15-441 Computer Networking

Lecture 8 – IP Addressing & Packets

Aside: Interaction with Link Layer

- How does one find the Ethernet address of a IP host?
- ARP
  - Broadcast search for IP address
    - E.g., “who-has 128.2.184.45 tell 128.2.206.138” sent to Ethernet broadcast (all FF address)
  - Destination responds (only to requester using unicast) with appropriate 48-bit Ethernet address
    - E.g, “reply 128.2.184.45 is-at 0:d0:bc:f2:18:58” sent to 0:c0:4f:d:ed:c6

Caching ARP Entries

- Efficiency Concern
  - Would be very inefficient to use ARP request/reply every time need to send IP message to machine
- Each Host Maintains Cache of ARP Entries
  - Add entry to cache whenever get ARP response
  - Set timeout of ~20 minutes

ARP Cache Example

- Show using command “arp -a”

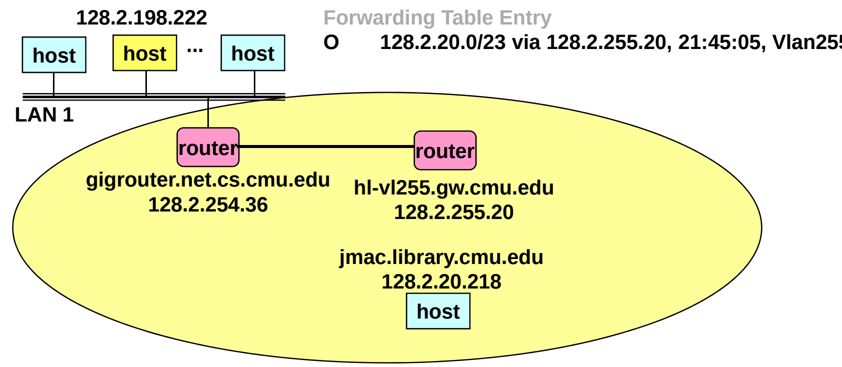
```
Interface: 128.2.222.198 on Interface 0x1000003
  Internet Address      Physical Address      Type
  128.2.20.218          00-b0-8e-83-df-50     dynamic
  128.2.102.129         00-b0-8e-83-df-50     dynamic
  128.2.194.66          00-02-b3-8a-35-bf     dynamic
  128.2.198.34          00-06-5b-f3-5f-42     dynamic
  128.2.203.3           00-90-27-3c-41-11     dynamic
  128.2.203.61          08-00-20-a6-ba-2b     dynamic
  128.2.205.192         00-60-08-1e-9b-fd     dynamic
  128.2.206.125         00-d0-b7-c5-b3-f3     dynamic
  128.2.206.139         00-a0-c9-98-2c-46     dynamic
  128.2.222.180         08-00-20-a6-ba-c3     dynamic
  128.2.242.182         08-00-20-a7-19-73     dynamic
  128.2.254.36          00-b0-8e-83-df-50     dynamic
```

ARP Cache Surprise

- How come 3 machines have the same MAC address?

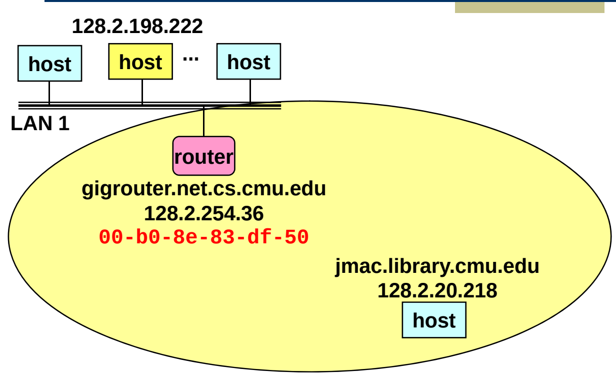
```
Interface: 128.2.222.198 on Interface 0x1000003
  Internet Address      Physical Address      Type
  128.2.20.218          00-b0-8e-83-df-50     dynamic
  128.2.102.129         00-b0-8e-83-df-50     dynamic
  128.2.194.66          00-02-b3-8a-35-bf     dynamic
  128.2.198.34          00-06-5b-f3-5f-42     dynamic
  128.2.203.3           00-90-27-3c-41-11     dynamic
  128.2.203.61          08-00-20-a6-ba-2b     dynamic
  128.2.205.192         00-60-08-1e-9b-fd     dynamic
  128.2.206.125         00-d0-b7-c5-b3-f3     dynamic
  128.2.206.139         00-a0-c9-98-2c-46     dynamic
  128.2.222.180         08-00-20-a6-ba-c3     dynamic
  128.2.242.182         08-00-20-a7-19-73     dynamic
  128.2.254.36          00-b0-8e-83-df-50     dynamic
```


CMU's Internal Network Structure



- CMU Uses Routing Internally
  - Maintains forwarding tables using OSPF
  - Most CMU hosts cannot be reached at link layer

Proxy ARP



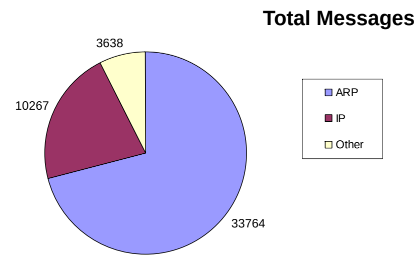
- Provides Link-Layer Connectivity Using IP Routing
  - Local router (gigrouter) sees ARP request
  - Uses IP addressing to locate host
  - Becomes "Proxy" for remote host
    - Using own MAC address
  - Requestor thinks that it is communicating directly with remote host

Things to keep in mind

- MAC ⇔ IP is not 1:1
- Tradeoff
  - Security?
  - Transparent backwards compatibility
- Encapsulation

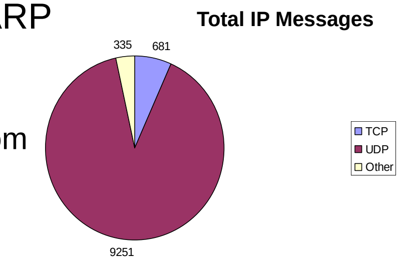
Monitoring Packet Traffic

- Experiment
  - Ran TCPDUMP for 15 minutes connected to CMU network
  - No applications running
    - But many background processes use network
  - Lots of ARP traffic (71% of total)
    - Average 37 ARP requests / second (why all from CS hosts?)
    - Only see responses from own machine (why?)



Monitoring Packet Traffic

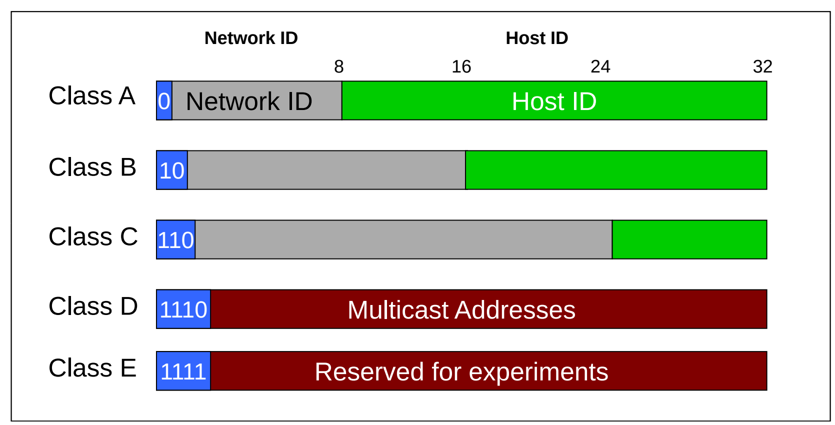
- Other Traffic
  - Mostly UDP
    - Encode low-level protocols such as bootp
  - Nothing very exciting (why?)
- Answers for UDP and ARP
  - On a switched network you only see broadcast traffic or traffic sent to/from you
  - TCP is never sent broadcast



Important Concepts

- Hierarchical addressing critical for scalable system
  - Don’t require everyone to know everyone else
  - Reduces number of updates when something changes
  - Interaction with routing tables

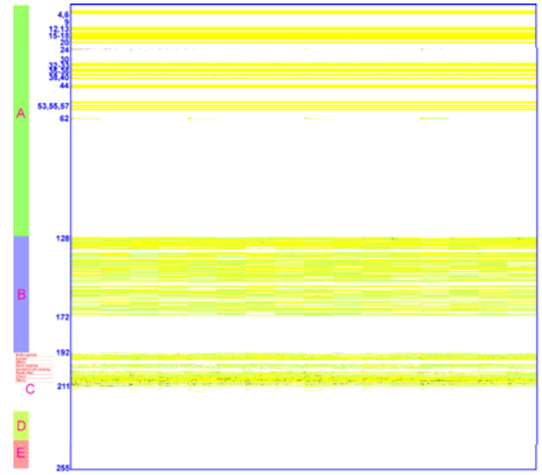
IP Address Classes
(Some are Obsolete)



IP Address Problem (1991)

- Address space depletion
  - In danger of running out of classes A and B
  - Why?
    - Class C too small for most domains
    - Very few class A – very careful about giving them out
    - Class B – greatest problem
- Class B sparsely populated
  - But people refuse to give it back
- Large forwarding tables
  - 2 Million possible class C groups

IP Address Utilization ('97)



http://www.caida.org/outreach/resources/learn/ipv4space/ -- broken

IP Address Utilization ('06)

http://xkcd.com/195/

IP Address Utilization ('06)

http://www.isi.edu/ant/address/browse/index.html

Classless Inter-Domain Routing (CIDR) – RFC1338

- Allows arbitrary split between network & host part of address
  - Do not use classes to determine network ID
  - Use common part of address as network number
  - E.g., addresses 192.4.16 - 192.4.31 have the first 20 bits in common. Thus, we use these 20 bits as the network number → 192.4.16/20
- Enables more efficient usage of address space (and router tables) → How?
  - Use single entry for range in forwarding tables
  - Combined forwarding entries when possible

CIDR Example

- Network is allocated 8 class C chunks, 200.10.0.0 to 200.10.7.255
  - Allocation uses 3 bits of class C space
  - Remaining 20 bits are network number, written as 201.10.0.0/21
- Replaces 8 class C routing entries with 1 combined entry
  - Routing protocols carry prefix with destination network address
  - Longest prefix match for forwarding

IP Addresses: How to Get One?

Network (network portion):

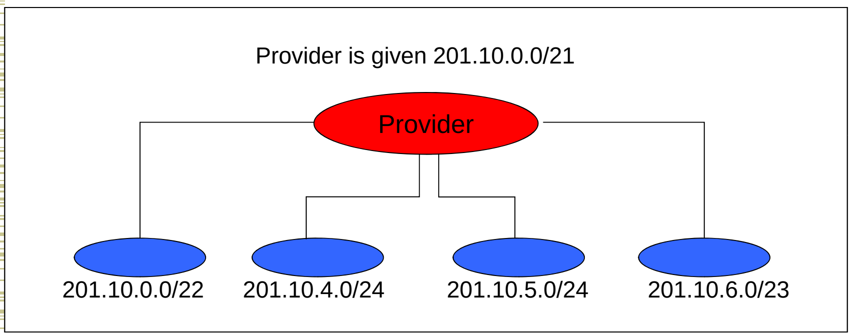
- Get allocated portion of ISP's address space:

ISP's block	11001000 00010111 00010000	00000000	200.23.16.0/20
Organization 0	11001000 00010111 00010000	00000000	200.23.16.0/23
Organization 1	11001000 00010111 00010010	00000000	200.23.18.0/23
Organization 2	11001000 00010111 00010100	00000000	200.23.20.0/23
...	.....	....	....
Organization 7	11001000 00010111 00011110	00000000	200.23.30.0/23

IP Addresses: How to Get One?

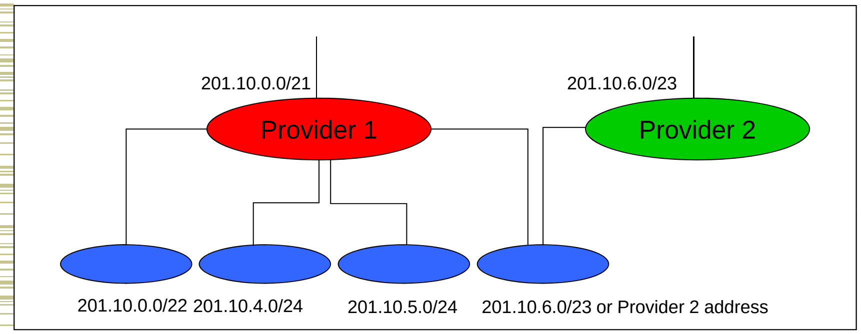
- How does an ISP get block of addresses?
  - From **Regional Internet Registries** (RIRs)
    - ARIN (North America, Southern Africa), APNIC (Asia-Pacific), RIPE (Europe, Northern Africa), LACNIC (South America)
- How about a single host?
  - Hard-coded by system admin in a file
  - DHCP: Dynamic Host Configuration Protocol: dynamically get address: "plug-and-play"
    - Host broadcasts "DHCP discover" msg
    - DHCP server responds with "DHCP offer" msg
    - Host requests IP address: "DHCP request" msg
    - DHCP server sends address: "DHCP ack" msg

CIDR Illustration



CIDR Implications

- Longest prefix match!!



CIDR

- Supernets
  - Assign adjacent net addresses to same org
  - Classless routing (CIDR)
- How does this help routing table?
  - Combine forwarding table entries whenever all nodes with same prefix share same hop

Aggregation with CIDR

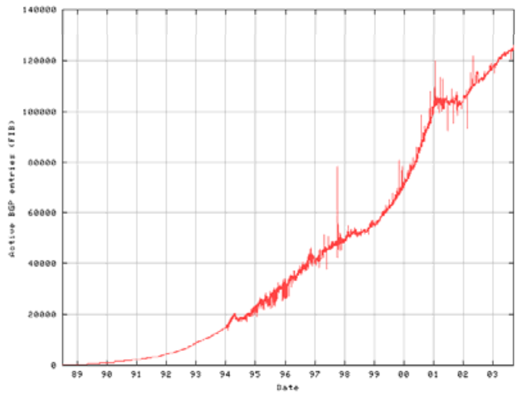
- Original Use: Aggregate Class C Addresses
- One organization assigned contiguous range of class C's
  - e.g., Microsoft given all addresses 207.46.192.X -- 207.46.255.X
  - Specify as CIDR address 207.46.192.0/18

0	8	16	24	31	
207	46	192	0		**Decimal**
cf	2e	c0	00		**Hexadecimal**
1100 1111	0010 1110	11xx xxxx	xxxx xxxx		**Binary**

Upper 18 bits frozen **Lower 14 bits arbitrary**

  - Represents 2^6 = 64 class C networks
- Use single entry in routing table
  - Just as if were single network address

Size of Complete Routing Table



- Source: www.cidr-report.org
- Shows that CIDR has kept # table entries in check
  - Currently require 124,894 entries for a complete table
  - Only required by backbone routers

Outline

- CIDR IP addressing
- Forwarding examples
- IP Packet Format

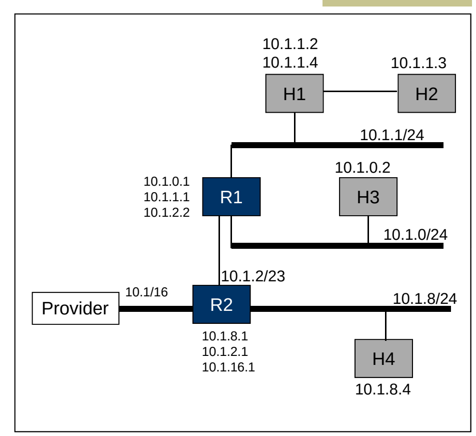
Host Routing Table Example

Destination	Gateway	Genmask	Iface
128.2.209.100	0.0.0.0	255.255.255.255	eth0
128.2.0.0	0.0.0.0	255.255.0.0	eth0
127.0.0.0	0.0.0.0	255.0.0.0	lo
0.0.0.0	128.2.254.36	0.0.0.0	eth0

- From "netstat –rn"
- Host 128.2.209.100 when plugged into CS ethernet
- Dest 128.2.209.100 → routing to same machine
- Dest 128.2.0.0 → other hosts on same ethernet
- Dest 127.0.0.0 → special loopback address
- Dest 0.0.0.0 → default route to rest of Internet
  - Main CS router: gigrouter.net.cs.cmu.edu (128.2.254.36)

Routing to the Network

- Packet to 10.1.1.3 arrives
- Path is R2 – R1 – H1 – H2

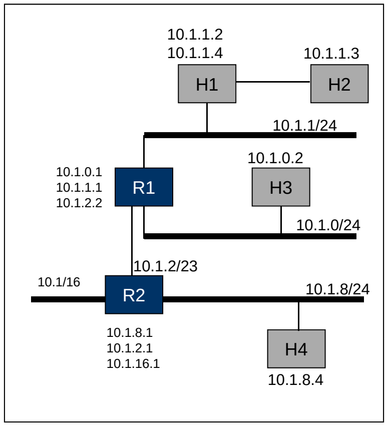


Routing Within the Subnet

- Packet to 10.1.1.3
- Matches 10.1.0.0/23

Routing table at R2

Destination	Next Hop	Interface
127.0.0.1	127.0.0.1	lo0
Default or 0/0	provider	10.1.16.1
10.1.8.0/24	10.1.8.1	10.1.8.1
10.1.2.0/23	10.1.2.1	10.1.2.1
10.1.0.0/23	10.1.2.2	10.1.2.1

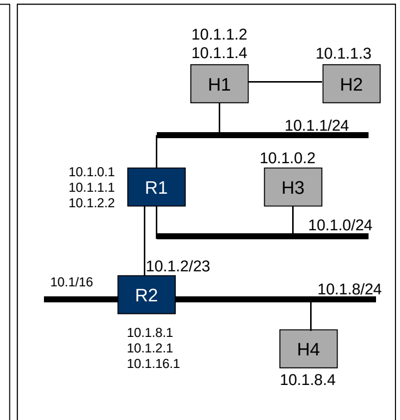


Routing Within the Subnet

- Packet to 10.1.1.3
- Matches 10.1.1.1/31
  - Longest prefix match

Routing table at R1

Destination	Next Hop	Interface
127.0.0.1	127.0.0.1	lo0
Default or 0/0	10.1.2.1	10.1.2.2
10.1.0.0/24	10.1.0.1	10.1.0.1
10.1.1.0/24	10.1.1.1	10.1.1.4
10.1.2.0/23	10.1.2.2	10.1.2.2
10.1.1.2/31	10.1.1.2	10.1.1.2



Aside: Interaction with Link Layer

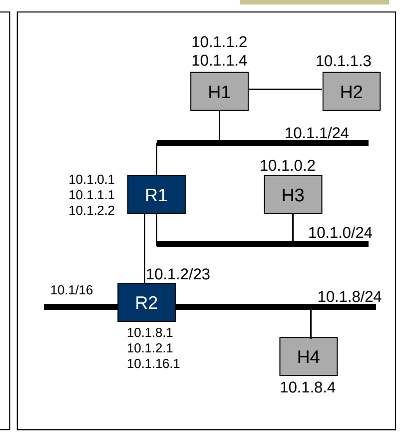
- How does one find the Ethernet address of a IP host?
- ARP
  - Broadcast search for IP address
    - E.g., “who-has 128.2.184.45 tell 128.2.206.138” sent to Ethernet broadcast (all FF address)
  - Destination responds (only to requester using unicast) with appropriate 48-bit Ethernet address
    - E.g, “reply 128.2.184.45 is-at 0:d0:bc:f2:18:58” sent to 0:c0:4f:d:ed:c6

Routing Within the Subnet

- Packet to 10.1.1.3
- Direct route
  - Longest prefix match

Routing table at H1

Destination	Next Hop	Interface
127.0.0.1	127.0.0.1	lo0
Default or 0/0	10.1.1.1	10.1.1.2
10.1.1.0/24	10.1.1.2	10.1.1.1
10.1.1.3/31	10.1.1.2	10.1.1.2



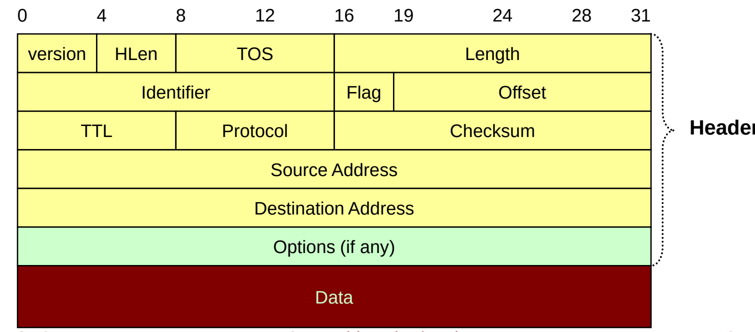
Outline

- CIDR IP addressing
- Forwarding examples
- IP Packet Format

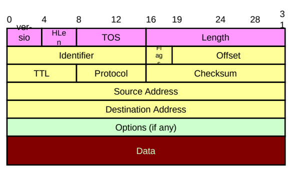
IP Service Model

- Low-level communication model provided by Internet
- Datagram
  - Each packet self-contained
    - All information needed to get to destination
    - No advance setup or connection maintenance
  - Analogous to letter or telegram

IPv4 Packet Format



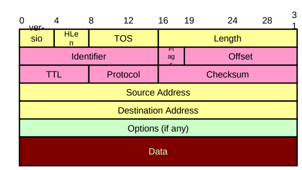
IPv4 Header Fields



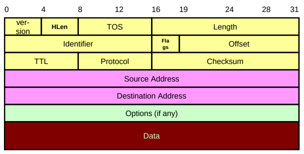
- Version: IP Version
  - 4 for IPv4
- HLen: Header Length
  - 32-bit words (typically 5)
- TOS: Type of Service
  - Priority information
- Length: Packet Length
  - Bytes (including header)
- Header format can change with versions
  - First byte identifies version
- Length field limits packets to 65,535 bytes
  - In practice, break into much smaller packets for network performance considerations

IPv4 Header Fields

- Identifier, flags, fragment offset → used primarily for fragmentation
- Time to live
  - Must be decremented at each router
  - Packets with TTL=0 are thrown away
  - Ensure packets exit the network
- Protocol
  - Demultiplexing to higher layer protocols
  - TCP = 6, ICMP = 1, UDP = 17…
- Header checksum
  - Ensures some degree of header integrity
  - Relatively weak – 16 bit
- Options
  - E.g. Source routing, record route, etc.
  - Performance issues
    - Poorly supported



IPv4 Header Fields



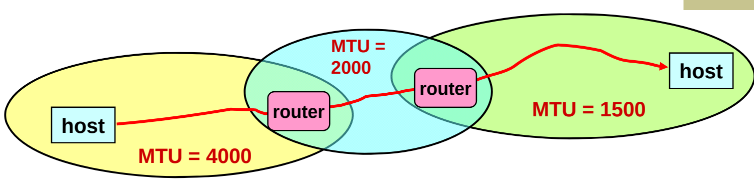
- Source Address
  - 32-bit IP address of sender
- Destination Address
  - 32-bit IP address of destination

- Like the addresses on an envelope
- Globally unique identification of sender & receiver

IP Delivery Model

- *Best effort service*
  - Network will do its best to get packet to destination
- Does NOT guarantee:
  - Any maximum latency or even ultimate success
  - Sender will be informed if packet doesn't make it
  - Packets will arrive in same order sent
  - Just one copy of packet will arrive
- Implications
  - Scales very well
  - Higher level protocols must make up for shortcomings
    - Reliably delivering ordered sequence of bytes → TCP
  - Some services not feasible
    - Latency or bandwidth guarantees

IP Fragmentation



- Every network has own Maximum Transmission Unit (MTU)
  - Largest IP datagram it can carry within its own packet frame
    - E.g., Ethernet is 1500 bytes
  - Don't know MTUs of all intermediate networks in advance
- IP Solution
  - When hit network with small MTU, fragment packets

Reassembly

- Where to do reassembly?
  - End nodes or at routers?

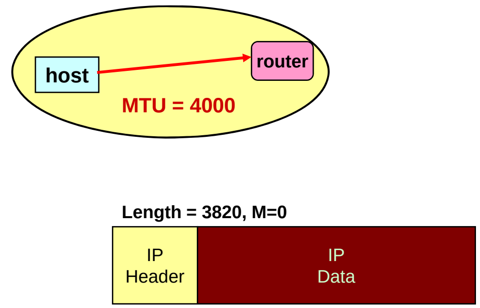
- End nodes
  - Avoids unnecessary work where large packets are fragmented multiple times
  - If any fragment missing, delete entire packet

- Dangerous to do at intermediate nodes
  - How much buffer space required at routers?
  - What if routes in network change?
    - Multiple paths through network
    - All fragments only required to go through destination

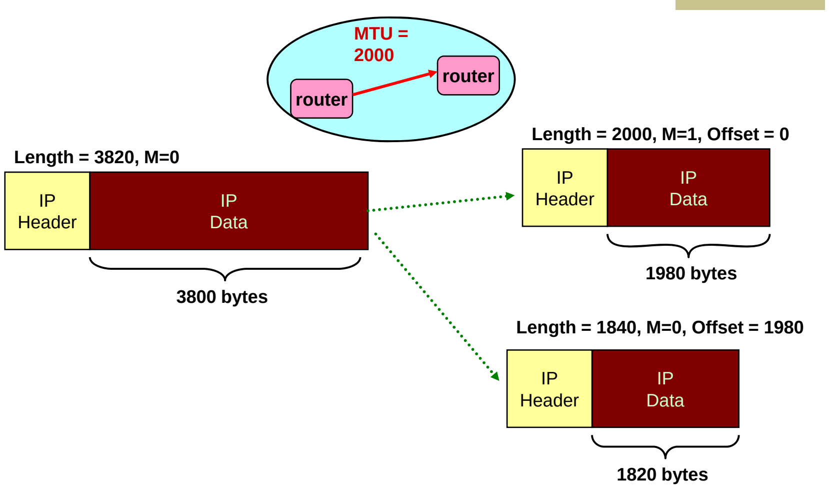
Fragmentation Related Fields

- Length
  - Length of IP fragment
- Identification
  - To match up with other fragments
- Flags
  - Don't fragment flag
  - More fragments flag
- Fragment offset
  - Where this fragment lies in entire IP datagram
  - Measured in 8 octet units (13 bit field)

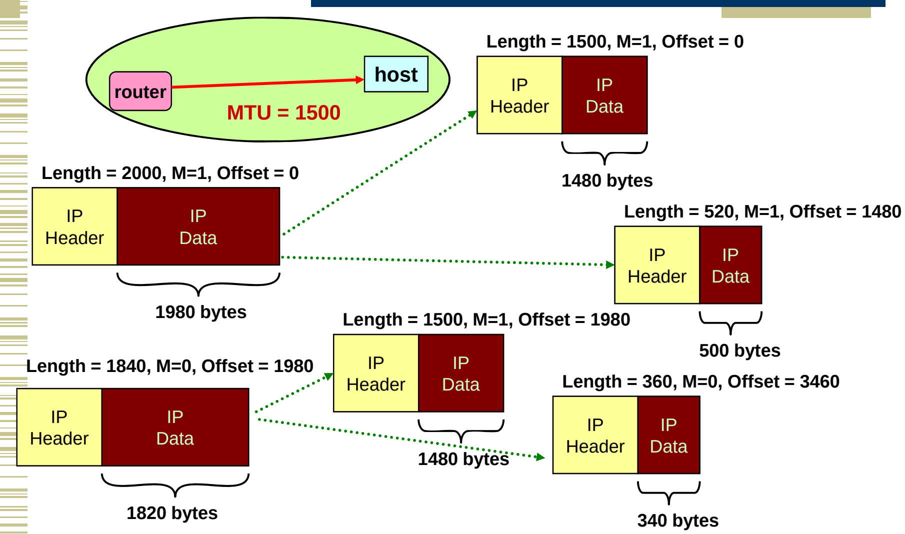
IP Fragmentation Example #1



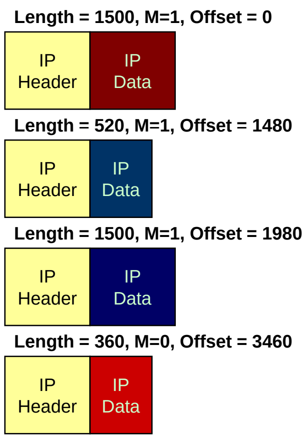
IP Fragmentation Example #2



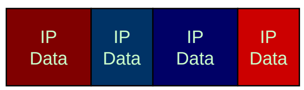
IP Fragmentation Example #3



IP Reassembly



- Fragments might arrive out-of-order
  - Don't know how much memory required until receive final fragment
- Some fragments may be duplicated
  - Keep only one copy
- Some fragments may never arrive
  - After a while, give up entire process



Fragmentation and Reassembly Concepts

- Demonstrates many Internet concepts
- Decentralized
  - Every network can choose MTU
- Connectionless
  - Each (fragment of) packet contains full routing information
  - Fragments can proceed independently and along different routes
- Best effort
  - Fail by dropping packet
  - Destination can give up on reassembly
  - No need to signal sender that failure occurred
- Complex endpoints and simple routers
  - Reassembly at endpoints

Fragmentation is Harmful

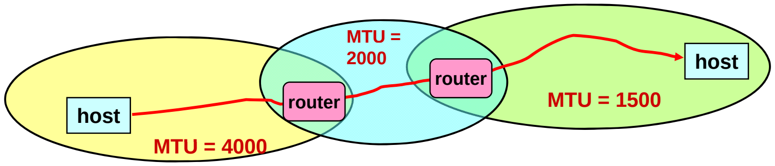
- Uses resources poorly
  - Forwarding costs per packet
  - Best if we can send large chunks of data
  - Worst case: packet just bigger than MTU
- Poor end-to-end performance
  - Loss of a fragment

- Path MTU discovery protocol → determines minimum MTU along route
  - Uses ICMP error messages
- Common theme in system design
  - Assure correctness by implementing complete protocol
  - Optimize common cases to avoid full complexity

Internet Control Message Protocol (ICMP)

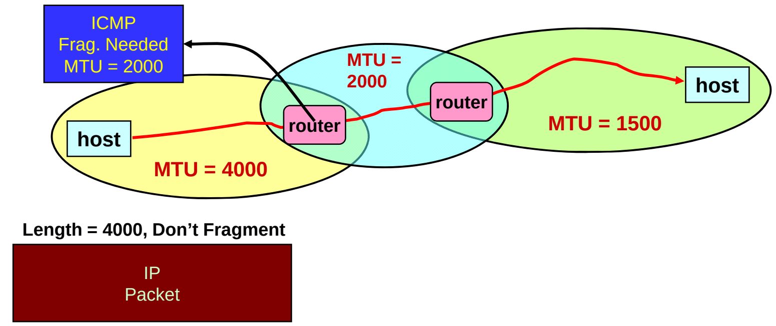
- Short messages used to send error & other control information
- Examples
  - Ping request / response
    - Can use to check whether remote host reachable
  - Destination unreachable
    - Indicates how packet got & why couldn't go further
  - Flow control
    - Slow down packet delivery rate
  - Redirect
    - Suggest alternate routing path for future messages
  - Router solicitation / advertisement
    - Helps newly connected host discover local router
  - Timeout
    - Packet exceeded maximum hop limit

IP MTU Discovery with ICMP

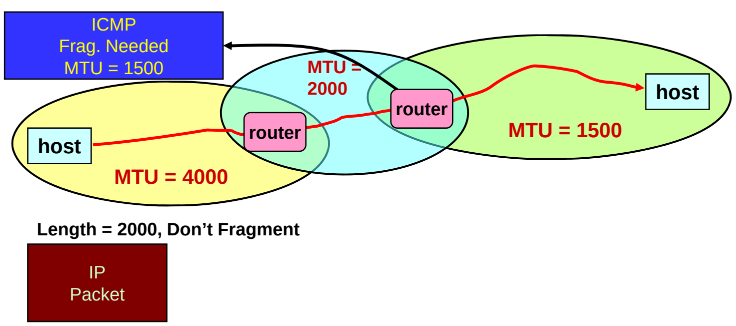


- Typically send series of packets from one host to another
- Typically, all will follow same route
  - Routes remain stable for minutes at a time
- Makes sense to determine path MTU before sending real packets
- Operation
  - Send max-sized packet with “do not fragment” flag set
  - If encounters problem, ICMP message will be returned
    - “Destination unreachable: Fragmentation needed”
    - Usually indicates MTU encountered

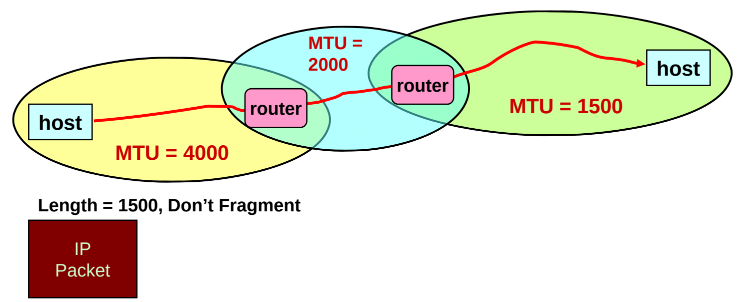
IP MTU Discovery with ICMP



IP MTU Discovery with ICMP



IP MTU Discovery with ICMP



- When successful, no reply at IP level
  - “No news is good news”
- Higher level protocol might have some form of acknowledgement

Important Concepts

- Base-level protocol (IP) provides minimal service level
  - Allows highly decentralized implementation
  - Each step involves determining next hop
  - Most of the work at the endpoints
- ICMP provides low-level error reporting

- IP forwarding → global addressing, alternatives, lookup tables
- IP addressing → hierarchical, CIDR
- IP service → best effort, simplicity of routers
- IP packets → header fields, fragmentation, ICMP

Next Lecture

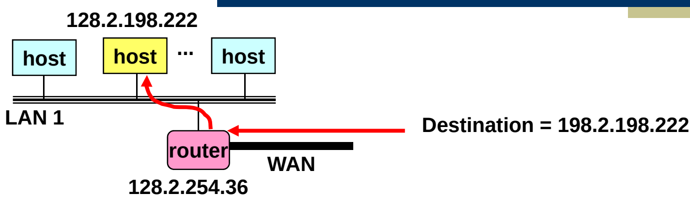
- How do forwarding tables get built?
- Routing protocols
  - Distance vector routing
  - Link state routing

EXTRA SLIDES

Some Special IP Addresses

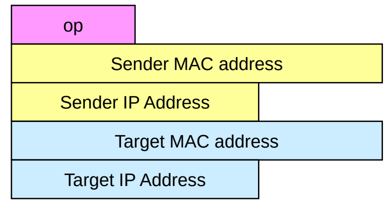
- 127.0.0.1: local host (a.k.a. the loopback address
- Host bits all set to 0: network address
- Host bits all set to 1: broadcast address

Finding a Local Machine



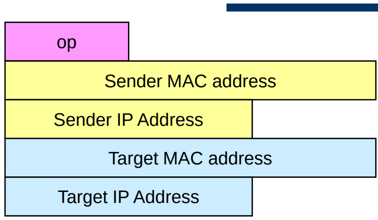
- Routing Gets Packet to Correct Local Network
  - Based on IP address
  - Router sees that destination address is of local machine
- Still Need to Get Packet to Host
  - Using link-layer protocol
  - Need to know hardware address
- Same Issue for Any Local Communication
  - Find local machine, given its IP address

Address Resolution Protocol (ARP)



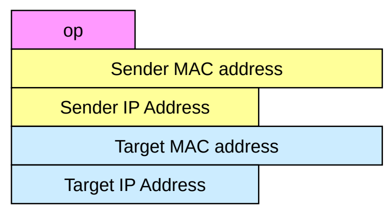
- op: Operation
  - 1: request
  - 2: reply
- Sender
  - Host sending ARP message
- Target
  - Intended receiver of message
- Diagrammed for Ethernet (6-byte MAC addresses)
- Low-Level Protocol
  - Operates only within local network
  - Determines mapping from IP address to hardware (MAC) address
  - Mapping determined dynamically
    - No need to statically configure tables
    - Only requirement is that each host know its own IP address

ARP Request



- op: Operation
  - 1: request
- Sender
  - Host that wants to determine MAC address of another machine
- Target
  - Other machine
- Requestor
  - Fills in own IP and MAC address as "sender"
    - Why include its MAC address?
- Mapping
  - Fills desired host IP address in target IP address
- Sending
  - Send to MAC address `ff:ff:ff:ff:ff:ff`
    - Ethernet broadcast

ARP Reply



- op: Operation
  - 2: reply
- Sender
  - Host with desired IP address
- Target
  - Original requestor
- Responder becomes "sender"
  - Fill in own IP and MAC address
  - Set requestor as target
  - Send to requestor's MAC address

ARP Example

Time	Source MAC	Dest MAC
09:37:53.729185	**0:2:b3:8a:35:bf**	**ff:ff:ff:ff:ff:ff 0806 60:**
arp who-has 128.2.222.198 tell 128.2.194.66		
09:37:53.729202	**0:3:47:b8:e5:f3**	**0:2:b3:8a:35:bf 0806 42:**
arp reply 128.2.222.198 is-at 0:3:47:b8:e5:f3		

- Exchange Captured with windump
  - Windows version of tcpdump
- Requestor:
  - blackhole-ad.scs.cs.cmu.edu (128.2.194.66)
  - MAC address 0:2:b3:8a:35:bf
- Desired host:
  - bryant-tp2.vlsi.cs.cmu.edu (128.2.222.198)
  - MAC address 0:3:47:b8:e5:f3