Questions to be answered today

- What must a host know before it can operate?
  - Local information
  - Remote information

- How do you avoid manual configuration?
  - Management: most important issue in networking today!

- How can host learn about local network?

- How can host learn about the rest of the Internet?

Answers Involve….

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

- Network control messages (global)
  - Internet Control Message Protocol (ICMP)
  - Exploiting ICMP for discovering Internet path properties

Bootstrap and Control Protocols

- Very different mechanisms

- For very different environments

Internet versus LAN

- Scale:
  - Huge vs Limited

- Management:
  - Ad Hoc vs Managed

- Delivery Model:
  - No broadcast vs broadcast

As a result…..

- Local mechanisms: broadcast to find things
  - “Bootstrapping”

- Remote mechanisms: investigate path
  - How to use what routing has already found
  - “Network Control Messages”
Preliminary Observations

Broadcast at Link-Level
- Use broadcast address: ff:ff:ff:ff:ff:ff
- If have return MAC address, use that in response
- Unless want everyone to know result

Broadcast at IP Level
- Can’t broadcast to all IP hosts
- But application might want to send “local” broadcast
- Uses IP broadcast address 225.225.225.225
- Link-layer then users link-layer broadcast

Reaching a Host
- First look up IP address
- Need to know where local DNS server is
  – How does a host know this?

Sending a Packet
- On same subnet:
  – Use MAC address of destination
  – How do a host know that?
- On some other subnet:
  – Use MAC address of first-hop router
  – How do a host know that?

Bootstrapping a Host
What Does a Host Need to Know?

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

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: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

Arriving client

DHCP discover (broadcast)

DHCP offer (broadcast)

DHCP request (broadcast)

DHCP ACK (broadcast)

Why all the broadcasts?

DHCP server

203.1.2.5

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 with 255.255.254.0
  - Is it the 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 (very small) local routing table
  - E.g., by default, route via 1.2.3.19
  - Now do 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
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-FA-20-B0"
  - Sender caches result in its ARP table
- Link-layer protocol (RFC 826)
  - Because necessary to bootstrap IP connectivity

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 updates frames w/ correct MAC address and forwards whatever it receives to the legit 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

Network Control Messages (and how to use them for discovery)

Error/Status Reporting

• Examples of errors a router may see
  – Router doesn’t know where to forward a packet
  – Packet’s time-to-live (hop count) 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 as TCP and UDP
  – Though viewed as an integral part of IP (not transport)
• Diagnostics
  – Triggered when an IP packet encounters a problem
    o E.g., Time Exceeded or Destination Unreachable
  – ICMP packet sent back to the source IP address
    o Includes the error information (e.g., type and code)
    o … and IP header plus 8+ byte excerpt from original packet
  – Source host receives the ICMP packet
    o Inspects excerpt (e.g., protocol and ports) to identify socket
  – Exception: ICMP not sent if problem packet is ICMP
    o And just for fragment 0 of a group of fragments

Types of Control Messages

• Need Fragmentation
  – IP packet too large for link layer, DF set
• TTL Expired
  – Decremented at each hop; generated if $\Rightarrow 0$
• Unreachable
  – Subtypes: network / host / port
    o (who generates Port Unreachable?)
• Source Quench
  – Old-style signal asking sender to slow down
• Redirect
  – Tells source to use a different local router
Discovering Network Path Properties

• ICMP provides way for routers to talk to end hosts
• Can be used in clever ways to probe the network to discover things about its internals:
  – PMTU Discovery:
    o What is largest packet that can be sent completely through the network w/o needing fragmentation?
    • Most efficient size to use
    • (Plus fragmentation can amplify loss)
  – Traceroute:
    o What is the series of routers that a packet traverses as it travels through the network?

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 Need Fragmentation ICMP ...
    o ... 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 in PMTU (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
    o Since ICMP is sent using IP, 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

Traceroute to www.whitehouse.gov (204.102.114.49), 30 hops max, 40 byte packets
traceroute to www.whitehouse.gov (204.102.114.49),
30 hops max, 40 byte packets
1 cory115-1-gw.EECS.Berkeley.EDU (128.32.48.1)
  0.829 ms 0.660 ms 0.565 ms
2 cory-cr-1-1-soda-cr-1-2.EECS.Berkeley.EDU (169.229.59.233)
  0.953 ms 0.857 ms 0.727 ms
3 soda-cr-1-1-soda-br-6-2.EECS.Berkeley.EDU (169.229.59.225)
  1.461 ms 1.260 ms 1.137 ms
4 g3-8.inr-202-recev.Berkeley.EDU (128.32.255.169)
  1.402 ms 1.298 ms *
5 ge-1-3-0.inr-002-recev.Berkeley.EDU (128.32.0.38)
  1.428 ms 1.889 ms 1.378 ms
6 oak-dc2--ucb-ge.cenic.net (137.164.23.29)
  1.731 ms 1.643 ms 1.680 ms
7 dc-oak-dc1--oak-dc2-p2p-2.cenic.net (137.164.22.194)
  2.045 ms 1.640 ms 1.630 ms
8 * * *
9 dc-lax-dc1--sac-dc1-pos.cenic.net (137.164.22.21)
  13.104 ms 13.163 ms 12.988 ms
10 137.164.22.21 (137.164.22.21)
  13.328 ms 42.981 ms 13.548 ms
11 dc-tus-dc1--lax-dc2-pos.cenic.net (137.164.22.43)
  18.775 ms 17.469 ms 21.652 ms
12 a204-102-114-49.deploy.akamaitechnologies.com (204.102.114.49)
  18.137 ms 14.905 ms 19.730 ms

Lost Reply
Router doesn't send ICMPs
No PTR record for address
Final Hop

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)
    o Unreachable, Redirect
  – Impaired performance
    o 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)
• Next lecture: Shortest-Path Routing
  – K&R 4.5, 4.6.1, 4.6.2