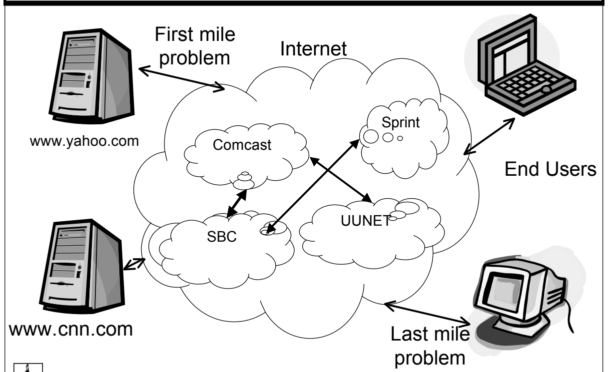
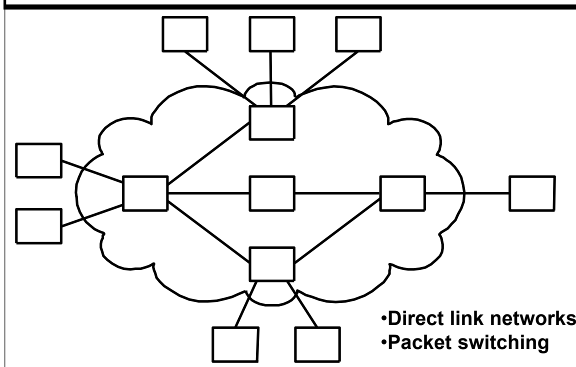
Recap

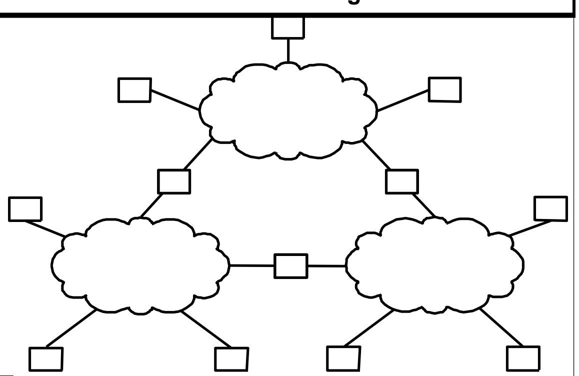


Recap



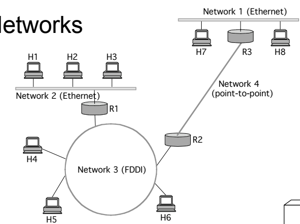
- Direct link networks
- Packet switching

Internetworking

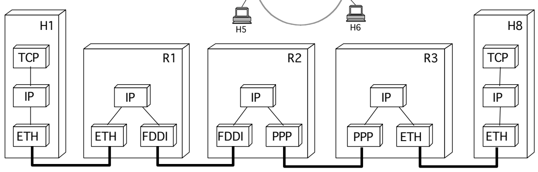


IP Internet

- Concatenation of Networks



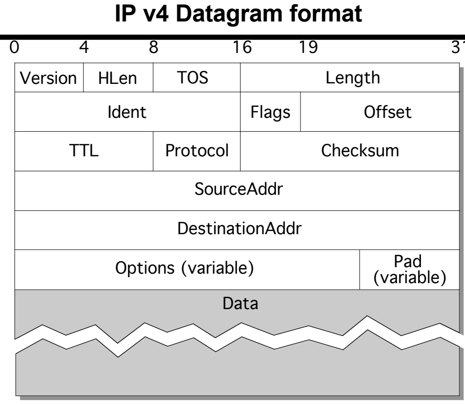
- Protocol Stack



Service Model

- Connectionless (datagram-based)
- Best-effort delivery (unreliable service)
  - packets are lost
  - packets are delivered out of order
  - duplicate copies of a packet are delivered
  - packets can be delayed for a long time

IP v4 Datagram format



IP v6 format

- http://ipv6.internet2.edu/boston/presentations/09-ipv6-under-the-hood.ppt
  - Version (4 bits) – only field to keep same position and name
  - Class (8 bits) – was Type of Service (TOS), renamed
  - Flow Label (20 bits) – new field
  - Payload Length (16 bits) – length of data, slightly different from total length
  - Next Header (8 bits) – type of the next header, new idea
  - Hop Limit (8 bits) – was time-to-live, renamed
  - Source address (128 bits)
  - Destination address (128 bits)

IPv4 and IPv6 headers

Version	Class	Flow Label	
Payload Length		Next Header	Hop Limit
Source Address			
Destination Address			

Version	IHL	Type of Service	Total Length	
Identification			Flags	Fragment Offset
Time-to-live		Protocol	Header Checksum	
Source Address				
Destination Address				
Options				Padding

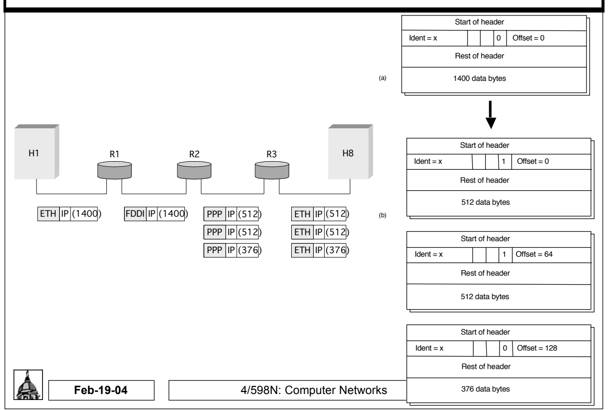
Basic Headers - IPV6

- Simplifications
  - Fixed length of all fields, not like old options field – IHL, or header length irrelevant
  - Remove Header Checksum – rely on checksums at other layers
  - No hop-by-hop fragmentation – fragment offset irrelevant
    - MTU discovery (IPv4 also support Path MTU discovery)
  - Add extension headers – next header type (sort of a protocol type, or replacement for options)
  - Basic Principle: Routers along the way should do minimal processing

Fragmentation and Reassembly

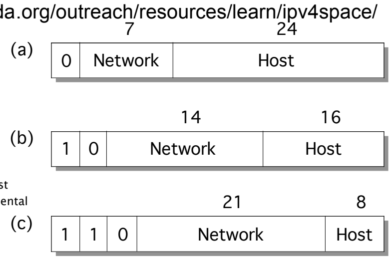
- Each network has some MTU
- Design decisions
  - fragment when necessary (MTU < Datagram)
  - try to avoid fragmentation at source host
  - re-fragmentation is possible
  - fragments are self-contained datagrams
  - use CS-PDU (not cells) for ATM
  - delay reassembly until destination host
  - do not recover from lost fragments

Example



Global Addresses

- Properties (www.iana.org/assignments/ipv4-address-space)
  - globally unique
  - hierarchical: network + host
  - http://www.caida.org/outreach/resources/learn/ipv4space/
- Dot Notation
  - 10.3.2.4
  - 128.96.33.81
  - 192.12.69.77



- Class D (224 – 239) Multicast
- Class E (240 – 247) Experimental

Datagram Forwarding

- Strategy
  - every datagram contains destination's address
  - if connected to destination network, then forward to host
  - if not directly connected, then forward to some router
  - forwarding table maps network number into next hop
  - each host has a default router
  - each router maintains a forwarding table

Example (R2)

Network number	Next
1	R3
2	R1
3	Interface 1
4	Interface 0

Address Translation

- Map IP addresses into physical addresses
  - destination host
  - next hop router
- Techniques
  - encode physical address in host part of IP address
  - table-based
- ARP
  - table of IP to physical address bindings
  - broadcast request if IP address not in table
  - target machine responds with its physical address
  - table entries are discarded if not refreshed

ARP Details

- Request Format
  - HardwareType: type of physical network (e.g., Ethernet)
  - ProtocolType: type of higher layer protocol (e.g., IP)
  - HLEN & PLEN: length of physical and protocol addresses
  - Operation: request or response
  - Source/Target-Physical/Protocol addresses
- Notes
  - table entries timeout in about 10 minutes
  - update table with source when you are the target
  - update table if already have an entry
  - do not refresh table entries upon reference

ARP Packet Format

0	8	16	31
Hardware type = 1		ProtocolType = 0x0800	
HLen = 48	PLen = 32	Operation	
SourceHardwareAddr (bytes 0 – 3)			
SourceHardwareAddr (bytes 4 – 5)		SourceProtocolAddr (bytes 0 – 1)	
SourceProtocolAddr (bytes 2 – 3)		TargetHardwareAddr (bytes 0 – 1)	
TargetHardwareAddr (bytes 2 – 5)			
TargetProtocolAddr (bytes 0 – 3)			

Sample arp table in darwin.cc.nd.edu

- arp -a

Net to Media Table: IPv4

```
Device   IP Address               Mask      Flags   Phys Addr
------ -------------------- --------------- ----- ---------------
hme0   eafs-e06.gw.nd.edu   255.255.255.255       00:d0:c0:d3:aa:40
hme0   bind.nd.edu          255.255.255.255       08:00:20:8a:5f:cf
hme0   honcho-jr.cc.nd.edu  255.255.255.255       00:b0:d0:82:83:7f
hme0   mail-vip.cc.nd.edu   255.255.255.255       02:e0:52:0c:56:c4
hme0   john.helios.nd.edu   255.255.255.255       08:00:20:85:db:c4
hme0   casper.helios.nd.edu 255.255.255.255       08:00:20:b1:f8:e1
hme0   pinky.helios.nd.edu  255.255.255.255       08:00:20:a9:88:30
```


ARP problems

- ARP trusts any response - no authentication method
  - Works great at home, how about Notre Dame
- Replies which do not correspond to requests are allowed to update cache in many instances
- New information must supercede old info

Internet Control Message Protocol (ICMP)

- Echo (ping)
  - /usr/src/sbin/ping/ping.c
- Redirect (from router to source host)
- Destination unreachable (protocol, port, or host)
- TTL exceeded (so datagrams don't cycle forever)
  - /usr/src/contrib/traceroute/traceroute.c
- Checksum failed
- Reassembly failed
- Cannot fragment

Virtual Private Networks (VPN) and tunnel

- Create a virtual network connecting different networks across the general Internet
  - Connect ND campus in South Bend and London to make them look like a single LAN even though packets traverse general IP network
- Use IP tunneling or IP over IP
  - Encapsulate IP packets inside other IP packets
  - Extra overhead