Lecture 8 (Feb 12, 2004)

Outline

- ICMP
- RARP
- DHCP
- NAT

Internet Control Message Protocol (ICMP)

- Echo (ping)
- Redirect (from router to source host)
- Destination unreachable (protocol, port, or host)
- TTL exceeded (so datagrams don't cycle forever)
- Checksum failed
- Reassembly failed
- Cannot fragment

ICMP

- Uses IP but is a separate protocol in the network layer

ICMP HEADER

IP HEADER	**IP HEADER** **PROTOCOL = 1**
IP DATA	**TYPE CODE CHECKSUM** **REMAINDER OF ICMP MESSAGE (FORMAT IS TYPE SPECIFIC)**

Echo and Echo Reply

TYPE CODE CHECKSUM
IDENTIFIER SEQUENCE #
DATA ….

TYPE: 8 = ECHO, 0 = ECHO REPLY CODE; CODE = 0

IDENTIFIER
An identifier to aid in matching echoes and replies
SEQUENCE #
Same use as for IDENTIFIER
UNIX "ping" uses echo/echo reply

Ping Example

```
C:\WINDOWS\Desktop>ping www.soi.wide.ad.jp

 Pinging asari.soi.wide.ad.jp [203.178.137.88] with 32 bytes of data:
 Reply from 203.178.137.88: bytes=32 time=253ms TTL=240
 Reply from 203.178.137.88: bytes=32 time=231ms TTL=240
 Reply from 203.178.137.88: bytes=32 time=225ms TTL=240
 Reply from 203.178.137.88: bytes=32 time=214ms TTL=240

Ping statistics for 203.178.137.88:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 214ms, Maximum =  253ms, Average =  230ms
```


Redirect when no route to Destination

TYPE CODE CHECKSUM
NEW ROUTER ADDRESS
IP HEADER + 64 bits data
from original DG

TYPE = 5
CODE =
0 = Network redirect
1 = Host redirect
2 = Network redirect for specific TOS
3 = Host redirect for specific TOS

Destination Unreachable

TYPE CODE CHECKSUM
UNUSED
IP HEADER + 64 bits data from original DG

TYPE = 3

CODE 0 = Net unreachable
1 = Host unreachable
2= Protocol unreachable
3 = Port unreachable
4 = Fragmentation needed but DF set
5 = Source route failed

Source Quench

TYPE CODE CHECKSUM
UNUSED
IP HEADER + 64 bits data from original DG

TYPE = 4; CODE = 0

Indicates that a router has dropped the original DG or may indicate that a router is approaching its capacity limit.

Correct behavior for source host is not defined.

Traceroute

- UNIX utility - displays router used to get to a specified Internet Host
- Operation
  - router sends ICMP Time Exceeded message to source if TTL is decremented to 0
  - if TTL starts at 5, source host will receive Time Exceeded message from router that is 5 hopes away
- Traceroute sends a series of probes with different TTL values… and records the source address of the ICMP Time Exceeded message for each
- Probes are formatted to that the destination host will send an ICMP Port Unreachable message

TraceRoute Example

```
C:\windows\desktop> tracert www.soi.wide.ad.jp
Tracing route to asari.soi.wide.ad.jp [203.178.137.88]
over a maximum of 30 hops:
  1    19 ms    27 ms    23 ms  208.166.201.1
  2    17 ms    13 ms    14 ms  204.189.71.9
  3    25 ms    29 ms    29 ms  aar1-serial4-1-0-0.Minneapolismpn.cw.net [208.174.7.5]
  4    24 ms    27 ms    24 ms  acr1.Minneapolismpn.cw.net [208.174.2.61]
  5    26 ms    22 ms    23 ms  acr2-loopback.Chicagochd.cw.net [208.172.2.62]
  6    29 ms    29 ms    27 ms  cand-w-private-peering.Chicagochd.cw.net [208.172.1.222]
  7    28 ms    24 ms    28 ms  0.so-5-2-0.XL2.CHI2.ALTER.NET [152.63.68.6]
  8    26 ms    27 ms    28 ms  0.so-7-0-0.XR2.CHI2.ALTER.NET [152.63.67.134]
  9    25 ms    24 ms    26 ms  292.at-2-0-0.TR2.CHI4.ALTER.NET [152.63.64.234]
 10    73 ms    74 ms    73 ms  106.ATM7-0.TR2.LAX2.ALTER.NET [146.188.136.142]
 11    74 ms    76 ms    76 ms  198.ATM7-0.XR2.LAX4.ALTER.NET [146.188.249.5]
 12    73 ms    75 ms    77 ms  192.ATM5-0.GW9.LAX4.ALTER.NET [152.63.115.77]
 13    80 ms    73 ms    76 ms  kdd-gw.customer.ALTER.NET [157.130.226.14]
 14    84 ms    84 ms    91 ms  202.239.170.236
 15    97 ms    81 ms    86 ms  cisco1-eth-2-0.LosAngeles.wide.ad.jp [209.137.144.98]
 16   174 ms   174 ms   178 ms  cisco5.otemachi.wide.ad.jp [203.178.136.238]
 17   201 ms   196 ms   194 ms  cisco2.otemachi.wide.ad.jp [203.178.137.34]
 18   183 ms   182 ms   196 ms  foundry2.otemachi.wide.ad.jp [203.178.140.216]
 19   183 ms   185 ms   178 ms  gsr1.fujisawa.wide.ad.jp [203.178.138.252]
 20   213 ms   205 ms   201 ms  asari.soi.wide.ad.jp [203.178.137.88]
Trace complete.
```


Determining an IP Address at Startup

- How does a machine without permanent storage determine its IP address?
  - OS images with specific IP's cannot be used on multiple machines
  - Critical for network appliances or embedded systems
- Use the network to obtain an IP from a remote server
  - System must use its physical address to to communicate
  - Requests address from server which maintains table of IP's
  - System doesn't know the server - sends broadcast request for address

Reverse Address Resolution Protocol

- RARP is part of the TCP/IP specification
- RARP operates much like ARP
  - A requestor broadcasts is RARP request
  - Servers respond by sending response directly to requestor
  - Requestor keeps IP delivered by first responder
  - Requestor keeps sending requests until it gets an IP
- Clearly there is a need for redundant RARP servers for reliability
  - Timeouts can be used to activate backup RARP servers
    - Backup servers reply to a RARP request if they don't hear the RARP response from the primary server after some time

Alternatives to RARP

- RARP has shortcomings
  - Most are subtle and all deal with fact that RARP operates at physical level
- BOOTstrap Protocol (BOOTP) was developed as an alternative to RARP – moves process to network level
  - Uses UDP/IP packets to carry messages
    - Hosts are still identified by MAC address
  - How can UDP running over IP be used by a computer to discover its IP address?
    - Uses special case IP address 255.255.255.255 – limited broadcast – not forwarded by routers
    - Forces IP to broadcast on LAN before host IP is known
    - BOOTP server responds using limited broadcast
    - Request transmission via random timeout to avoid synchronization

Dynamic Configuration

- BOOTP was designed for relatively static environment where each host has a permanent network connection
  - Net manager creates a BOOTP config file with parameters for each host – file is typically stable for long periods
- Wireless networking enables environments much more dynamic
  - BOOTP does not provide for dynamic address assignment
- Dynamic configuration is the primary method for IP address allocation used today
  - Not only facilitates mobility but also efficient use of IPs

Dynamic Host Configuration Protocol

- DHCP extends BOOTP
  - Still supports static allocation
  - Supports automatic configuration where addresses are permanent but assigned by DHCP
  - Supports temporary allocation
- Relies on existence of a DHCP server
  - Repository for host configuration information
  - Maintains a pool of available IP's for use on demand
  - Considerably reduces administration overhead
    - Autoconfiguration of course depends on administrative policy
  - Uses UDP to send messages
    - Uses a *relay agent* to communicate with servers off LAN (same as BOOTP)
      - Relay agent is statically configured with DHCP server address

DHCP Implementation

- State machine (6 states) which determines DHCP operation
  - Host boots into *INITIALIZE* state
- To contact the DHCP server(s) a client sends DHCPDISCOVER message to IP broadcast address and moves to *SELECT* state
  - Unique header format with variable length options field
  - UDP packet sent to well known BOOTP port 67
- Server(s) respond with DHCPOFFER message
  - Client can receive 0 or more responses and responds to one
- Client moves to *REQUEST* state to negotiate IP lease with 1 server
  - Sends DHCPREQUEST message to server which responds with DHCPACK
- Client is then in *BOUND* (normal) state

DHCP Implementation contd.

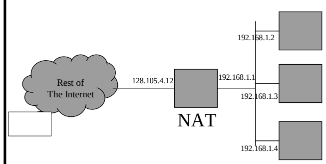
- From *BOUND*, client can issue DHCPRELEASE and return to *INITIALIZE* state
  - This is simply client deciding it no longer needs the IP
- When lease reaches 50% of lease expiration time, it issues DHCPREQUEST to extend lease of current IP with server and moves to *RENEW* state
  - Receipt of DHCPACK moves client back to *BOUND* state
  - Receipt of DHCPNACK moves client back to *INITIALIZE* state
- If no response is received by 87.5% of lease expiration time, the client resends the DHCPREQUEST and moves to *REBIND* state
  - Receipt of DHCPACK moves client back to *BOUND* state
  - Receipt of DHCPNACK or timeout moves client back to *INITIALIZE* state

DHCP Details

- Without relay agent, DHCP would not scale since it would require large number of servers (one per LAN)
- Addresses which are leased over a given period of time and must be updated
  - This means that DHCP requests might have to be made multiple times by the same system (RENEW requests)

Network Address Translation

- Maps an internal <address, port> to an external <address, port>
- Source address, port of outgoing packet changed
- Destination address, port of incoming packet changed



Maintains a table
to translate
<IP addr, port> pairs

Internal	External
192.168.1.4, 1336	128.105.4.12, 1330
192.168.1.3, 1455	128.105.4.12, 1331
192.168.1.2, 1336	128.105.4.12, 1332
192.168.1.2, 1771	128.105.4.12, 1333

•Used to temporarily tide over IP address space depletion
•Also avoids re-numbering of IP addresses when customer changes provider
•But breaks end-to-end properties