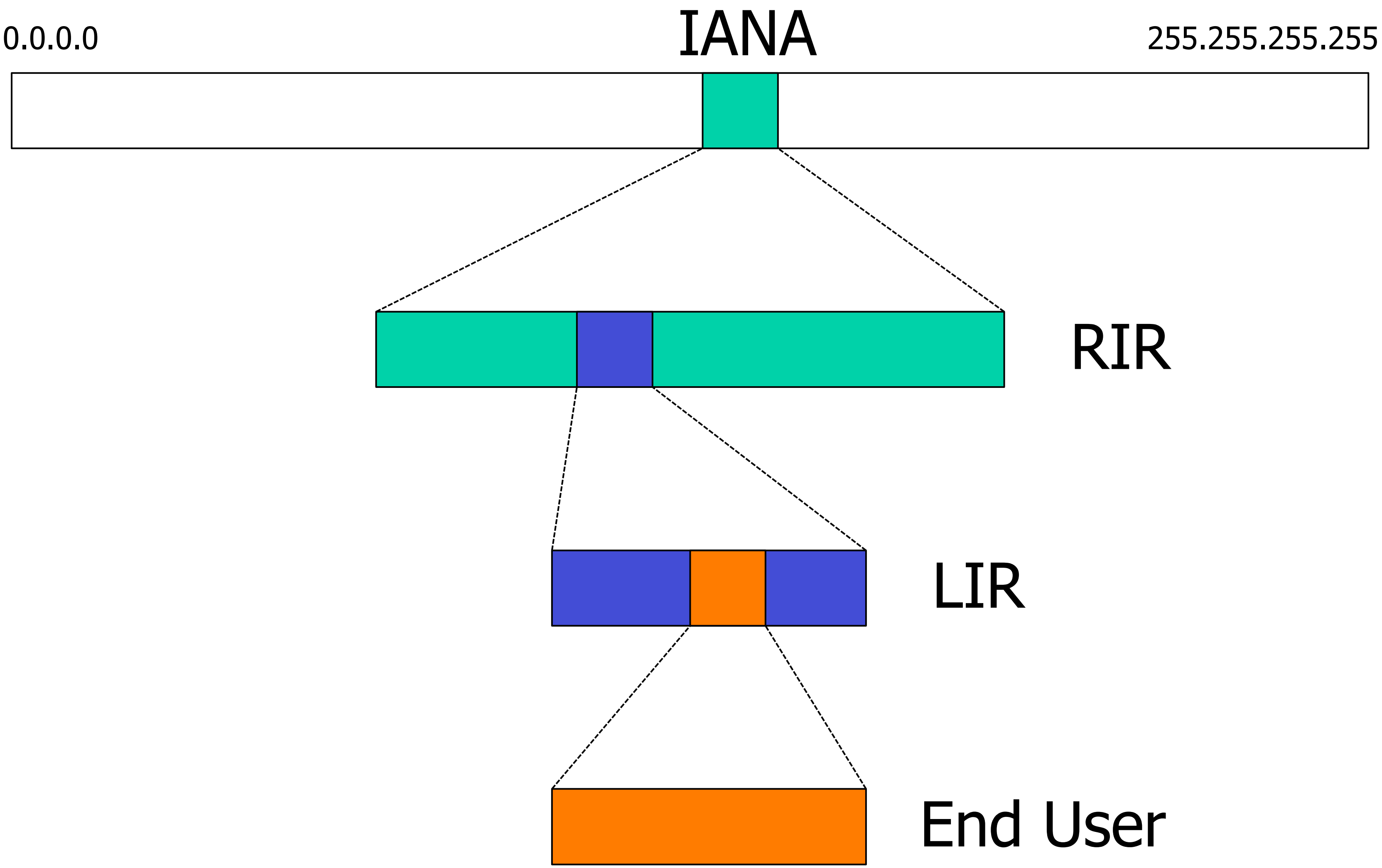


10000000110111111001110100010011

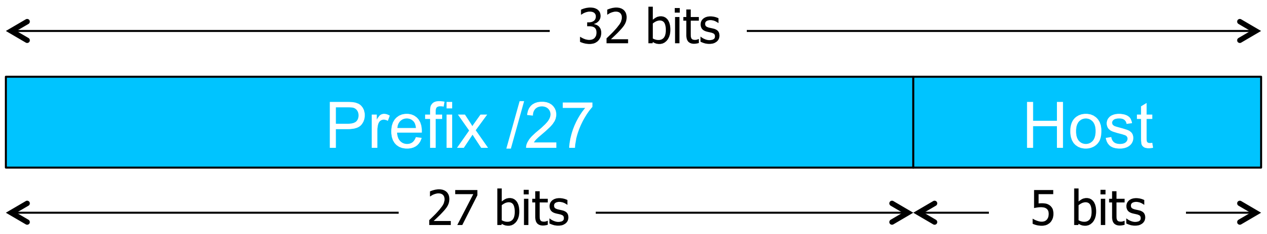
128 . 223 . 157 . 19

Can you explain why 00010011 = 19 in decimal?

Hierarchical address allocation



Prefixes



- A range of IP addresses is given as a *prefix*, e.g. 192.0.2.128/27
- In this example:
  - How many addresses are available?
  - What are the lowest and highest addresses?

Prefix calculation

192 . 0 . 2 . 128

11000000000000000000001010000000

Prefix length /27 ➔ First 27 bits are fixed

Lowest address:

11000000000000000000001010000000

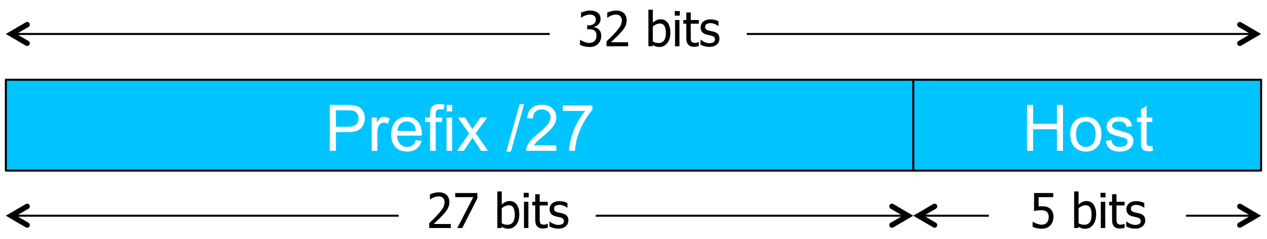
192 . 0 . 2 . 128

Highest address:

11000000000000000000001010011111

192 . 0 . 2 . 159

IPv4 "Golden Rules"



1. All hosts on the same L2 network must share the *same* prefix
2. All hosts with the same prefix have *different* host part
3. Host part of all-zeros and all-ones are reserved

Golden Rules for 192.0.2.128/27

- Lowest 192.0.2.128 = network address
- Highest 192.0.2.159 = broadcast address
- Usable: 192.0.2.129 to 192.0.2.158
- Number of usable addresses: 32 - 2 = 30

Exercises

- Network 10.10.10.0/25
    - How many addresses in total?
    - How many usable addresses?
    - What are the lowest and highest usable addresses?
- Network 10.10.20.0/22
    - How many addresses in total?
    - How many usable addresses?
    - What the the lowest and highest usable addresses?

An edge case

- How many usable addresses in a /30 prefix?
- What is this used for?
    - (Note: modern routers support /31 for this purpose to reduce IP address wastage)

Netmask

- Netmask is just an alternative (old) way of writing the prefix length
- A '1' for a prefix bit and '0' for a host bit
- Hence N x 1's followed by (32-N) x 0's

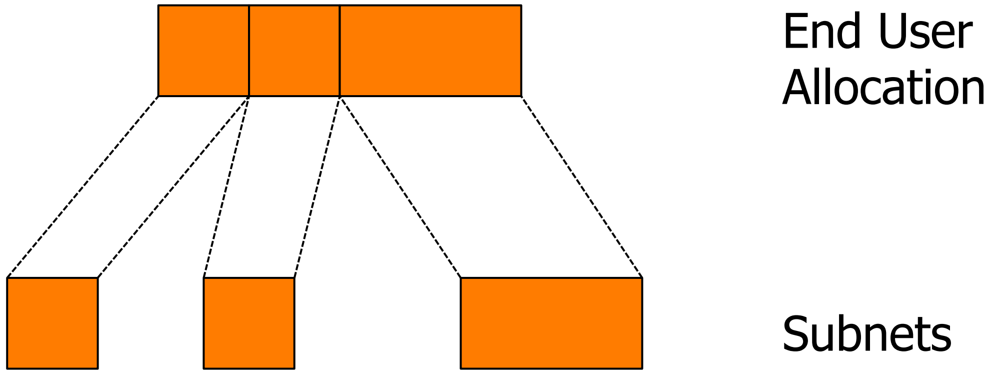
/27 =

11111111111111111111111111100000

255 . 255 . 255 . 224

Subnetting

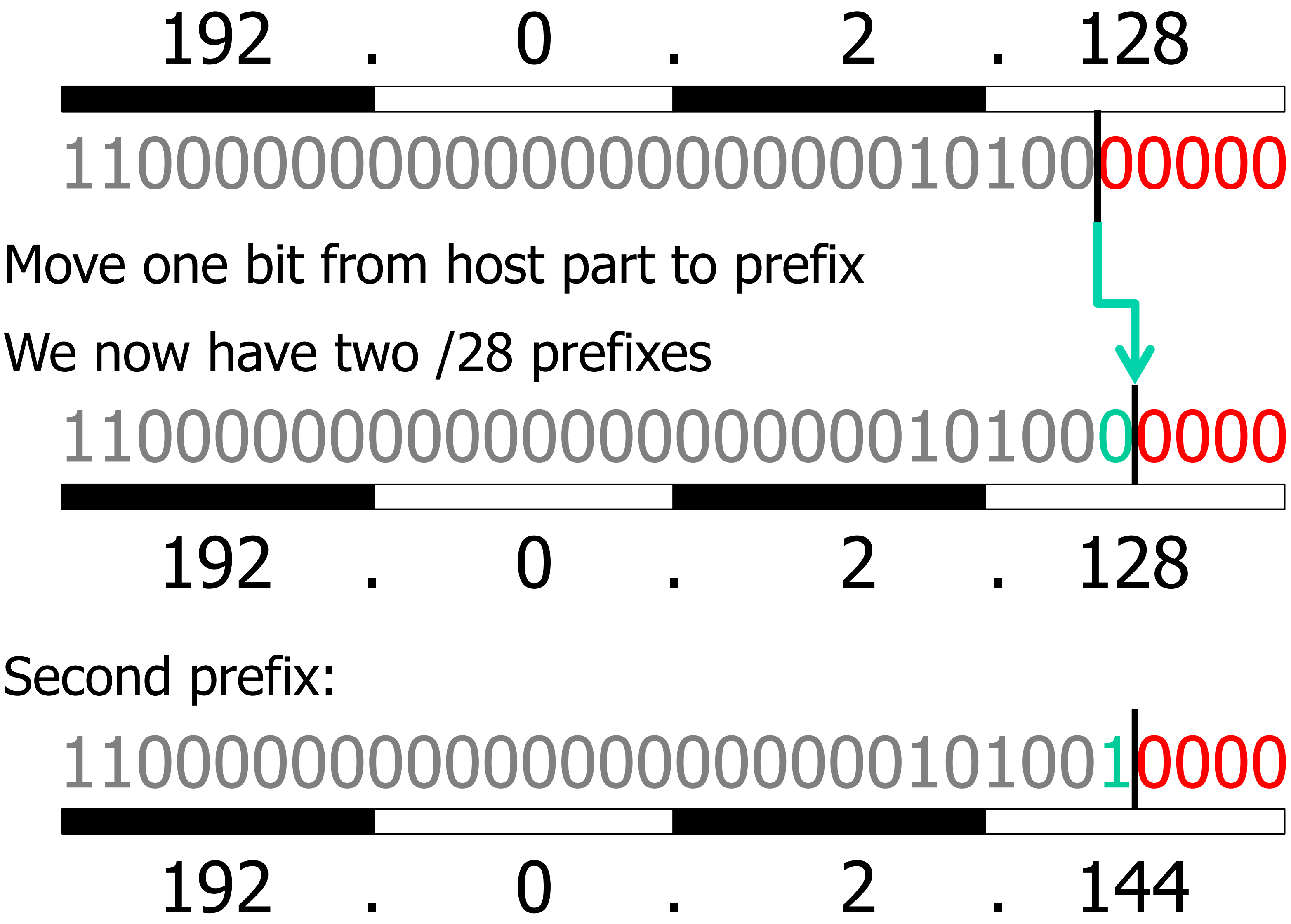
- Since each L2 network needs its own prefix, then if you route more than one network you need to divide your allocation
- Ensure each prefix has enough IPs for the number of hosts on that network



Subnetting Example

- You have been given 192.0.2.128/27
- However you want to build two Layer 2 networks and route between them
- The Golden Rules demand a different prefix for each network
- Let's split this address space into two equal-sized pieces

Subnetting /27



192 . 0 . 2 . 128

110000000000000000000010100 | 00000

Move one bit from host part to prefix

We now have two /28 prefixes

1100000000000000000000101000 | 0000

192 . 0 . 2 . 128

Second prefix:

1100000000000000000000101001 | 0000

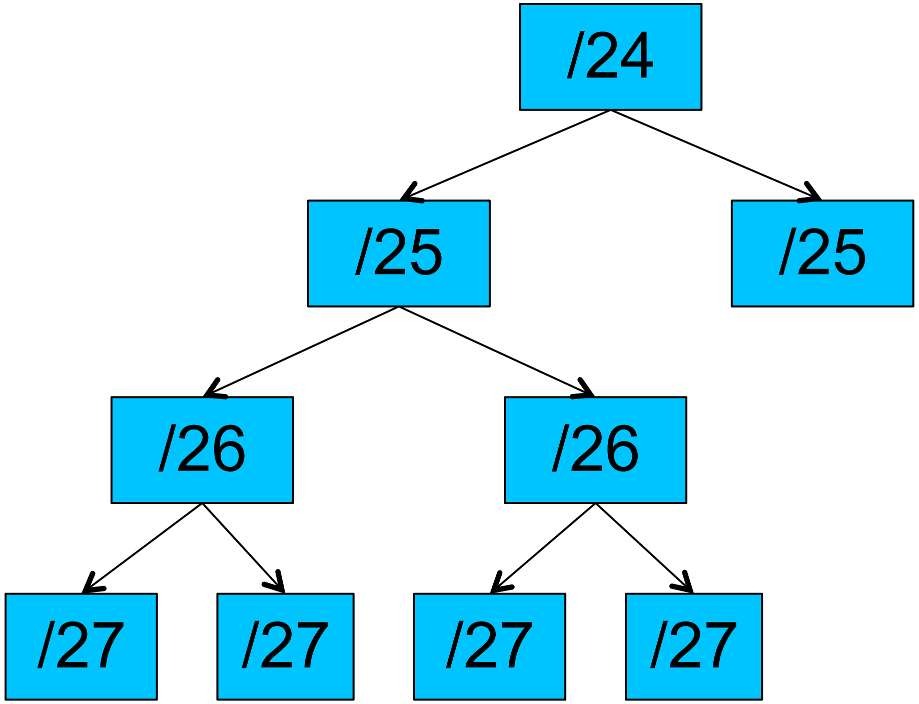
192 . 0 . 2 . 144

Check correctness

- Expand each new prefix into lowest and highest
- Ranges should not overlap
  - 192.0.2.128/28
    - Lowest (network) = 192.0.2.128
    - Highest (broadcast) = 192.0.2.143
  - 192.0.2.144/28
    - Lowest (network) = 192.0.2.144
    - Highest (broadcast) = 192.0.2.159
  - How many usable addresses now?

Aggregation tree

- Continue to divide prefixes as required
- Can visualize this as a tree



Questions on IPv4?

IPv6 addresses

- 128-bit binary number
- Conventionally represented in hexadecimal
  – 8 words of 16 bits, separated by colons

2607:8400:2880:0004:0000:0000:80df:9d13

- Leading zeros can be dropped
- One contiguous run of all-zero words can be replaced by "::"

2607:8400:2880:4::80df:9d13

Hexadecimal

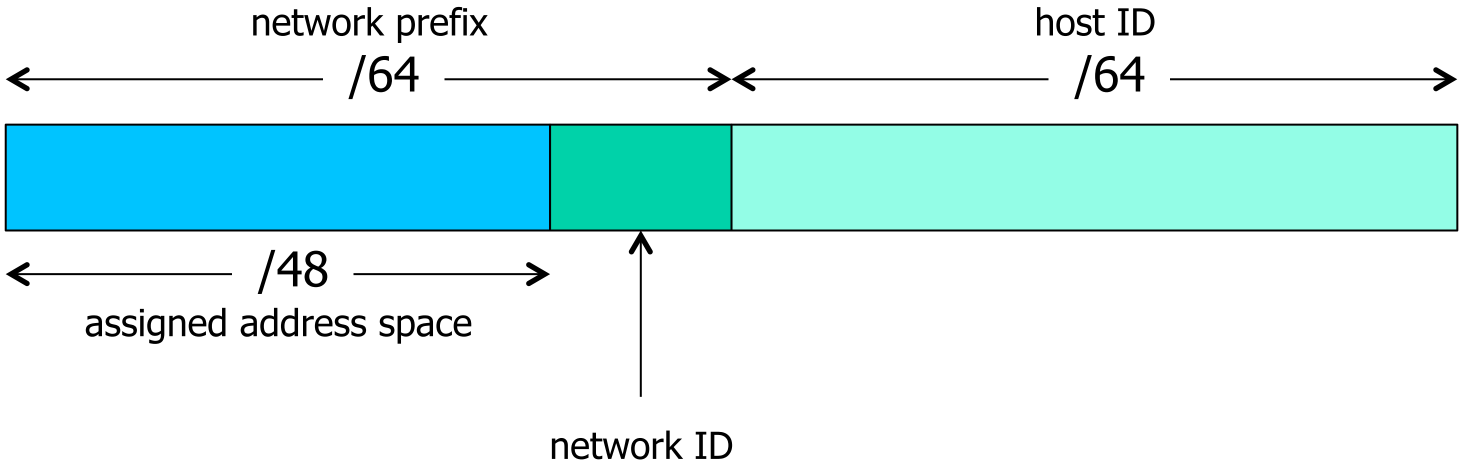
0000	**0**	1000	**8**
0001	**1**	1001	**9**
0010	**2**	1010	**a**
0011	**3**	1011	**b**
0100	**4**	1100	**c**
0101	**5**	1101	**d**
0110	**6**	1110	**e**
0111	**7**	1111	**f**

```
0000 = 0000000000000000
ffff = 1111111111111111
```

IPv6 rules

- With IPv6, every network prefix is /64
  - (OK, some people use /127 for P2P links)
- The remaining 64 bits can be assigned by hand, or picked automatically
  - e.g. derived from NIC MAC address
- There are special prefixes
  - e.g. link-local addresses start fe80::
- Total available IPv6 space is ≈ 2^{61} subnets
- Typical end-user allocation is /48

IPv6 addressing



- How many /64 networks can you build given a /48 allocation?

IPv6 addressing

- You are assigned 2001:db8:123::/48
    - 2001:0db8:0123:0000:0000:0000:0000:0000
- Lowest /64 network?
    - 2001:db8:123:0000::/64
    - written simply 2001:db8:123::/64
- Highest /64 network?
    - 2001:db8:123:ffff::/64

Ways to allocate the host part

- Do it automatically from MAC address – "stateless autoconfiguration"
  - Not recommended for servers: if you change the NIC then the IPv6 address changes!
- Can number sequentially from 1, or use the last octet of the IPv4 address
- Or embed the whole IPv4 address
  - e.g. 2607:8400:2880:4::80df:9d13
  - 80df9d13 hex = 128.223.157.19 in decimal
  - Can write 2607:8400:2880:4::128.223.157.19

Notes on IPv6

- Broadly similar to IPv4
- "ARP" is replaced by "NDP"
- IPv6 client configuration options
  - Stateless autoconf (router advertisements)
  - Stateless autoconf + stateless DHCPv6
  - Stateful DHCPv6
- Interfaces typically get both a link-local address and one or more routable prefixes
- "Dual stack" = v4 and v6 side-by-side

Questions on IPv6?