

Announcements

- Reading: Chapter 16
- Project #6 Due on next Tuesday
- Midterm #2 re-grade requests due today
- Project #5 style grading is done

Sending Data

- Data is split into *packets*
 - limited size units of sending information
 - can be
 - fixed sized (ATM)
 - variable size (Ethernet)
- Need to provide a destination for the packet
 - need to identify two levels of information
 - machine to send data to
 - comm abstraction (e.g. process) to get data
 - address may be:
 - a globally unique destination
 - for example every host has a unique id
 - may unique between hops
 - unique id between two switches

TCP/IP Protocol

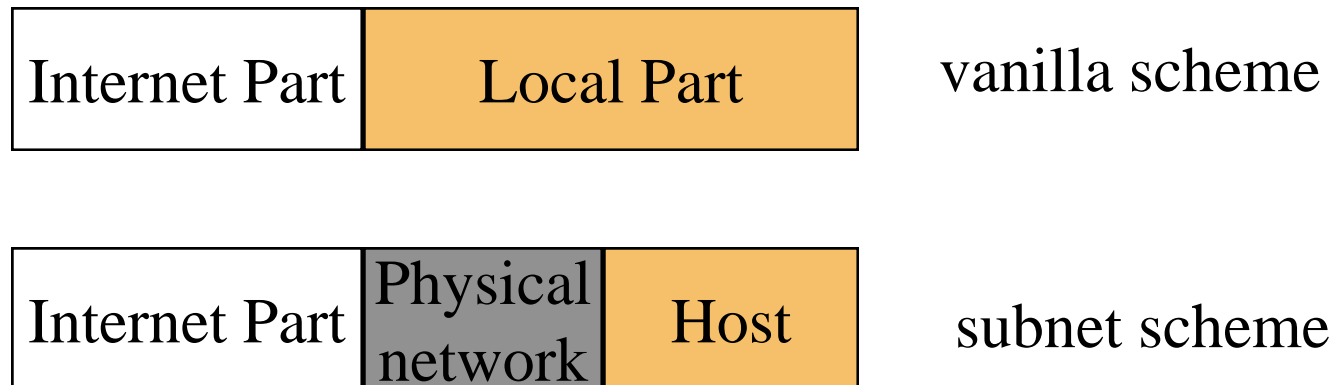
- Name for a family of Network and Transport layers
 - can run over many link layers:
 - Arpanet, Ethernet, Token Ring, SLIP/PPP, T1/T3, etc.
- IP - Internet Protocol
 - network level packet oriented protocol
 - 32 bit host addresses (dotted quad 128.8.128.84)
 - 8 bit protocol field (e.g. TCP, UDP, ICMP)
- TCP - Transmission Control Protocol
 - transport protocol
 - end-to-end reliable byte streams
 - provides ports for application specific end-points
- UDP- user datagram protocol
 - transport protocol
 - unreliable packet service
 - provides ports for application specific end-points

TCP/IP History

- Arpanet was the origin of today's Internet
 - started in 1969 to connect universities and DoD sites
 - early example of packet switched network
 - original links were 64kbps and 9.6kbps
- TCP/IP v4
 - started in use Jan 1, 1983
 - This was a *flag day*
 - all systems had to change to the new protocol at once
 - with the modern Internet this would be **hard** to do
- TCP/IP v6
 - Moves to 128 bit addresses
 - Simplified packet header

Subnet Addressing

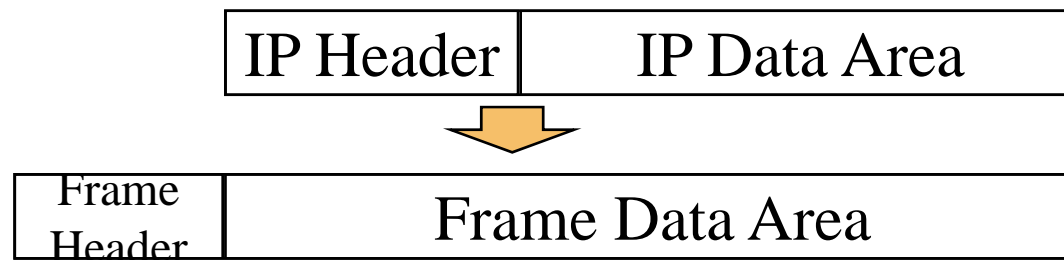
- Single site which has many physical networks
 - Only local routers know about all the physical nets
 - Site chooses part of address that distinguishes between physical networks
- subnet mask: splits the IP address into two parts
- Common Class B site mask 255.255.255.0
 - use 3rd byte to represent physical net
 - use 4th byte to represent host



Encapsulation

How do we send higher layer packets over lower layers?

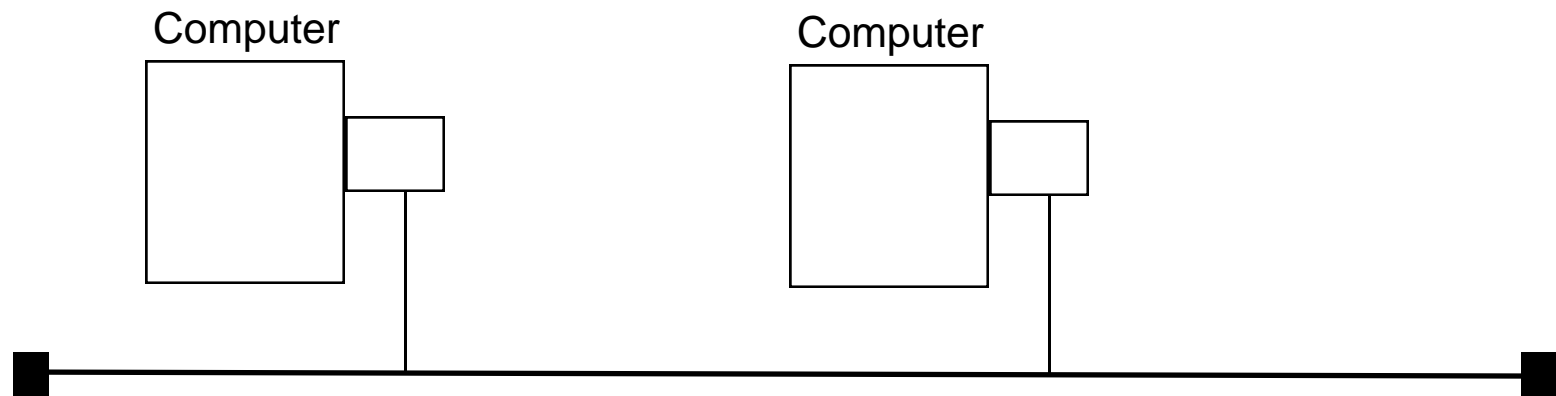
- Higher level info is opaque to lower layers
 - it's just data to be moved from one point to another



- Higher levels may support larger sizes than lower
 - could need to *fragment* a higher level packet
 - split into several lower level packets
 - need to re-assemble at the end
 - examples:
 - ATM cells are 48 bytes, but IP packets can be 64K
 - IP packets are 64K, but files are megabytes

Ethernet

- 10 Mbps (to 10 Gbps)
- milli-second latency
- limited to several kilometers in distance
- variable sized units of transmission
- Conceptually a bus based protocol
 - requests to use the network can collide
- addresses are 48 bits
 - unique to each interface



Switched Ethernet

- Logically it is still a bus
- Physically, it is a star configuration
 - the hub is at the center of the network
- Switches provide:
 - better control of hosts
 - possible to restrict traffic to only the desired target
 - can shutdown a host's connection at the hub if its Ethernet device is misbehaving
 - easier wiring
 - can use twisted pair wiring
- 100 Mbps/1 Gbps Ethernet
 - is only available with switches
- 10 Gbps Ethernet
 - Requires cat-6 (to 100 feet) or cat-7 wiring (to 100 meters)

Ethernet Collisions

- If one host is sending, other hosts must wait
 - called Carrier Sense with Multiple Access (CSMA)
- Possible for two hosts to try to send at once
 - each host can detect this event (cd- Collision Detection)
 - both hosts must re-send information
 - if they both try immediately, will collide again
 - instead each waits a random interval then tries again
- Only provides statistical guarantee of transmission
 - however, the probability of success is higher than the probability of hardware failures and other events

Routing

- How does a packet find its destination?
 - problem is called routing
- Several options:
 - source routing
 - end points know how to get everywhere
 - each packet is given a list of hops before it is sent
 - hop-by-hop
 - each host knows for each destination how to get one more hop in the right direction
- Can route packets:
 - per session
 - each packet in a connection takes same path
 - per packet
 - packets may take different routes
 - possible to have out of order delivery

Routing IP Datagrams

- **Direct Delivery:**

- a machine on a physical network can send a physical frame directly to another
- transmission of an IP datagram between two machines on a single physical network does not involve routers.
 - Sender encapsulates datagram into a physical frame, maps destination IP address to a physical address and sends frame directly to destination
- Sender knows that a machine is on a directly connected network
 - compare network portion of destination ID with own ID - if these match, the datagram can be sent directly
- Direct delivery can be viewed as the final step in any datagram transmission

Routing Datagrams (cont.)

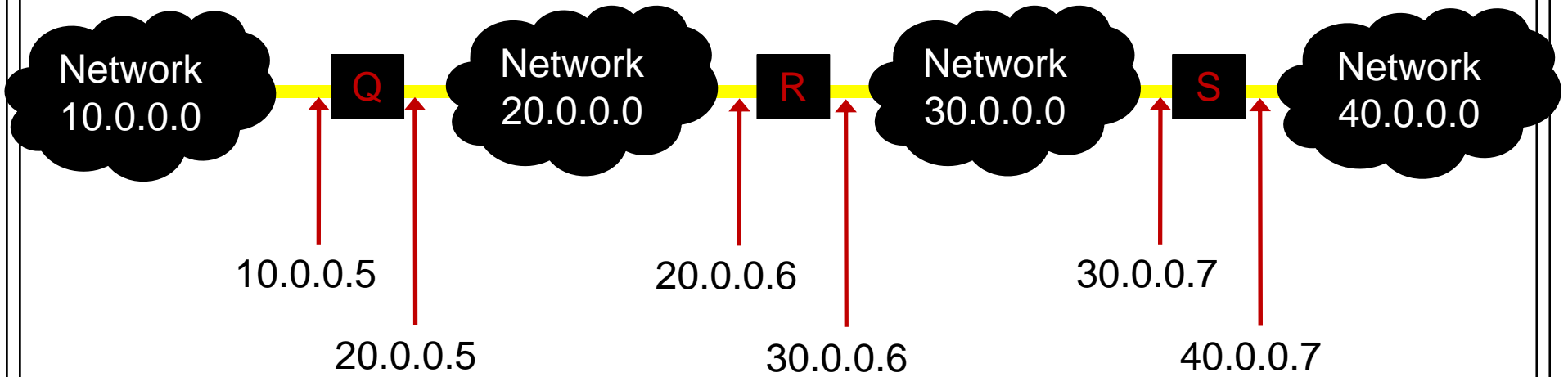
- Indirect Delivery

- sender must identify a router to which a datagram can be sent
- sending processor can reach a router on the sending processor's physical network (otherwise the network is isolated!)
- when frame reaches router, router extracts encapsulated datagram and IP software selects the next router
 - datagram is placed in a frame and sent off to the next router

Table Driven Routing

- Routing tables on each machine store information about possible destinations and how to reach them
- Routing tables only need to contain network prefixes, not full IP addresses
 - No need to include information about specific hosts
- Each entry in a routing table points to a router that can be reached across a single network
- Hosts and routers decide
 - can packet be directly sent?
 - which router should be responsible for a packet (if there is more than one on physical net)

Routing



20.0.0.0	<DIRECT>
30.0.0.0	<DIRECT>
10.0.0.0	20.0.0.5
40.0.0.0	30.0.0.7

Example from Comer book: Internetworking with TCP/IP: volume 1 [Third Edition]

Algorithm: RouteDatagram (Datagram, RoutingTable)

Extract destination IP address, D, from datagram
and compute network prefix N

If N matches any directly connected network
address

[Direct delivery]

Else if the table contains a host-specific route for D
[send datagram to next-hop specified in table]

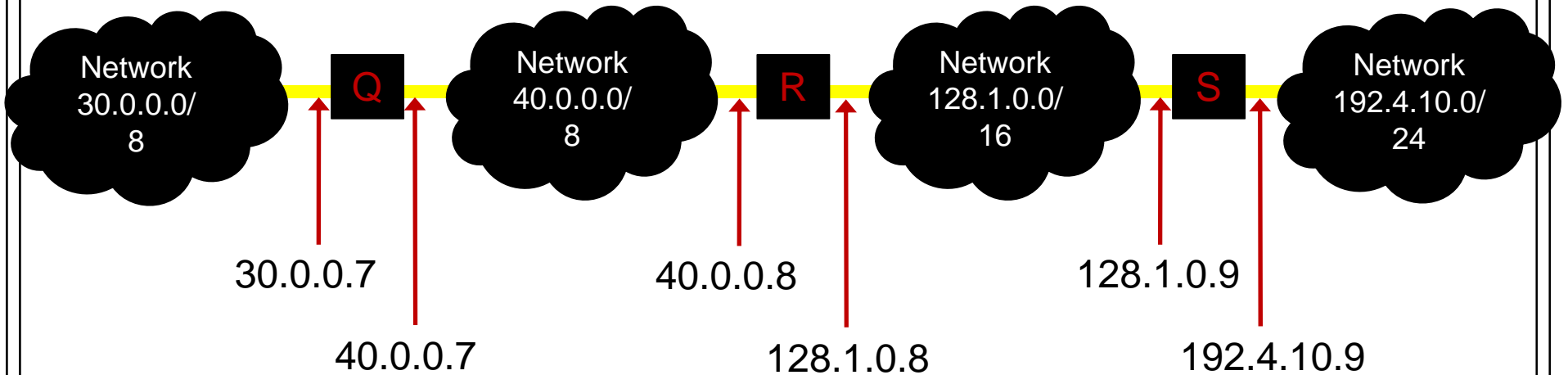
Else if the table contains a route for network N
[send datagram to next-hop specified in table]

Else if the table contains a default route
[send the datagram to the default route]

Else *declare a routing error*

Algorithm from Comer book: Internetworking with TCP/IP: volume 1 [Third Edition]

Routing (w/ subnets)



30.0.0.0	255.0.0.0	40.0.0.7
40.0.0.0	255.0.0.0	<DIRECT>
128.1.0.0	255.255.0.0	<DIRECT>
192.4.10.0	255.255.255.0	128.1.0.9

Mask field is used to extract the network part of an address during lookup.

If $((Mask[i] \& D) == Destination[i])$ forward to nextHop[i]

Consider a datagram destined for address 192.4.10.3 and the datagram arrives at router R

Extract destination IP address, D from datagram and compute network prefix N

$255.0.0.0 \& 192.4.10.3$ is not equal to 30.0.0.0

<same for entry 2 and 3>

$255.255.255.0 \& 192.4.10.3 = 192.4.10.0$
 → send to 128.1.0.9

Example from Comer book: Internetworking with TCP/IP: volume 1 [Third Edition]