CSC 358 – Principles of Computer Networks

Handout # 4: Interconnecting LANs; Internet Protocol (IP)

Joe Lim

Department of Mathematical and Computational Sciences

joe.lim@utoronto.ca

Announcements

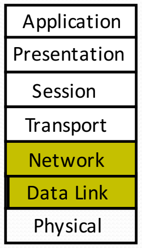
- PS1 Due Friday, Feb 2nd at 11:59pm.
  - Remember the late submission policy.
  - Remember academic integrity guidelines.
  - Submit electronically on MarkUs.
    - ***NOTE. File names must be:*** ps1.pdf
  - You can scan and save as a pdf file.
    - Not preferred, but acceptable.
    - Make sure scan is readable!
- Programming assignment 1
  - Have you tested Mininet?
  - Have you been able to run your VM?
  - Start early! Start early! Start early! …

Announcements – Cont'd

- This week's tutorial
  - PA1 Overview
- Discussion Board – DisCourse
  - https://mcs.utm.utoronto.ca/forum/

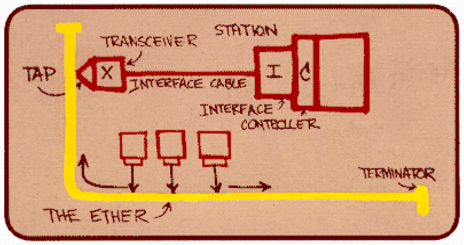
The Story

- So far …
  - Layers, and protocols
  - Link layer
    - Media type, encoding
    - Framing, link model
    - Error detection, correction
- This time
  - Interconnecting LANs
    - Hubs, switches, and bridges
  - The Internet Protocol



Ethernet

- Dominant wired LAN technology
- First widely used LAN technology
- Simpler, cheaper than token LANs and ATM
- Kept up with speed race: 10 Mbps – 10 Gbps

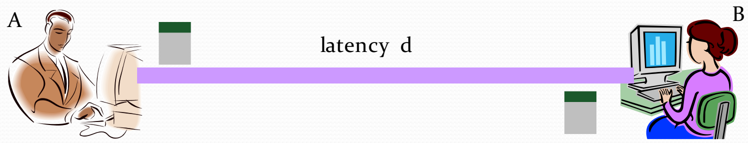


Metcalfe's Ethernet sketch

Ethernet Uses CSMA/CD

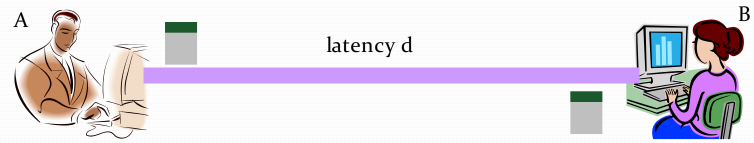
- Carrier sense: wait for link to be idle
  - Channel idle: start transmitting
  - Channel busy: wait until idle
- Collision detection: listen while transmitting
  - No collision: transmission is complete
  - Collision: abort transmission, and send jam signal
- Random access: exponential back-off
  - After collision, wait a random time before trying again
  - After m^{th} collision, choose K randomly from $\{0, \ldots, 2^m-1\}$
  - … and wait for K*512 bit times before trying again

Limitations on Ethernet Length



- Latency depends on physical length of link
  - Time to propagate a packet from one end to the other
- Suppose A sends a packet at time t
  - And B sees an idle line at a time just before t+d
  - … so B happily starts transmitting a packet
- B detects a collision, and sends jamming signal
  - But A doesn't see collision till t+2d

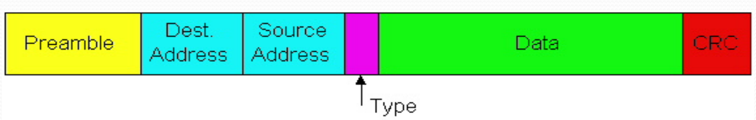
Limitations on Ethernet Length



- A needs to wait for time 2d to detect collision
  - So, A should keep transmitting during this period
  - … and keep an eye out for a possible collision
- Imposes restrictions on Ethernet
  - Maximum length of the wire: 2500 meters
  - Minimum length of the packet: 512 bits (64 bytes)

Ethernet Frame Structure

- Addresses: source and destination MAC addresses
  - Adaptor passes frame to network-level protocol
    - If destination address matches the adaptor
    - Or the destination address is the broadcast address
  - Otherwise, adapter discards frame
- Type: indicates the higher layer protocol
  - Usually IP
  - But also Novell IPX, AppleTalk, …
- CRC: cyclic redundancy check
  - Checked at receiver
  - If error is detected, the frame is simply dropped



Unreliable, Connectionless Service

- Connectionless
  - No handshaking between sending and receiving adapter.
- Unreliable
  - Receiving adapter doesn't send ACKs or NACKs
  - Packets passed to network layer can have gaps
  - Gaps will be filled if application is using TCP
  - Otherwise, the application will see the gaps

Shuttling Data at Different Layers

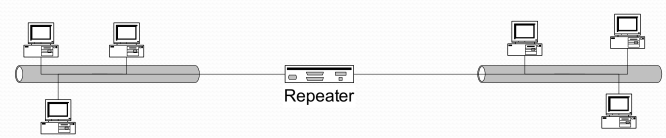
- Different devices switch different things
  - Physical layer: electrical signals (repeaters and hubs)
  - Link layer: frames (bridges and switches)
  - Network layer: packets (routers)

Application gateway
Transport gateway
Router
Bridge, switch
Repeater, hub

Frame header	Packet header	TCP header	User data

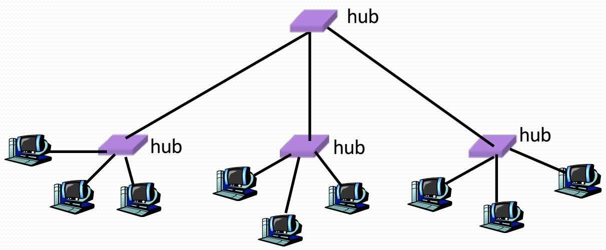
Physical Layer: Repeaters

- Distance limitation in local-area networks
  - Electrical signal becomes weaker as it travels
  - Imposes a limit on the length of a LAN
- Repeaters join LANs together
  - Analog electronic device
  - Continuously monitors electrical signals on each LAN
  - Transmits an amplified copy



Physical Layer: Hubs

- Joins multiple input lines electrically
  - Designed to hold multiple line cards
  - Do not necessarily amplify the signal
- Very similar to repeaters
  - Also operates at the physical layer

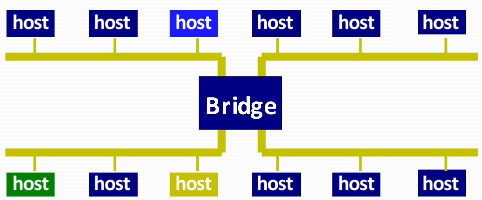


Limitations of Repeaters and Hubs

- One large collision domain
  - Every bit is sent everywhere
  - So, aggregate throughput is limited
  - E.g., three departments each get 10 Mbps independently
  - … and then connect via a hub and must share 10 Mbps
- Cannot support multiple LAN technologies
  - Does not buffer or interpret frames
  - So, can't interconnect between different rates or formats
  - E.g., 10 Mbps Ethernet and 100 Mbps Ethernet
- Limitations on maximum nodes and distances
  - Does not circumvent the limitations of shared media
  - E.g., still cannot go beyond 2500 meters on Ethernet

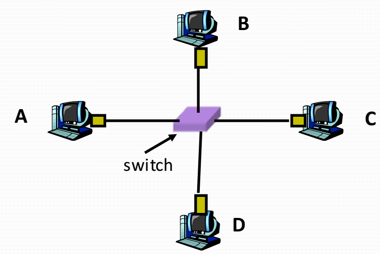
Link Layer: Bridges

- Connects two or more LANs at the link layer
  - Extracts destination address from the frame
  - Looks up the destination in a table
  - Forwards the frame to the appropriate LAN segment
- Each segment is its own collision domain



Link Layer: Switches

- Typically connects individual computers
  - A switch is essentially the same as a bridge
  - … though typically used to connect hosts, not LANs
- Like bridges, support concurrent communication
  - Host A can talk to C, while B talks to D

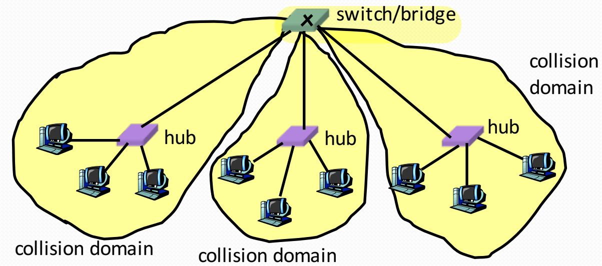


Dedicated Access and Full Duplex

- Dedicated access
  - Host has direct connection to the switch
  - … rather than a shared LAN connection
- Full duplex
  - Each connection can send in both directions
  - Host sending to switch, and host receiving from switch
  - E.g., in 10BaseT and 100Base T
- Completely avoids collisions
  - Each connection is a bidirectional point-to-point link
  - No need for carrier sense, collision detection, and so on

Bridges/Switches: Traffic Isolation

- Switch breaks subnet into LAN segments
- Switch filters packets
  - Frame only forwarded to the necessary segments
  - Segments become separate collision domains



Advantages Over Hubs/Repeaters

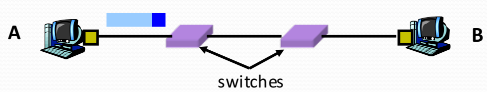
- Only forwards frames as needed
  - Filters frames to avoid unnecessary load on segments
  - Sends frames only to segments that need to see them
- Extends the geographic span of the network
  - Separate collision domains allow longer distances
- Improves privacy by limiting scope of frames
  - Hosts can "snoop" the traffic traversing their segment
  - … but not all the rest of the traffic
- Applies carrier sense and collision detection
  - Does not transmit when the link is busy
  - Applies exponential back-off after a collision
- Joins segments using different technologies

Disadvantages Over Hubs/Repeaters

- Delay in forwarding frames
  - Bridge/switch must receive and parse the frame
  - … and perform a look-up to decide where to forward
  - Storing and forwarding the packet introduces delay
  - Solution: cut-through switching
- Need to learn where to forward frames
  - Bridge/switch needs to construct a forwarding table
  - Ideally, without intervention from network administrators
  - Solution: self-learning
- Higher cost
  - More complicated devices that cost more money

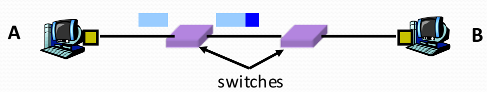
Motivation For Cut-Through Switching

- Buffering a frame takes time
  - Suppose L is the length of the frame
  - And R is the transmission rate of the links
  - Then, receiving the frame takes L/R time units
- Buffering delay can be a high fraction of total delay
  - Propagation delay is small over short distances
  - Making buffering delay a large fraction of total
  - Analogy: large group walking through NYC



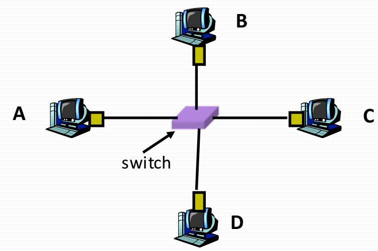
Cut-Through Switching

- Start transmitting as soon as possible
  - Inspect the frame header and do the look-up
  - If outgoing link is idle, start forwarding the frame
- Overlapping transmissions
  - Transmit the head of the packet via the outgoing link
  - … while still receiving the tail via the incoming link
  - Analogy: different folks crossing different intersections



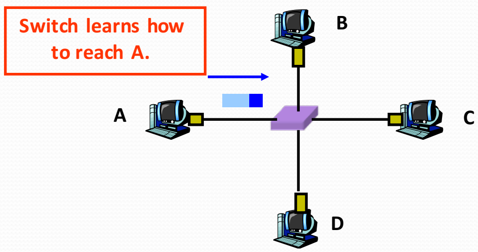
Motivation For Self Learning

- Switches forward frames selectively
  - Forward frames only on segments that need them
- Switch table
  - Maps destination MAC address to outgoing interface
  - Goal: construct the switch table automatically



Self Learning: Building the Table

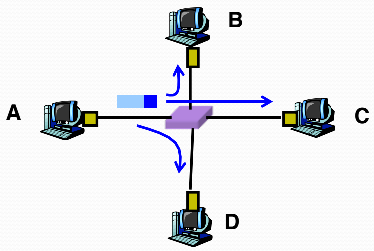
- When a frame arrives
  - Inspect the source MAC address
  - Associate the address with the incoming interface
  - Store the mapping in the switch table
  - Use a time-to-live field to eventually forget the mapping



Self Learning: Handling Misses

- When frame arrives with unfamiliar destination
  - Forward the frame out all of the interfaces
  - … except for the one where the frame arrived
  - Hopefully, this case won't happen very often

When in doubt, shout!



Switch Filtering/Forwarding

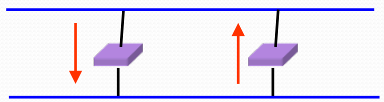
When switch receives a frame:

index switch table using MAC dest address
if entry found for destination
then {
if dest on segment from which frame arrived
then drop the frame
else forward the frame on interface indicated
}
else flood

forward on all but the interface on which the frame arrived

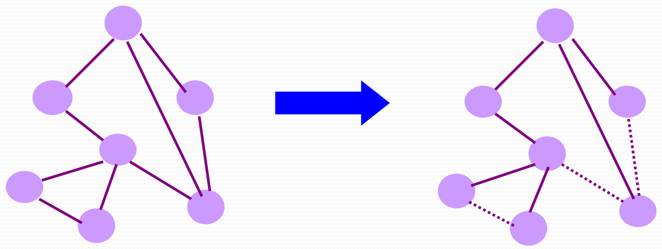
Flooding Can Lead to Loops

- Switches sometimes need to broadcast frames
  - Upon receiving a frame with an unfamiliar destination
  - Upon receiving a frame sent to the broadcast address
- Broadcasting is implemented by flooding
  - Transmitting frame out every interface
  - … except the one where the frame arrived
- Flooding can lead to forwarding loops
  - E.g., if the network contains a cycle of switches
  - Either accidentally, or by design for higher reliability



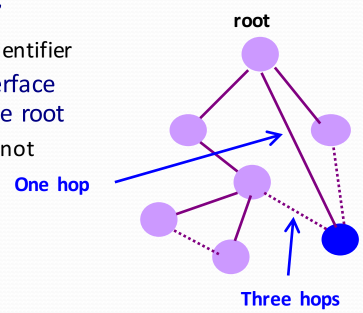
Solution: Spanning Trees

- Ensure the topology has no loops
  - Avoid using some of the links when flooding
  - … to avoid forming a loop
- Spanning tree
  - Sub-graph that covers all vertices but contains no cycles
  - Links not in the spanning tree do not forward frames



Constructing a Spanning Tree

- Need a distributed algorithm
  - Switches cooperate to build the spanning tree
  - … and adapt automatically when failures occur
- Key ingredients of the algorithm
  - Switches need to elect a "root"
    - The switch with the smallest identifier
  - Each switch identifies if its interface is on the shortest path from the root
    - And exclude it from the tree if not
  - Messages (Y, d, X)
    - From node X
    - Claiming Y is the root
    - And the distance is d

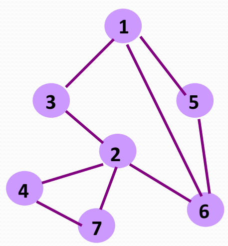


Steps in Spanning Tree Algorithm

- Initially, each switch thinks it is the root
  - Switch sends a message out every interface
  - … identifying itself as the root with distance 0
  - Example: switch X announces (X, 0, X)
- Switches update their view of the root
  - Upon receiving a message, check the root id
  - If the new id is smaller, start viewing that switch as root
- Switches compute their distance from the root
  - Add 1 to the distance received from a neighbor
  - Identify interfaces not on a shortest path to the root
  - … and exclude them from the spanning tree

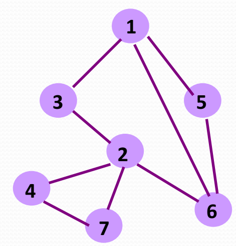
Example From Switch #4's Viewpoint

- Switch #4 thinks it is the root
  - Sends (4, 0, 4) message to 2 and 7
- Then, switch #4 hears from #2
  - Receives (2, 0, 2) message from 2
  - … and thinks that #2 is the root
  - And realizes it is just one hop away
- Then, switch #4 hears from #7
  - Receives (2, 1, 7) from 7
  - And realizes this is a longer path
  - So, prefers its own one-hop path
  - And removes 4-7 link from the tree



Example From Switch #4's Viewpoint

- Switch #2 hears about switch #1
  - Switch 2 hears (1, 1, 3) from 3
  - Switch 2 starts treating 1 as root
  - And sends (1, 2, 2) to neighbors
- Switch #4 hears from switch #2
  - Switch 4 starts treating 1 as root
  - And sends (1, 3, 4) to neighbors
- Switch #4 hears from switch #7
  - Switch 4 receives (1, 3, 7) from 7
  - And realizes this is a longer path
  - So, prefers its own three-hop path
  - And removes 4-7 link from the tree



Robust Spanning Tree Algorithm

- Algorithm must react to failures
  - Failure of the root node
    - Need to elect a new root, with the next lowest identifier
  - Failure of other switches and links
    - Need to recompute the spanning tree
- Root switch continues sending messages
  - Periodically reannouncing itself as the root (1, 0, 1)
  - Other switches continue forwarding messages
- Detecting failures through timeout (soft state!)
  - Switch waits to hear from others
  - Eventually times out and claims to be the root

Switches vs. Routers

- Advantages of switches over routers
  - Plug-and-play
  - Fast filtering and forwarding of frames
  - No pronunciation ambiguity (e.g., "rooter" vs. "rowter")
- Disadvantages of switches over routers
  - Topology is restricted to a spanning tree
  - Large networks require large ARP tables
  - Broadcast storms can cause the network to collapse

Comparing Hubs, Switches, & Routers

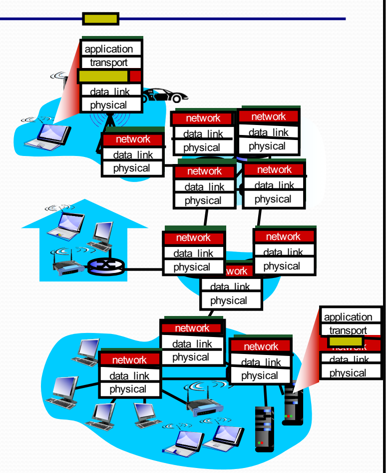
	hubs	routers	switches
traffic isolation	no	yes	yes
plug & play	yes	no	yes
optimal routing	no	yes	no
cut through	yes	no	yes

Part II – The Internet Protocol (IP)

- IP: The Internet Protocol
  - Service characteristics
  - The IP Datagram format
  - IP addresses
  - Classless Inter-Domain Routing (CIDR)
  - An aside: Turning names into addresses (DNS)

Network Layer

- transport segment from sending to receiving host
- on sending side encapsulates segments into datagrams
- on receiving side, delivers segments to transport layer
- network layer protocols in *every* host, router
- router examines header fields in all IP datagrams passing through it



Two key Network Layer Functions

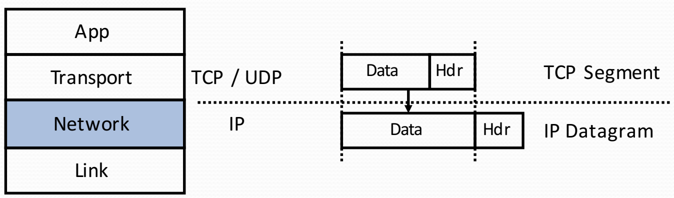
- *forwarding:* move packets from router's input to appropriate router output
- *routing:* determine route taken by packets from source to dest.
  - *routing algorithms*

analogy:

- *routing:* process of planning trip from source to dest
- *forwarding:* process of getting through single interchange

The Internet Protocol (IP)

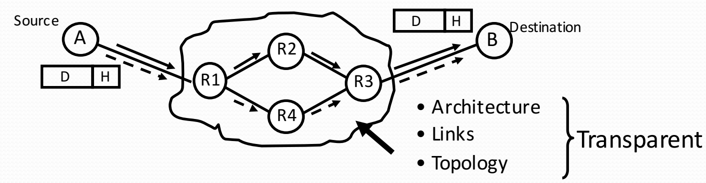
Protocol Stack



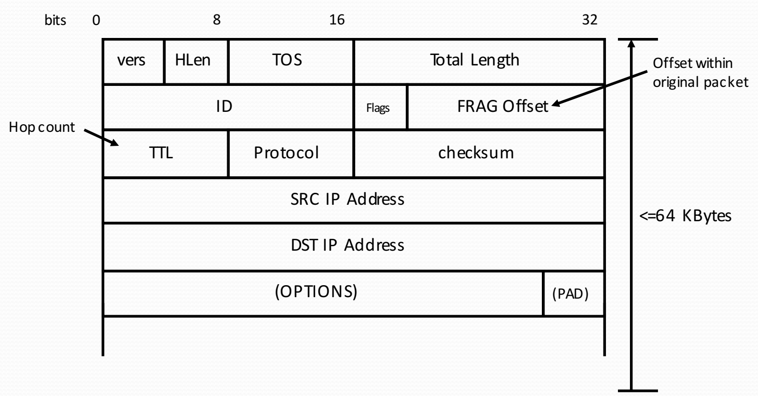
The Internet Protocol (IP)

Characteristics of IP

- CONNECTIONLESS: mis-sequencing
- UNRELIABLE: may drop packets…
- BEST EFFORT: … but only if necessary
- DATAGRAM: individually routed

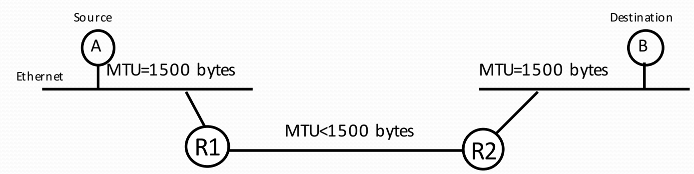


The IP Datagram

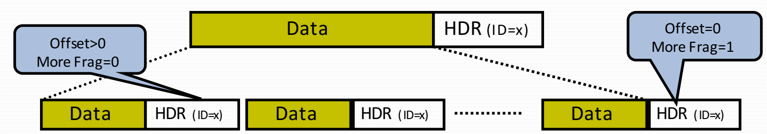


Fragmentation

Problem: A router may receive a packet larger than the maximum transmission unit (MTU) of the outgoing link.



Solution: R1 fragments the IP datagram into multiple, self-contained datagrams.



Fragmentation

- Fragments are re-assembled by the destination host; not by intermediate routers.
- To avoid fragmentation, hosts commonly use path MTU discovery to find the smallest MTU along the path.
- Path MTU discovery involves sending various size datagrams until they do not require fragmentation along the path.
- Most links use MTU>=1500bytes today.
- Try:

```
traceroute -F www.uwaterloo.ca 1500 and
traceroute -F www.uwaterloo.ca 1501
```

- (DF=1 set in IP header; routers send "ICMP" error message, which is shown as "!F").
- Bonus: Can you find a destination for which the path MTU < 1500 bytes?

Fragmentation, Reassembly

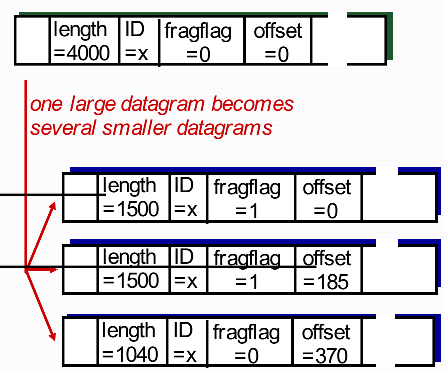
example:

- 4000 byte datagram
- MTU = 1500 bytes

1480 bytes in data field

Subtracted 20 bytes for IP header

Offset (byte addressable) =1480/8

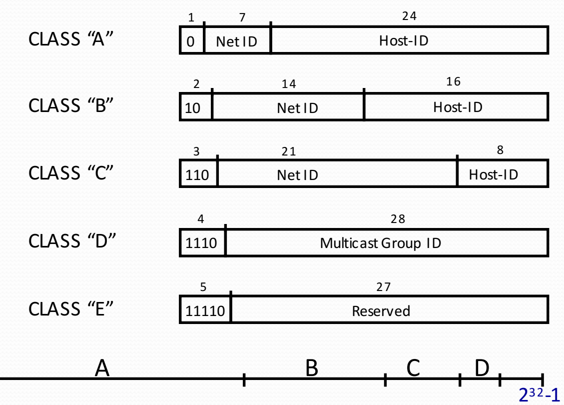


IP Addresses

- IP (Version 4) addresses are 32 bits long
- Every interface has a unique IP address:
  - A computer might have two or more IP addresses
  - A router has many IP addresses
- IP addresses are hierarchical
  - They contain a network ID and a host ID
  - E.g. Apple computers addresses start with: 17….
- IP addresses are assigned statically or dynamically (e.g. DHCP)
- IP (Version 6) addresses are 128 bits long

IP Addresses

Originally there were 5 classes:



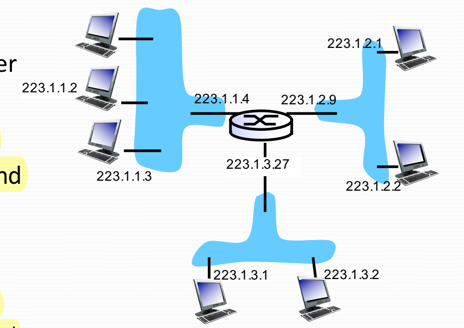
IP Addresses – *Examples*

Class "A" address: www.mit.edu
18.7.22.83
(18<128 => Class A)

Class "B" address: www.toronto.edu
142.150.210.13
(128<142<128+64 => Class B)

IP Addresses

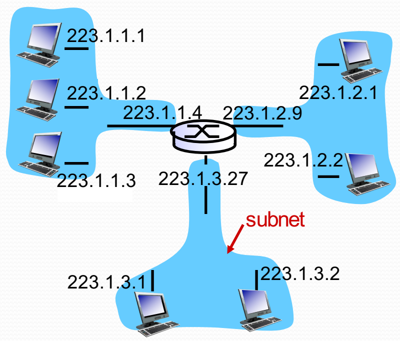
- *IP address:* 32-bit identifier for host, router *interface*
- *interface:* connection between host/router and physical link
  - router's typically have multiple interfaces
  - host typically has one or two interfaces (e.g., wired Ethernet, wireless 802.11)
- *IP addresses associated with each interface*



$$223.1.1.1 = \underbrace{11011111}_{223}\ \underbrace{00000001}_{1}\ \underbrace{00000001}_{1}\ \underbrace{00000001}_{1}$$

Subnets

- IP address:
  - subnet part - high order bits
  - host part - low order bits
- *what's a subnet ?*
  - device interfaces with same subnet part of IP address
  - can physically reach each other *without intervening router*



network consisting of 3 subnets

IP Addressing

Problem:

- Address classes were too "rigid". For most organizations, Class C were too small and Class B too big. Led to inefficient use of address space, and a shortage of addresses.
- Organizations with internal routers needed to have a separate (Class C) network ID for each link.
- And then every other router in the Internet had to know about every network ID in every organization, which led to large address tables.
- Small organizations wanted Class B in case they grew to more than 255 hosts. But there were only about 16,000 Class B network IDs.

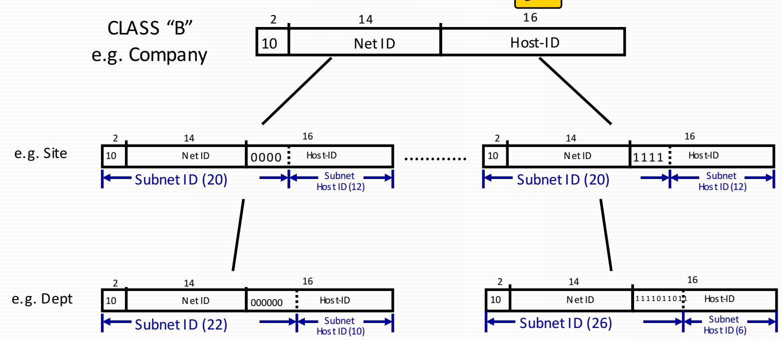
IP Addressing

Two solutions were introduced:

- Subnetting within an organization to subdivide the organization's network ID.
- Classless Inter-Domain Routing (CIDR) in the Internet backbone was introduced in 1993 to provide more efficient and flexible use of IP address space.

- CIDR is also known as "supernetting" because subnetting and CIDR are basically the same idea.

Subnetting



Subnetting

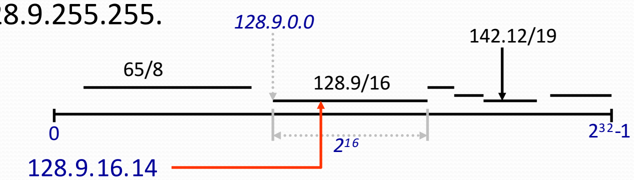
- Subnetting is a form of hierarchical routing.
- Subnets are usually represented via an address plus a subnet mask or "netmask".
- E.g.

```
yganjali@apps0.cs.toronto.edu > ifconfig eth0
   Link encap:Ethernet  HWaddr 00:15:17:1C:85:30
   inet addr:128.100.3.40  Bcast:128.100.3.255  Mask:ffffff00
   UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
```

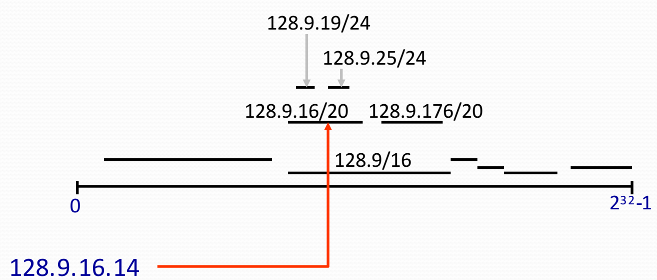
- Netmask ffffff00: the first 24 bits are the subnet ID, and the last 8 bits are the host ID.
- Can also be represented by a "prefix + length", e.g. 128.100.3.0/24, or just 128.100.3/24.

Classless Inter-Domain Routing (CIDR) Addressing

- The IP address space is broken into line segments.
- Each line segment is described by a prefix.
- A prefix is of the form x/y where x indicates the prefix of all addresses in the line segment, and y indicates the length of the segment.
- E.g. The prefix 128.9/16 represents the line segment containing addresses in the range: 128.9.0.0 … 128.9.255.255.



Classless Inter-Domain Routing (CIDR) – Addressing



Most specific route = "longest matching prefix"

Classless Inter-Domain Routing (CIDR) – Addressing

Prefix aggregation:

If a service provider serves two organizations with prefixes, it can (sometimes) aggregate them to form a shorter prefix. Other routers can refer to this shorter prefix, and so reduce the size of their address table.

E.g. ISP serves 128.9.14.0/24 and 128.9.15.0/24, it can tell other routers to send it all packets belonging to the prefix 128.9.14.0/23.

ISP Choice:

In principle, an organization can keep its prefix if it changes service providers.

Detour: Map Computer Names to IP addresses

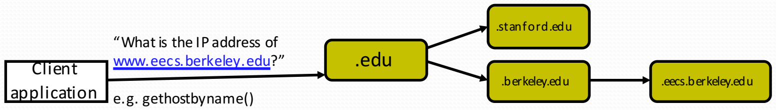
The Domain Naming System (DNS)

- Names are hierarchical and belong to a domain:
  - e.g. apps0.cs.utoronto.ca
  - Common domain names: .com, .edu, .gov, .org, .net, .ca (or other country-specific domain).
  - Top-level names are assigned by the Internet Corporation for Assigned Names and Numbers (ICANN).
  - A unique name is assigned to each organization.
- DNS Client-Server Model
  - DNS maintains a hierarchical, distributed database of names.
  - Servers are arranged in a hierarchy.
  - Each domain has a "root" server.
  - An application needing an IP address is a DNS client.

Mapping Computer Names to IP addresses

The Domain Naming System (DNS)

- A DNS Query
  - Client asks local server.
  - If local server does not have address, it asks the root server of the requested domain.
  - Addresses are cached in case they are requested again.



Example: On CDF machines, try "host www.eecs.berkeley.edu"

ICMP: internet control message protocol

- used by hosts & routers to communicate network-level information
  - error reporting: unreachable host, network, port, protocol
  - echo request/reply (used by ping)
- network-layer "above" IP:
  - ICMP msgs carried in IP datagrams
- ICMP message: type, code plus first 8 bytes of IP datagram causing error

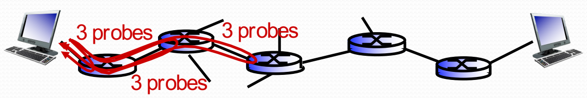
Type	Code	description
0	0	echo reply (ping)
3	0	dest. network unreachable
3	1	dest host unreachable
3	2	dest protocol unreachable
3	3	dest port unreachable
3	6	dest network unknown
3	7	dest host unknown
4	0	source quench (congestion control - not used)
8	0	echo request (ping)
9	0	route advertisement
10	0	router discovery
11	0	TTL expired
12	0	bad IP header

Traceroute and ICMP

- source sends series of UDP segments to destination
  - first set has TTL =1
  - second set has TTL=2, etc.
  - unlikely port number
- when datagram in *n*th set arrives to nth router:
  - router discards datagram and sends source ICMP message (type 11, code 0)
  - ICMP message include name of router & IP address
- when ICMP message arrives, source records RTTs

stopping criteria:

- UDP segment eventually arrives at destination host
- destination returns ICMP "port unreachable" message (type 3, code 3)
- source stops



An Aside – Error Reporting (ICMP) and traceroute

On CDF machines try: *traceroute www.google.com*

```
traceroute to www.google.com (74.125.159.147), 30 hops max, 40 byte packets
 1 butler.syslab.sandbox (192.168.70.100) 0.103 ms 0.092 ms 0.082 ms
 2 foundry0.cs.toronto.edu (128.100.5.210) 2.146 ms 4.061 ms 5.977 ms
 3 sf-cs1.gw.utoronto.ca (128.100.1.253) 2.184 ms 2.175 ms 2.168 ms
 4 murus-gpb.gw.utoronto.ca (128.100.96.2) 2.146 ms 2.483 ms 3.037 ms
 5 skye2murus-blue.gw.utoronto.ca (128.100.200.210) 7.088 ms 7.207 ms 7.198 ms
 6 murus2skye-yellow.gw.utoronto.ca (128.100.200.217) 3.310 ms 11.325 ms 12.061 ms
 7 ut-hub-utoronto-if.gtanet.ca (205.211.94.129) 12.681 ms 2.541 ms 2.535 ms
 8 ORION-GTANET-RNE.DIST1-TORO.IP.orion.on.ca (66.97.23.57) 3.638 ms 4.391 ms 4.384 ms
 9 BRDR2-TORO-GE2-1.IP.orion.on.ca (66.97.16.121) 4.368 ms 4.729 ms 4.844 ms
10 74.125.51.233 (74.125.51.233) 12.459 ms 12.453 ms 12.808 ms
11 216.239.47.114 (216.239.47.114) 4.681 ms 4.795 ms 12.661 ms
12 209.85.250.111 (209.85.250.111) 23.666 ms 23.659 ms 13.226 ms
13 209.85.242.215 (209.85.242.215) 32.436 ms 32.431 ms 32.913 ms
14 72.14.232.213 (72.14.232.213) 33.537 ms 72.14.232.215 (72.14.232.215) 33.525 ms 72.14.232.213
   (72.14.232.213) 164.315 ms
15 209.85.254.14 (209.85.254.14) 45.864 ms 209.85.254.10 (209.85.254.10) 42.232 ms 209.85.254.6
   (209.85.254.6) 42.346 ms
16 yi-in-f147.google.com (74.125.159.147) 34.728 ms 34.727 ms 34.713 ms
```


An Aside – Error Reporting (ICMP) and traceroute

Internet Control Message Protocol

- Used by a router/end-host to report some types of error:
- E.g. Destination Unreachable: packet can't be forwarded to/towards its destination.
- E.g. Time Exceeded: TTL reached zero, or fragment didn't arrive in time. traceroute uses this error to its advantage.
- An ICMP message is an IP datagram, and is sent back to the source of the packet that caused the error.

Summary

- Shuttling data from one link to another
  - Bits, frames, packets, …
  - Repeaters/hubs, bridges/switches, routers, …
- Key ideas in switches
  - Cut-through switching
  - Self learning of the switch table
  - Spanning trees
- Internet Protocol
  - Addresses, subnets, CIDR
  - DNS, Traceroute, ICMP

Questions?