Announcements

• Midterm grading 2/3s done, completed by Friday
• I am away next week with limited email
  – ACM Internet Measurement Conference
• Monday Oct 23: Dilip (Routing)
• Weds Oct 25: Prof. Scott Shenker (BGP)
• Dilip takes over the Weds section from Sukun on Weds Oct 25

Except picture w/o the beard
Goals of Today’s Lecture

• Bootstrapping an end host
  – Learning its own configuration parameters (DHCP)
  – Learning the link-layer addresses of other nodes (ARP)

• Network control messages
  – Error/status reporting
  – Monitoring
  – Internet Control Message Protocol (ICMP)

How To Bootstrap an End Host?

• What IP address the host should use?
• What local DNS server to use?
• How to send packets to remote destinations?
• How to tell which destinations are local?
• How do we address them using local network?

??? 1.2.3.7 1.2.3.156
1.2.3.0/23

1.2.3.19

5.6.7.0/24

host host host host host

DNS DNS DNS

router router router
Avoiding Manual Configuration

• **Dynamic Host Configuration Protocol (DHCP)**
  – End host learns how to send packets
  – Learn IP address, DNS servers, "gateway", what's local

• **Address Resolution Protocol (ARP)**
  – For local destinations, learn mapping between IP address and MAC address

Key Ideas in Both Protocols

• **Broadcasting**: when in doubt, shout!
  – Broadcast query to all hosts in the local-area-network
  – … when you don’t know how to identify the right one

• **Caching**: remember the past for a while
  – Store the information you learn to reduce overhead
  – Remember your own address & other host’s addresses

• **Soft state**: eventually forget the past
  – Associate a time-to-live field with the information
  – … and either refresh or discard the information
  – Key for robustness in the face of unpredictable change
MAC Address vs. IP Address

- **MAC addresses**
  - **Hard-coded** in read-only memory when adaptor is built
  - Like a social security number
  - Flat name space of 48 bits (e.g., 00-0E-9B-6E-49-76)
  - Portable, and can stay the same as the host moves
  - Used to get packet between interfaces on same network

- **IP addresses**
  - **Configured**, or learned dynamically
  - Like a postal mailing address
  - Hierarchical name space of 32 bits (e.g., 12.178.66.9)
  - Not portable, and depends on where the host is attached
  - Used to get a packet to destination IP subnet

Bootstrapping Problem

- Host doesn’t have an IP address yet
  - So, host doesn’t know what source address to use

- Host doesn’t know who to ask for an IP address
  - So, host doesn’t know what destination address to use

- Solution: shout to “**discover**” server that can help
  - **Broadcast** a server-discovery message (**ff:ff:ff:ff:ff**)
  - Server(s) sends a reply offering an address
Response from the DHCP Server

- DHCP “offer” message from the server
  - Configuration parameters (proposed IP address, mask, gateway router, DNS server, ...)
  - Lease time (duration the information remains valid)

- Multiple servers may respond
  - Multiple servers on the same broadcast network
  - Each may respond with an offer

- Accepting one of the offers
  - Client sends a DHCP “request” echoing the parameters
  - The DHCP server responds with an “ACK” to confirm
  - … and the other servers see they were not chosen

Dynamic Host Configuration Protocol
Deciding What IP Address to Offer

- Server as centralized configuration database
  - Parameters might be *statically* configured in the server
  - E.g., a dedicated IP address for each MAC address
  - Avoids complexity of configuring hosts directly
  - … while still having a permanent IP address per host

- Or, *dynamic* assignment of IP addresses
  - Server maintains a *pool* of available addresses
  - … and assigns them to hosts on demand
  - Leads to less configuration complexity
  - … and more efficient use of the pool of addresses
  - But: harder to track the same host over time

Soft State: Refresh or Forget

- Why is a lease time necessary?
  - Client can release the IP address (DHCP RELEASE)
    - E.g., "ipconfig /release" at the DOS prompt
    - E.g., clean shutdown of the computer
  - But, host *might not* release the address
    - E.g., the host crashes (blue screen of death!)
    - E.g., buggy client software
  - And you don’t want the address to be allocated forever

- Performance trade-offs
  - Short lease time: returns inactive addresses quickly
  - Long lease time: avoids overhead of frequent renewals & lessens frequency of lease being denied
So, Now the Host Knows Things

- IP address
- Mask
- Gateway router
- DNS server

- And can send packets to other IP addresses
- But: how to use the local network to do this?

Figuring Out Where To Send Locally

- Two cases:
  - Destination is on the local network
    - So need to address it directly
  - Destination is not local ("remote")
    - Need to figure out the first "hop" on the local network

- Determining if it’s local: use the netmask
  - E.g., mask destination IP address w/ 255.255.254.0
  - Same value as when we mask our own address?
    - Yes = local
    - No = remote
Where To Send Locally, con’t

• If it’s remote, look up first hop in (small) local routing table
  – E.g., by default, route via 1.2.3.19
  – Now do to the local case but for 1.2.3.19 rather than ultimate destination IP address

• For the local case, need to determine the destination’s MAC address

Sending Packets Over a Link

• Adaptors only understand MAC addresses
  – Translate the destination IP address to MAC address
  – Encapsulate the IP packet inside a link-level frame
5 Minute Break

Questions Before We Proceed?

Address Resolution Protocol

• Every node maintains an ARP table
  – <IP address, MAC address> pair

• Consult the table when sending a packet
  – Map destination IP address to destination MAC address
  – Encapsulate and transmit the data packet

• But: what if IP address not in the table?
  – Sender broadcasts: “Who has IP address 1.2.3.156?”
  – Receiver responds: “MAC address 58-23-D7-FA-20-B0”
  – Sender caches result in its ARP table

• Link-layer protocol (RFC826), not IP or UDP or TCP
  – Why?
Example: A Sending a Packet to B

How does host A send an IP packet to host B?

1. A sends packet to R.
2. R sends packet to B.

Host A Decides to Send Through R

- Host A constructs an IP packet to send to B
  - Source 111.111.111.111, destination 222.222.222.222
- Host A has a gateway router R
  - Used to reach destinations outside of 111.111.111.0/24
  - Address 111.111.111.110 for R learned via DHCP
Host A Sends Packet Through R

- Host A learns the MAC address of R's interface
  - ARP request: broadcast request for 111.111.111.110
  - ARP response: R responds with E6-E9-00-17-BB-4B
- Host A encapsulates the packet and sends to R

R Decides how to Forward Packet

- Router R’s adaptor receives the packet
  - R extracts the IP packet from the Ethernet frame
  - R sees the IP packet is destined to 222.222.222.222
- Router R consults its forwarding table
  - Packet matches 222.222.222.0/24 via other adaptor
R Sends Packet to B

- Router R’s learns the MAC address of host B
  - ARP request: broadcast request for 222.222.222.222
  - ARP response: B responds with 49-BD-D2-C7-56-2A

- Router R encapsulates the packet and sends to B

Security Analysis of ARP

- **Impersonation**
  - Any node that hears request can answer …
  - … and can say whatever they want

- Actual legit receiver never sees a problem
  - Because even though later packets carry its IP address, its NIC doesn’t capture them since not its MAC address

- Or: **Man-in-the-middle** attack
  - Imposter forwards everything it receives for destination but gets to inspect (& maybe alter) it first

- Does the attacker have to “win” a race?
  - Maybe not, if sender blindly believes ARP responses

- (Note: different attack than in HW #3 problem)
Network Control Messages

Error/Status Reporting

• Examples of errors a router may see
  – Router doesn’t know where to forward a packet
  – Packet’s time-to-live field expires
  – Packet is too big for link-layer router needs to use

• Router doesn’t really need to respond
  – Best effort means never having to say you’re sorry
  – So, IP could conceivably just silently drop packets

• But: silent failures are **really hard to diagnose**
  – IP includes basic feedback about network problems
  – Internet Control Message Protocol (ICMP / RFC 792)
Internet Control Message Protocol

- **ICMP runs on top of IP**
  - Same level TCP and UDP
  - Though still viewed as an integral part of IP
    - *Not* viewed as a transport protocol

- **Diagnostics**
  - Triggered when an IP packet encounters a problem
    - E.g., *time exceeded* or *destination unreachable*
  - ICMP packet sent back to the source IP address
    - Includes the error information (e.g., type and code)
    - … and IP header plus 8+ byte excerpt from original packet
  - Source host receives the ICMP packet
    - Inspects excerpt (e.g., protocol and ports)
    - … to identify which socket should receive the error

Types of Control Messages

- **Need Fragmentation**
  - IP packet too large for link layer, DF set

- **TTL Expired**
  - Decremented at each hop; generated if ⇒ 0

- **Unreachable**
  - Subtypes: network / host / port
    - (who generates Port Unreachable?)

- **Source Quench**
  - Old-style signal asking sender to slow down

- **Redirect**
  - Tells source to use a different local router
Path MTU Discovery

• **MTU** = Maximum Transmission Unit
  – Largest IP packet that a link supports

• **Path MTU** (PMTU) = minimum end-to-end MTU
  – Sender must keep datagrams no larger to avoid fragmentation

• How does the sender know the PMTU is?

• Strategy (RFC 1191):
  – Try a desired value
  – Set **DF** to prevent fragmentation
  – Upon receiving **Needs Fragmentation** ICMP …
    - … oops, that didn’t work, try a smaller value

Issues with Path MTU Discovery

• What set of values should the sender try?
  – Usual strategy: work through “likely suspects”
  – E.g., 4352 (FDDI), 1500 (Ethernet),
  1480 (IP-in-IP over Ethernet), 296 (some modems)

• What if the PMTU changes? (how could it?)
  – Sender will immediately see reductions (how?)
  – Sender can periodically try larger values

• What if **Needs Fragmentation** ICMP is lost?
  – Retransmission will elicit another one

• How can **The Whole Thing Fail**?
  – “PMTU Black Holes”: routers that don’t send the ICMP
Discovering Routing via *Time Exceeded*

- Host sends an IP packet
  - Each router decrements the time-to-live field
- If **TTL** reaches 0
  - Router sends **Time Exceeded** ICMP back to the source
  - Message identifies router sending it
    - Since ICMP is sent using IP, so it’s just the IP source address

Traceroute: Exploiting *Time Exceeded*

- Time-To-Live field in IP packet header
  - Source sends a packet with TTL ranging from 1 to *n*
  - Each router along the path decrements the TTL
  - “TTL exceeded” sent when TTL reaches 0
- Traceroute tool exploits this TTL behavior

Send packets with TTL=1, 2, … and record source of *Time Exceeded* message
## Example Traceroute: Berkeley to CNN

### Hop number, IP address, DNS name

<table>
<thead>
<tr>
<th>Hop</th>
<th>IP Address</th>
<th>DNS Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>169.229.62.1</td>
<td>inr-daedalus-0.CS.Berkeley.EDU</td>
</tr>
<tr>
<td>2</td>
<td>169.229.59.225</td>
<td>soda-cr-1-1-soda-br-6-2</td>
</tr>
<tr>
<td>3</td>
<td>128.32.255.169</td>
<td>vlan242.inr-202-doecev.Berkeley.EDU</td>
</tr>
<tr>
<td>4</td>
<td>128.32.0.249</td>
<td>gigE6-0-0.inr-666-doecev.Berkeley.EDU</td>
</tr>
<tr>
<td>5</td>
<td>128.32.0.66</td>
<td>qsv-juniper--ucb-gw.calren2.net</td>
</tr>
<tr>
<td>6</td>
<td>209.247.159.109</td>
<td>POS1-0.hsipaccess1.SanJose1.Level3.net</td>
</tr>
<tr>
<td>7</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>64.159.1.46</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>209.247.9.170</td>
<td>pos8-0.hsa2.Atlanta2.Level3.net</td>
</tr>
<tr>
<td>10</td>
<td>66.185.138.33</td>
<td>pop2-atm-P0-2.atdn.net</td>
</tr>
<tr>
<td>11</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>66.185.136.17</td>
<td>pop1-atl-P4-0.atdn.net</td>
</tr>
<tr>
<td>13</td>
<td>64.236.16.52</td>
<td>www4.cnn.com</td>
</tr>
</tbody>
</table>

### traceroute to www.whitehouse.gov (204.102.114.49), 30 hops max, 40 byte packets

<table>
<thead>
<tr>
<th>Hop</th>
<th>IP Address</th>
<th>DNS Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>cory115-1-gw.EECS.Berkeley.EDU (128.32.48.1)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.829 ms</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>cory-cr-1-1-soda-cr-1-2.EECS.Berkeley.EDU (169.229.59.233)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.953 ms</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>soda-cr-1-1-soda-br-6-2.EECS.Berkeley.EDU (169.229.59.225)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.461 ms</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>g3-8.inr-202-reccev.Berkeley.EDU (128.32.255.169)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.402 ms</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>ge-1-3-0.inr-002-reccev.Berkeley.EDU (128.32.0.38)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.428 ms</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>oak-dc2--ucb-ge.cenic.net (137.164.23.29)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.731 ms</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>dc-oak-dc1--oak-dc2-p2p-2.cenic.net (137.164.22.194)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3.045 ms</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>dc-lax-dc1--sac-dc1-pos.cenic.net (137.164.22.126)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>13.104 ms</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>137.164.22.21</td>
<td>(137.164.22.21)</td>
</tr>
<tr>
<td></td>
<td>13.328 ms</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>dc-tus-dc1--lax-dc2-pos.cenic.net (137.164.22.43)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>18.775 ms</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>a204-102-114-49.deploy.akamaitech.com (204.102.114.49)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>18.137 ms</td>
<td></td>
</tr>
</tbody>
</table>
Ping: Echo and Reply

- ICMP includes simple “echo” functionality
  - Sending node sends an ICMP Echo Request message
  - Receiving node sends an ICMP Echo Reply

- Ping tool
  - Tests connectivity with a remote host
  - ... by sending regularly spaced Echo Request
  - ... and measuring delay until receiving replies

- ICMP includes other forms of probing
  - See /usr/include/netinet/ip_icmp.h on a Unix system
  - However, very often disabled ...

- Probing hosts
  - Try (say) traceroute www.cs.duke.edu
  - and ping www.cs.duke.edu

Security Implications of ICMP?

- Attacker can cause host to accept an ICMP if the excerpt looks correct (assuming the host checks)
  - Must guess recent IP packet header & 8B of payload
  - All that really matters is source/destination addresses and ports

- Threat:
  - Denial-of-Service (DoS)
    - Unreachable, Redirect
  - Impaired performance
    - Need Fragmentation, Source Quench
Summary

• Important control functions
  – Bootstrapping
  – Error/status reporting and monitoring

• Internet control protocols
  – Dynamic Host Configuration Protocol (DHCP)
  – Address Resolution Protocol (ARP)
  – Internet Control Message Protocol (ICMP)

• Reminder: I’m away next week
  – Dilip lecturing on Monday
  – Scott Shenker on Wednesday