Outline

- Review – ARP and switches puzzle
- CIDR IP addressing
- Forwarding examples
- IP Packet Format

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

<table>
<thead>
<tr>
<th>Internet Address</th>
<th>Physical Address</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>128.2.20.218</td>
<td>00-b0-8e-83-df-50</td>
<td>dynamic</td>
</tr>
<tr>
<td>128.2.102.129</td>
<td>00-b0-8e-83-df-50</td>
<td>dynamic</td>
</tr>
<tr>
<td>128.2.194.66</td>
<td>00-02-b3-8a-35-bf</td>
<td>dynamic</td>
</tr>
<tr>
<td>128.2.198.34</td>
<td>00-06-5b-f3-5f-42</td>
<td>dynamic</td>
</tr>
<tr>
<td>128.2.203.3</td>
<td>00-90-27-3c-41-11</td>
<td>dynamic</td>
</tr>
<tr>
<td>128.2.203.61</td>
<td>08-00-20-a6-ba-2b</td>
<td>dynamic</td>
</tr>
<tr>
<td>128.2.205.192</td>
<td>00-60-08-1a-9b-fd</td>
<td>dynamic</td>
</tr>
<tr>
<td>128.2.206.125</td>
<td>00-d0-b7-c5-b3-f3</td>
<td>dynamic</td>
</tr>
<tr>
<td>128.2.206.139</td>
<td>00-a0-c9-98-2c-46</td>
<td>dynamic</td>
</tr>
<tr>
<td>128.2.222.180</td>
<td>08-00-20-a6-ba-c3</td>
<td>dynamic</td>
</tr>
<tr>
<td>128.2.242.182</td>
<td>08-00-20-a7-19-73</td>
<td>dynamic</td>
</tr>
<tr>
<td>128.2.254.36</td>
<td>00-b0-8e-83-df-50</td>
<td>dynamic</td>
</tr>
</tbody>
</table>

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?)

- 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

ARP Cache Example

- Show using command “arp -a”

Interface: 128.2.222.198 on Interface 0x1000003

<table>
<thead>
<tr>
<th>Internet Address</th>
<th>Physical Address</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>128.2.20.218</td>
<td>00-b0-8e-83-df-50</td>
<td>dynamic</td>
</tr>
<tr>
<td>128.2.102.129</td>
<td>00-b0-8e-83-df-50</td>
<td>dynamic</td>
</tr>
<tr>
<td>128.2.194.66</td>
<td>00-02-b3-8a-35-bf</td>
<td>dynamic</td>
</tr>
<tr>
<td>128.2.198.34</td>
<td>00-06-5b-f3-5f-42</td>
<td>dynamic</td>
</tr>
<tr>
<td>128.2.203.3</td>
<td>00-90-27-3c-41-11</td>
<td>dynamic</td>
</tr>
<tr>
<td>128.2.203.61</td>
<td>08-00-20-a6-ba-2b</td>
<td>dynamic</td>
</tr>
<tr>
<td>128.2.205.192</td>
<td>00-60-08-1a-9b-fd</td>
<td>dynamic</td>
</tr>
<tr>
<td>128.2.206.125</td>
<td>00-d0-b7-c5-b3-f3</td>
<td>dynamic</td>
</tr>
<tr>
<td>128.2.206.139</td>
<td>00-a0-c9-98-2c-46</td>
<td>dynamic</td>
</tr>
<tr>
<td>128.2.222.180</td>
<td>08-00-20-a6-ba-c3</td>
<td>dynamic</td>
</tr>
<tr>
<td>128.2.242.182</td>
<td>08-00-20-a7-19-73</td>
<td>dynamic</td>
</tr>
<tr>
<td>128.2.254.36</td>
<td>00-b0-8e-83-df-50</td>
<td>dynamic</td>
</tr>
</tbody>
</table>
ARP Cache Surprise

• How come 3 machines have the same MAC address?

<table>
<thead>
<tr>
<th>Interface</th>
<th>Internet Address</th>
<th>Physical Address</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>128.2.20.218</td>
<td>00-b0-8e-83-df-50</td>
<td>dynamic</td>
<td></td>
</tr>
<tr>
<td>128.2.102.129</td>
<td>00-b0-8e-83-df-50</td>
<td>dynamic</td>
<td></td>
</tr>
<tr>
<td>128.2.194.66</td>
<td>00-02-b3-8a-35-bf</td>
<td>dynamic</td>
<td></td>
</tr>
<tr>
<td>128.2.198.34</td>
<td>00-06-5b-f3-5f-42</td>
<td>dynamic</td>
<td></td>
</tr>
<tr>
<td>128.2.203.3</td>
<td>00-90-27-3c-41-11</td>
<td>dynamic</td>
<td></td>
</tr>
<tr>
<td>128.2.203.61</td>
<td>08-00-20-a6-ba-2b</td>
<td>dynamic</td>
<td></td>
</tr>
<tr>
<td>128.2.205.192</td>
<td>00-60-01-9b-7d-f0</td>
<td>dynamic</td>
<td></td>
</tr>
<tr>
<td>128.2.206.125</td>
<td>00-0b-7c-53-f3</td>
<td>dynamic</td>
<td></td>
</tr>
<tr>
<td>128.2.206.139</td>
<td>00-a0-c9-98-2c-46</td>
<td>dynamic</td>
<td></td>
</tr>
<tr>
<td>128.2.222.180</td>
<td>08-00-20-a6-ba-c3</td>
<td>dynamic</td>
<td></td>
</tr>
<tr>
<td>128.2.242.182</td>
<td>08-00-20-a7-19-73</td>
<td>dynamic</td>
<td></td>
</tr>
<tr>
<td>128.2.254.36</td>
<td>00-b0-8e-83-df-50</td>
<td>dynamic</td>
<td></td>
</tr>
</tbody>
</table>

CMU’s Internal Network Structure

• CMU routers kept reasonable tables

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

Outline

• Review – ARP and switches puzzle
  • CIDR IP addressing
  • Forwarding examples
  • IP Packet Format
IP Address Classes (Some are Obsolete)

<table>
<thead>
<tr>
<th>Class</th>
<th>Network ID</th>
<th>Host ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class A</td>
<td>1</td>
<td>24</td>
</tr>
<tr>
<td>Class B</td>
<td>16</td>
<td>24</td>
</tr>
<tr>
<td>Class C</td>
<td>110</td>
<td>24</td>
</tr>
<tr>
<td>Class D</td>
<td>1110</td>
<td>24</td>
</tr>
<tr>
<td>Class E</td>
<td>1111</td>
<td>24</td>
</tr>
</tbody>
</table>

Multicast Addresses

Reserved for experiments

IP Address Utilization ('97)

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

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

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
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
  - Represents $2^{18} = 64$ class C networks
- Use single entry in routing table
- Just as if were single network address

CIDR Illustration

Provider is given 201.10.0.0/21

CIDR Implications

- Longest prefix match!!

IP Addresses: How to Get One?

Network (network portion):
- Get allocated portion of ISP's address space:
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
    - 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

IP Address Utilization (‘06)
http://xkcd.com/195/

IP Address Utilization (‘06)
http://www.isi.edu/ant/address/browse/index.html

IP Address Utilization (late ‘10)
http://www.isi.edu/ant/address/browse/index.html
What Now?

• Last /8 given to RIR in 1/2011
• Mitigation
  • Reclaim addresses (e.g. Stanford gave back class A in 2000)
  • More NAT?
• Resale markets
• Slow down allocation from RIRs to LIRs (i.e. ISPs)
• IPv6?
Outline

- Review – ARP and switches puzzle
- CIDR IP addressing
- Forwarding examples
- IP Packet Format

Host Routing Table Example

<table>
<thead>
<tr>
<th>Destination</th>
<th>Gateway</th>
<th>Genmask</th>
<th>Iface</th>
</tr>
</thead>
<tbody>
<tr>
<td>128.2.209.100</td>
<td>0.0.0.0</td>
<td>255.255.255.255</td>
<td>eth0</td>
</tr>
<tr>
<td>128.2.0.0</td>
<td>0.0.0.0</td>
<td>255.255.0.0</td>
<td>eth0</td>
</tr>
<tr>
<td>127.0.0.0</td>
<td>0.0.0.0</td>
<td>255.0.0.0</td>
<td>lo</td>
</tr>
<tr>
<td>0.0.0.0</td>
<td>128.2.254.36</td>
<td>0.0.0.0</td>
<td>eth0</td>
</tr>
</tbody>
</table>

- 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

<table>
<thead>
<tr>
<th>Destination</th>
<th>Next Hop</th>
<th>Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>127.0.0.1</td>
<td>127.0.0.1</td>
<td>lo</td>
</tr>
<tr>
<td>Default or 0/0</td>
<td>provider</td>
<td>10.1.16.1</td>
</tr>
<tr>
<td>10.1.0.0/24</td>
<td>10.1.1.1</td>
<td>10.1.1.1</td>
</tr>
<tr>
<td>10.1.2.0/23</td>
<td>10.1.2.1</td>
<td>10.1.2.1</td>
</tr>
<tr>
<td>10.1.0.0/23</td>
<td>10.1.2.2</td>
<td>10.1.2.2</td>
</tr>
<tr>
<td>10.1.16.1</td>
<td>10.1.16.1</td>
<td>10.1.16.1</td>
</tr>
</tbody>
</table>

H1
H2
R1
R2
H3
Routing Within the Subnet

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

Routing table at R1

<table>
<thead>
<tr>
<th>Destination</th>
<th>Next Hop</th>
<th>Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>127.0.0.1</td>
<td>127.0.0.1</td>
<td>lo0</td>
</tr>
<tr>
<td>Default or 0</td>
<td>10.1.2.1</td>
<td>10.1.2.2</td>
</tr>
<tr>
<td>10.1.0/24</td>
<td>10.1.0.1</td>
<td>10.1.0.1</td>
</tr>
<tr>
<td>10.1.1/24</td>
<td>10.1.1.1</td>
<td>10.1.1.1</td>
</tr>
<tr>
<td>10.1.2/23</td>
<td>10.1.2.2</td>
<td>10.1.2.2</td>
</tr>
<tr>
<td>10.1.1.2/31</td>
<td>10.1.1.2</td>
<td>10.1.1.1</td>
</tr>
</tbody>
</table>

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 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

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

- Host -> Router
  - MTU = 4000
  - Length = 3820, M=0

IP Fragmentation Example #2

- Router -> Router
  - MTU = 2000
  - Length = 3820, M=0
  - 3800 bytes

- Router
  - IP Header
  - IP Data
  - Length = 2000, M=1, Offset = 0
  - 1980 bytes

- Router
  - IP Header
  - IP Data
  - Length = 1840, M=0, Offset = 1980
  - 1820 bytes
IP Fragmentation Example #3

- Length = 2000, M=1, Offset = 0
  - 1980 bytes
- Length = 1840, M=0, Offset = 1980
  - 500 bytes
- Length = 360, M=0, Offset = 3460
  - 340 bytes

IP Reassembly

- Length = 1500, M=1, Offset = 0
- Length = 520, M=1, Offset = 1480
- Length = 1500, M=1, Offset = 1980
- Length = 360, M=0, Offset = 3460

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 \( \rightarrow \) 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

**Diagram: IP MTU Discovery with ICMP**

- MTU = 4000
- MTU = 1500
- MTU = 2000
- IP Packet
- Length = 4000, Don’t Fragment
- "IP Packet"
- "Frag. Needed"
- "MTU = 2000"

- MTU = 2000
- MTU = 1500
- MTU = 2000
- IP Packet
- Length = 2000, Don’t Fragment
- "IP Packet"
### 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

![Diagram of IP MTU Discovery with ICMP]

### 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

### Now for some really bad jokes...

- I tried to come up with an IPv4 joke, but the good ones were all already exhausted.
- The sad thing about IPv6 jokes is that almost no one understands them and no one is using them yet.
- WHO HAS any ARP jokes?
- Fragmentation jokes...are always......told in parts.
- An IPv4 address space walks in to a bar, "A strong CIDR please. I'm exhausted."