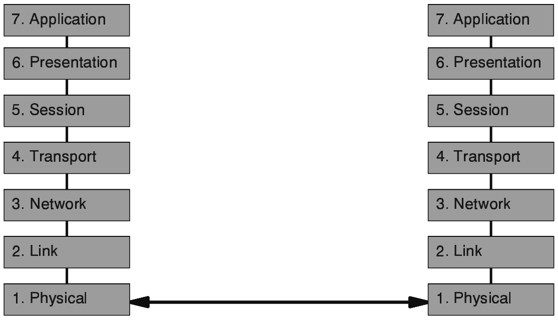
The OSI model (Open Systems Interconnection)

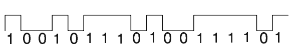


The OSI model

- A generic model, not a specific protocol
- Breaks down networking into simpler parts
- Helps us understand, discuss and compare networks

Layer 1 - Physical Layer

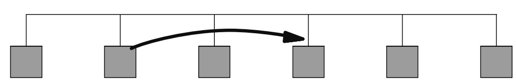
- Transfers stream of *bits* from A to B
- Defines connectors, type of cable, maximum length, topology, voltages for 0 and 1, speed (bits per second)



- No concept of bytes or frames

Layer 2 - Link Layer

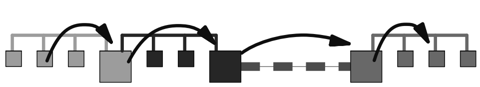
- Send data between adjacent machines
- Organise bits into bytes and frames
  - Special bit patterns as delimiters
- Address frames to a specific machine on a shared (broadcast) medium



- Some layer 2's detect corrupted frames
- Some layer 2's retransmit corrupted frames
  (but *not* ethernet)

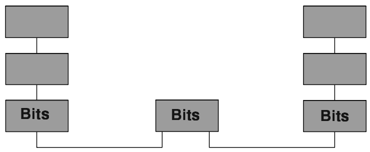
Layer 3 - Network Layer

- Send data through multiple hops to far distant networks - "internetwork"
- Move data between different Layer 2 types
- Uniform numbering scheme
- Globally scalable

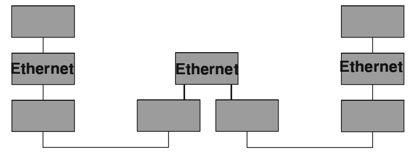


Building networks

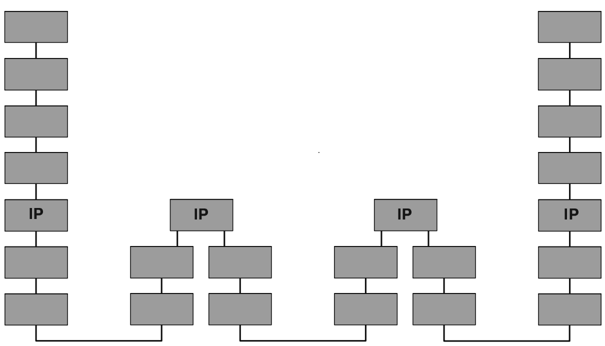
Repeaters (Hubs)



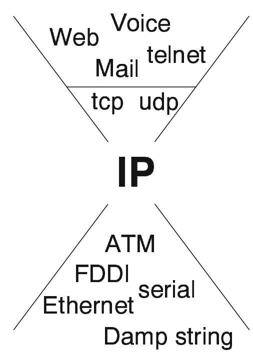
Switching (Bridging)



IP Forwarding - multiple hops



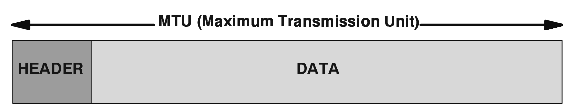
The Hourglass Model



The Internet Protocol (IP)

- Layer 3 in the TCP/IP stack
- Delivers chunks of data - "datagrams" - across an internetwork
- Scales to global network - The Internet
- Integrates different LAN technologies
- RFC 791

IP Datagram Structure



- Header
  - Source IP address - *where it came from*
  - Destination IP address - *where it is going to*
  - Header checksum
  - Other fields (TTL, Layer 4 protocol identifier, Fragmentation information)
- Data
  - The actual data you want to carry
- Total size up to MTU bytes
  - What limits the MTU?

IP Addresses / IP Numbers

- IP number identifies a device (host)
- Globally unique for every host
  - Why?
- Independent of layer 2 addresses
- 32 bit binary number

Example:

11001110000110111110111000000101

IP numbers (continued)

- Convert to decimal for convenience
- Group into bytes (8 bits) and convert each in turn; separate with periods

```
11001110  00011011  11101110  00000101
   ↓         ↓         ↓         ↓
  206   .   27    .   238   .    5
```

- There is nothing special about the 8 bit boundaries; to the computer it is still a single 32 bit number

IP number range

- What is the SMALLEST possible IP number?
- What is the LARGEST possible IP number?
- How many IP numbers are there in total?

IP numbers are divided into PREFIX and HOST part

- The position of this split defines the size of the network
- e.g. a "/28" network has 28 bits of PREFIX and the remaining 4 bits of HOST number

11001110 00011011 11101110 00000101

Prefix — Host

- We can choose any combination from 0000 to 1111 in the remaining 4 bits, to give a unique IP for each host
- How many IP addresses does this give?

The GOLDEN RULES for allocating IP addresses

- All hosts on the same **physical network** share the **same unique prefix**

  10101001 11011110 00010000 0xxxxxxx

- The remaining part *(host number)* must be different for each host

  10101001 11011110 00010000 0**xxxxxxx**

- Host numbers of all zeros and all ones are reserved

  Network Number

  10101001 11011110 00010000 0**0000000**

  Broadcast Address

  10101001 11011110 00010000 0**1111111**

Example: a /29 network

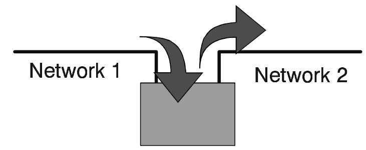
- How many bits for the host number?
- How many combinations?
- But two of these are reserved (000 and 111)
- So how many hosts can be attached to this network?

Nowadays, prefixes must be given explicitly

- Some software lets you enter e.g. "/28"
- Older software requires you to enter a netmask
  - /28 = 28 ones followed by 4 zeros
  - 11111111 11111111 11111111 11110000
  - 255.255.255.240 (decimal)
  - 0xfffffff0 (hex)
- Just use a conversion table

What is forwarding?

- Receiving a datagram on one interface and resending it on another



- Why? Because Layer 2 can only send to direct neighbours (on the same link)
- Also lets us transfer data between different network types

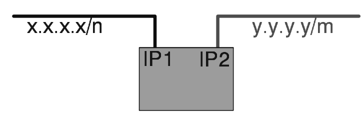
What is a router?

- A host with two or more interfaces *and* which has been configured to forward datagrams *
- Works at layer 3: receives datagrams and forwards them based on the destination IP address
- For a full definition see RFC 1812
- Older documents call it a "Gateway" but this term is normally used for a layer 7 gateway

* Otherwise it is just a multi-homed host

What is a router? (Contd)

- A router is connected to two or more networks
- The Golden Rules say that every network must have its own prefix
- Therefore a router must have two or more IP addresses - one for each interface



What happens when a router receives a datagram?

- First it checks to see if the destination address is local (i.e. the router itself is the final destination)
- Next it decrements the Time To Live (TTL) field in the IP header. If it reaches zero, the datagram is discarded
- Finally it looks up the destination in a *forwarding table* to decide where to send it next
- Could be on a directly-connected network, or could have to send it via another router

Forwarding is hop-by-hop

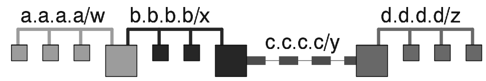
- Each router can only communicate directly with devices which are on the same networks that it is connected to (Layer 2)
- Each router gets the datagram one hop closer to the destination
- Each router makes an *independent* decision as to the next hop (the route is not pre-planned)
- Each router has a different view of the world so has a different forwarding table

Using prefixes for forwarding

- We don't list every IP number on the Internet - the table would be huge
- Instead, the forwarding table contains prefixes (network numbers)
- "If the first /n bits matches this entry, send the datagram this way"
- If more than one prefix matches, the longest prefix wins (more specific route)
- 0.0.0.0/0 is "default route" - matches anything, but only if no other prefix matches

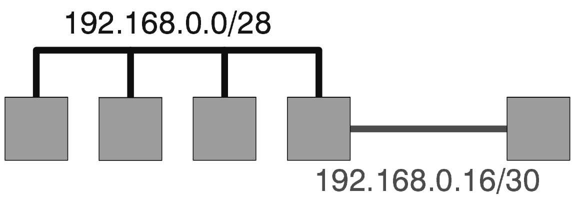
Allocating IP numbers on an Internetwork

- Easy: follow the Golden Rules!
  1. Give each network its own prefix
  2. Give each machine an IP number using this prefix plus a host number
  3. Don't use the first and last host numbers (all 0's and all 1's)



IP allocation example

- Allocate IP numbers to the following devices



Subdividing prefixes

- Every network must have its own network number (prefix)
- You will most likely only get one prefix from your provider
- So if you have more than one network, you have to divide a large prefix into several smaller ones

Subdividing prefixes: example

- You have been allocated 195.176.112.0/25
- How many IP numbers is this?

- What is the smallest IP number available? The largest IP number available?

Example (contd)

- We have decided we need 8 separate networks, linked by routers
- Each network needs its own prefix
- So we need *more* prefixes but each with *fewer* IP numbers available

Example (contd)

- Prefix we have been given

195.176.112.0/25

11000011 10110000 01110100 0hhhhhhh

195.176.112.0 to 195.176.112.127

(195.176.112.1 to 195.176.112.126 usable)

Example (contd)

- Now let's make the prefix longer

11000011 10110000 01110100 0nnnhhhh

- What is the new prefix length?
- Now we have 3 more bits of network number and 3 less bits of host number
- How many combinations of nnn are there?

Example (contd)

- We have created these new prefixes

11000011 10110000 01110100 0000hhhh
11000011 10110000 01110100 0001hhhh
11000011 10110000 01110100 0010hhhh
11000011 10110000 01110100 0011hhhh
11000011 10110000 01110100 0100hhhh
11000011 10110000 01110100 0101hhhh
11000011 10110000 01110100 0110hhhh
11000011 10110000 01110100 0111hhhh

Example (contd)

- How many hosts can each network have?
- What are the network numbers we have created?

Dividing Prefixes - class example

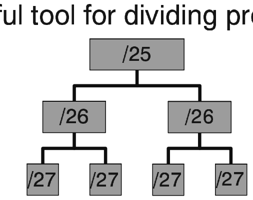
- Here is a prefix:

192.168.34.0/24

- What is the smallest and largest IP number in this range?
- Turn this into two /25 prefixes
- What are the smallest and largest IP numbers in each of the new ranges?

Aggregation Tree

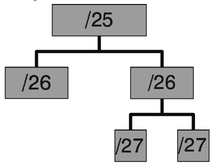
- A useful tool for dividing prefixes



- "Aggregation" is the process of combining smaller prefixes into larger ones - the reverse of what we have been doing

Aggregation Tree (contd)

- You don't have to divide your space into equal sized prefixes



This is OK: one /26 network and two /27 networks

- But check your work - work out the ranges of IP numbers and see that they don't overlap