Names and Addresses: IPv4

The Internet Protocol allows two computers to exchange messages across a network that's built out of many different link layers. It does so through addresses. An IP packet has a source and a destination address. Routers decide which link to forward a packet over based on the packet's destination address. Let’s look in detail at what IP version 4 addresses look like, how they are formatted, and how they are allocated.

Goal of Internet Protocol Addresses

- Stitch many different networks together
- Need network-independent, unique address

The original goal of the Internet Protocol was to take many different networks and stitch them together. For this to work, the protocol needed a way to refer to a computer that was independent of the network it was on and unique. So a computer on an IBM network and a computer connected to a router over a serial line can talk to each other, and need a way to address each other. Today, IPv4 addresses are a bit more complicated, they're not totally unique due to a bunch of special cases and uses, but for now let's just assume they're unique.

Internet Protocol, Version 4

- An IPv4 address identifies a device on the Internet
  - Layer 3 (network) address
- 32 bits long (4 octets): a.b.c.d
  - Example: 171.64.64.64
  - Example: 128.30.76.82
  - Example: 12.22.58.30
- Netmask: apply this mask, if it matches, in the same network
  - Netmask of 255.255.255.0 means if the first 24 bits match
  - Netmask of 255.255.252.0 means if the first 22 bits match
  - Netmask of 255.128.0.0 means if the first 9 bits match
  - Smaller netmask (fewer 1s) means larger network

An Internet Protocol, version 4 address is 32 bits long. This 32 bits is often written as 4 octets, 4 8 bit values, in the form a.b.c.d. Here are three examples. 171.64.64.64, 128.30.76.82, and 12.22.58.30. Every device connected through IPv4 has an IP address. The IP layer delivers packets whose destination is this address to that device.

In addition to an address, a device typically also has something called a netmask. A netmask tells the device which IP addresses are local -- on the same link -- and which require going through an IP router. Think, for example, of a laptop on a wireless network. In order to send a packet to another device in the same wireless network, the laptop doesn't need to go through an IP router. It can, in theory, just send the packet directly to the other device since it's on the same link.

A netmask is written as a string of consecutive 1s, starting with the most significant bit. A netmask of 255.255.255.0, for example, means the first 3 octets are all 1s (2 to the 8th -1 is 255) and the last octet is zero. This means that an IP address which matches the first three octets -- 24 bits -- of your IP address is in the same network. A netmask of 255.255.252.0 means the netmask is 22 bits long, while 255.128.0.0 is a 9 bit netmask.

You tell whether two computers are in the same network by taking a bitwise AND of their addresses with the netmask. If the resulting addresses are equal, they are in the same network.

Internet Protocol, Version 4

- An IPv4 address identifies a device on the Internet
  - Layer 3 (network) address
- 32 bits long (4 octets): a.b.c.d
  - Example: 171.64.64.64
  - Example: 128.30.76.82
  - Example: 12.22.58.30
- Netmask: apply this mask, if it matches, in the same network
  - Netmask of 255.255.255.0 means if the first 24 bits match
  - Netmask of 255.255.252.0 means if the first 22 bits match
  - Netmask of 255.128.0.0 means if the first 9 bits match
  - Smaller netmask (fewer 1s) means larger network

Let's see what this looks like on my computer. I can open up a terminal an use the ifconfig program. My computer is connected to the Internet over WiFi, which happens to be the link named "en1." If we look inside the "en1" information, we can see that my Internet Protocol version 4 address is 192.168.0.106 and my netmask is 0xffffff00, which is hexidecimal for 255.255.255.0. This means that if I send an IP packet to an address beginning with 192.168.0 I should send it directly, but if it doesn't begin with 192.168.0 I need to send it through a router.

Quiz

For each source, destination, and netmask, mark whether the destination is in the same network as the source.

Source	Destination	Netmask	Same Network?
128.34.1.15	128.35.1.15	255.255.0.0	
10.0.1.4	10.0.1.5	255.255.255.0	
10.0.1.4	10.0.2.5	255.255.255.0	
171.64.15.33	171.64.15.5	255.255.255.224	
171.64.15.33	171.19.201.2	255.0.0.0	

Here's a quiz. For each source, destination, and netmask, mark whether the destination is in the same network as the source.

Let's walk through the answers.

The answer to the first row is no. They are in different networks. The two addresses differ in their second octet -- 34 versus 35. If we take a bitwise AND of the two addresses with the netmask, we get 128.34.0.0 and 128.35.0.0, which are not the same.

The answer to the second row is yes. They are in the same network. If we take a bitwise AND of the two addresses with the netmask, we get 10.0.1.0 in both cases.

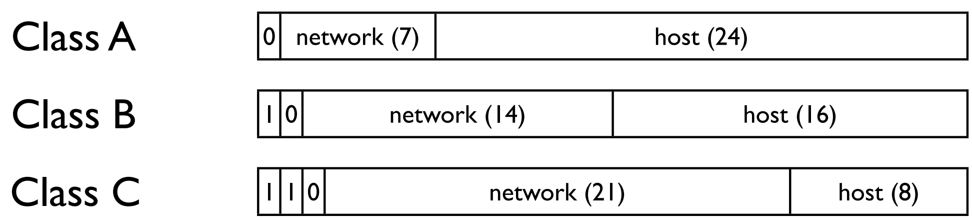
The answer to the third row is no. They are not in the same network, because they differ in their third octet. The source is in network 10.0.1.0, while the destination is in network 10.0.2.0.

The answer to the fourth row is no. They are on in the same network. Just think of the last byte. The address on the left, with 33, is 0x21. The address on the right is 0x05. 224 is 0xE0, which is three bits. Since the source has this bit set but the destination doesn't, they don't match. ANDed with the netmask, the source address is 171.64.15.32,while the destination is 171.64.15.0.

The answer to the final row is yes. They match in the first byte: both are 171.0.0.0 when ANDed with the netmask.

Address Structure (historical)

- Originally hierarchical: network + host
  - Network to get to correct network (administrative domain)
  - Host to get to correct device in network (within administrative domain)
- Originally 3 classes of addresses: class A, class B, class C



So how are IP addresses assigned? Originally, they were broken up into three classes: class A, class B, and class C. Each class separated an IP address into two part, network and host. The network part of the address denoted an administrative domain, such as MIT, BBN, or Stanford University. The host part of the address denoted which device within that network. Class A addresses had a leading 0, 7 bits of network (for 128 networks) and 24 bits of host, so could cover 16 million computers. Class B addresses had 16 bits of host, so could cover 65,536 computers. Class C addresses had 8 bits of host, so could cover 256 computers.

Address Structure Today

- Still assign contiguous ranges of addresses to nearby networks
  - Class A, B, C is too coarse grained (e.g., MIT dorms!)
  - http://news.stanford.edu/news/1999/january27/itss127.html
- Classless Inter-Domain Routing (CIDR)
  - Address block is a pair: *address,count*
  - Counts are powers of 2, specify netmask length
  - 171.64.0.0/16 means any address in the range 171.64.0.0 to 171.64.255.255
  - A /24 describes 256 addresses, a /20 describes 4,096 addresses
- Stanford today has 5 /16 blocks -- 325,000 addresses

While classes A, B, and C are simple, we quickly found out they were not flexible enough. For example, both MIT and Stanford received one of the first class A address blocks – over 4 million addresses. For a while, MIT would give each of its dorms the equivalent of a class B -- 65,000 addresses for a few hundred people! When IP addresses were plentiful, this wasn't a problem, but as their use increased we needed a better allocation policy.

A useful note: Stanford gave up its class A block in 1999, MIT still has its.

Today, IPv4 addresses are structured thought something called CIDR, or Classless Inter-Domain Routing. Rather than have prefixes only of length 8, 16, and 24 bits, CIDR allows prefixes to be any number of bits. This means all CIDR prefixes define a block of addresses that is a power of 2 in size. When we talk about a CIDR address, we refer to its netmask length. So, for example, when we talk about a "slash 16", we mean a netmask of length 16. This CIDR block describes 2 to the 16 addresses, or 65,536. When we talk about a "slash 20", we mean a netmask of length 20. This CIDR block describes 2 to the 12 addresses, or 4.096 addresses. CIDR blocks are how addresses are structured, addressed, and managed today.

Stanford today has 5 /16 blocks, about 325,000 IPv4 addresses.

IPv4 Address Assignment

- IANA: Internet Assigned Numbers Authority
  - Internet Corporation for Assignment of Names and Numbers (ICANN)'s job
- IANA gives out /8s to Regional Internet Registries (RIRs)
  - Ran out in February 2011, in special end case of giving 1 to each RIR
- RIRs responsible for geographic regions, each has own policy
  - AfriNIC: Africa
  - ARIN: U.S.A., Canada, Carribean, Antarctica
  - APNIC: Asia, Australia, New Zealand
  - LACNIC: Latin America, Carribean
  - RIPE NCC: Europe, Russia, Middle East, Central Asia

So how are IPv4 addresses allocated and managed? There's an organization called IANA, for the Internet Assigned Numbers Authority. The ultimate authority is ICANN, the Internet Corporation for Assignment of Names and Numbers. ICANN delegates the work to IANA.

IANA gives out slash-8s, describing 16 million addresses, to Regional Internet Registries, or RIRs. Each continent has its own RIR. The RIR for the United States is ARIN, while the RIR for the western Pacific is APNIC. These RIRs each have their own policy for how they break up the /8s into smaller blocks of addresses and assign them to parties who need them.

You might have read in the news is that we've run out of IP addresses. This isn't really true -- there are many unused addresses today. What *did* happen is that IANA ran out of /8s to give out. It reached a special end case in its charter. When reduced to its last 5 /8s, IANA gave one /8 to each RIR. Now address management and allocation is up to RIRs. In 2012, Jon Peterson, then a member of the Internet Architecture Board, gave a talk at Stanford on some of the political, economic, and technical complications this raises. The talk isn't required material for this course, but I recommend it highly.

So now you've seen the structure of IPv4 addresses, how they are allocated, and how end hosts make their first hop routing decisions, that is, whether to send to a local node or their gateway router. Addresses today are managed and allocated in terms of CIDR blocks, whose size is defined by their prefix length. A shorter prefix, say, a /8, is a larger block than a longer prefix, say, a /10.