Recap from last class

Network layer provides addressing, internetworking, and routing and forwarding

IP, the network layer for the Internet, provides a datagram abstraction
IPv4 addresses are written as 128.23.45.32, allocated as prefixes

DHCP: Hosts get their addresses and other essential information
ARP: Convert IP addresses to link layer addresses
Fragmentation
Fragmentation

• Problem: How do we connect networks with different maximum packet sizes?
  • Need to split up packets, or discover the largest size to use
Packet Size Problem

• Different networks have different max packet sizes
  • Or MTU (Maximum Transmission Unit)
  • E.g., Ethernet 1.5K, WiFi 2.3K

• Prefer large packets for efficiency
  • But what size is too large?
  • Difficult as node doesn’t know complete network path
Packet Size Solutions

• Fragmentation (now)
  • Split up large packets if they are too big to send
  • Classic method, dated

• Discovery (next)
  • Find the largest packet that fits on the network path
  • IP uses today instead of fragmentation
IPv4 Fragmentation

- Routers fragment packets too large to forward
- Receiving host reassembles to reduce load on routers

Fits on first link
IPv4 Fragmentation Fields

• Header fields used to handle packet size differences
  • Identification, Fragment offset, MF/DF control bits
IPv4 Fragmentation Procedure

• Routers split a packet that is too large:
  • Typically break into large pieces
  • Copy IP header to pieces
  • Adjust length on pieces
  • Set offset to indicate position
  • Set MF (More Fragments) on all pieces except last

• Receiving hosts reassembles the pieces:
  • Identification field links pieces together, MF tells receiver when complete
IPv4 Fragmentation (2)

Before
MTU = 2300
ID = 0x12ef
Data Len = 2300
Offset = 0
MF = 0

After
MTU = 1500
ID =
Data Len =
Offset =
MF =

ID =
Data Len =
Offset =
MF =

(Ignore length of headers)
IPv4 Fragmentation (3)

Before
MTU = 2300
ID = 0x12ef
Data Len = 2300
Offset = 0
MF = 0

After
MTU = 1500
ID = 0x12ef
Data Len = 1500
Offset = 0
MF = 1

ID = 0x12ef
Data Len = 800
Offset = 1500
MF = 0
IPv4 Fragmentation (4)

• It works!
  • Allows repeated fragmentation

• But fragmentation is undesirable
  • More work for routers, hosts
  • Tends to magnify loss rate
  • Security vulnerabilities too
Path MTU Discovery

• Discover the MTU that will fit
  • So we can avoid fragmentation
  • The method in use today

• Host tests path with large packet
  • Routers provide feedback if too large; they tell host what size would have fit
Path MTU Discovery (2)
Path MTU Discovery (3)
Path MTU Discovery (4)

• Process may seem involved
  • But usually quick to find right size
  • MTUs smaller on edges of network
• Path MTU depends on the path and can change
  • Search is ongoing
• Implemented with ICMP (next)
  • Set DF (Don’t Fragment) bit in IP header to get feedback
Internet Control Message Protocol (ICMP)
• Problem: What happens when something goes wrong during forwarding?
  • Need to be able to find the problem
Internet Control Message Protocol

- ICMP is a companion protocol to IP
  - They are implemented together
  - Sits on top of IP (IP Protocol=1)

- Provides error report and testing
  - Error is at router while forwarding
  - Also testing that hosts can use
ICMP Errors

• When router encounters an error while forwarding:
  • It sends an ICMP error report back to the IP source
  • It discards the problematic packet; host needs to rectify

Oh, now I see ...
ICMP Message Format (2)

• Each ICMP message has a Type, Code, and Checksum
• Often carry the start of the offending packet as payload
• Each message is carried in an IP packet

Portion of offending packet, starting with its IP header

<table>
<thead>
<tr>
<th>Src=router, Dst=A</th>
<th>Type=X, Code=Y</th>
<th>Src=A, Dst=B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Protocol = 1</td>
<td>ICMP header</td>
<td>XXXXXXXXXXXX</td>
</tr>
<tr>
<td>IP header</td>
<td></td>
<td>ICMP data</td>
</tr>
</tbody>
</table>
Example ICMP Messages

<table>
<thead>
<tr>
<th>Name</th>
<th>Type / Code</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dest. Unreachable (Net or Host)</td>
<td>3 / 0 or 1</td>
<td>Lack of connectivity</td>
</tr>
<tr>
<td>Dest. Unreachable (Fragment)</td>
<td>3 / 4</td>
<td>Path MTU Discovery</td>
</tr>
<tr>
<td>Time Exceeded (Transit)</td>
<td>11 / 0</td>
<td>Traceroute</td>
</tr>
<tr>
<td>Echo Request or Reply</td>
<td>8 or 0 / 0</td>
<td>Ping</td>
</tr>
</tbody>
</table>

Testing, not a forwarding error: Host sends Echo Request, and destination responds with an Echo Reply
Traceroute

- IP header contains TTL (Time to live) field
  - Decremented every router hop, with ICMP error at zero
  - Protects against forwarding loops
Traceroute (2)

- Traceroute repurposes TTL and ICMP functionality
  - Sends probe packets increasing TTL starting from 1
  - ICMP errors identify routers on the path
Network Address Translation (NAT)
Problem: Internet Growth

• Today, Internet connects
  • 4B people
  • ~50B devices

• And we’re using 32-bit addresses!
  • ~2B unique addresses
The End of New IPv4 Addresses

• Now running on leftover blocks held by the regional registries; much tighter allocation policies

Exhausted on 2/11!
Exhausted on 4/11 and 9/12!

IANA (All IPs) → ARIN (US, Canada) → ISPs
IANA (All IPs) → APNIC (Asia Pacific) → Companies
IANA (All IPs) → RIPE (Europe)
IANA (All IPs) → LACNIC (Latin Amer.)
IANA (All IPs) → AfriNIC (Africa)

End of the world? 12/21/12?
A market for IPv4 addresses

https://ipv4marketgroup.com/ipv4-price-trends/
Solution 1: Network Address Translation (NAT)

• Basic idea: Map many “Private” IP addresses to one “Public” IP.

• Allocate IPs for private use (192.168.x, 10.x)

I’m a NAT box too!
Layering Review

• Remember how layering is meant to work?
  • “Routers don’t look beyond the IP header.” Well …
Middleboxes

• Sit “inside the network” but perform “more than IP” processing on packets to add new functionality
  • NAT box, Firewall / Intrusion Detection System
Middleboxes (2)

• Advantages
  • A possible rapid deployment path when no other option
  • Control over many hosts (IT)

• Disadvantages
  • Breaking layering interferes with connectivity
    • strange side effects
  • Poor vantage point for many tasks
NAT (Network Address Translation) Box

• NAT box maps an internal IP to an external IP
  • Many internal hosts connected using few external addresses
  • Middlebox that “translates addresses”

• Motivated by IP address scarcity
  • Controversial at first, now accepted
NAT (2)

- Common scenario:
  - Home computers use “private” IP addresses
  - NAT (in AP/firewall) connects home to ISP using a single external IP address

![Diagram showing unmodified computers at home, NAT box, and ISP]

Unmodified computers at home $\downarrow$

Looks like one computer outside

ISP

NAT box
How NAT Works

• Keeps an internal/external translation table
  • Typically uses IP address + TCP port
  • This is address and port translation

<table>
<thead>
<tr>
<th>What host thinks</th>
<th>What ISP thinks</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Internal IP:port</strong></td>
<td><strong>External IP : port</strong></td>
</tr>
<tr>
<td>192.168.1.12 : 5523</td>
<td>44.25.80.3 : 1500</td>
</tr>
<tr>
<td>192.168.1.13 : 1234</td>
<td>44.25.80.3 : 1501</td>
</tr>
<tr>
<td>192.168.2.20 : 1234</td>
<td>44.25.80.3 : 1502</td>
</tr>
</tbody>
</table>

• Need ports to make mapping 1-1 since there are fewer external IPs
How NAT Works (2)

- Internal → External:
  - Look up and rewrite Source IP/port

<table>
<thead>
<tr>
<th>Internal IP:port</th>
<th>External IP : port</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.168.1.12 : 5523</td>
<td>44.25.80.3 : 1500</td>
</tr>
</tbody>
</table>

External destination IP=X, port=Y

Src = Dst = 192.168.1.12 : 5523
Src = Dst = 44.25.80.3 : 1500
How NAT Works (3)

• External → Internal
  • Look up and rewrite Destination IP/port

<table>
<thead>
<tr>
<th>Internal destination</th>
<th>Internal IP:port</th>
<th>External IP : port</th>
</tr>
</thead>
</table>
| 192.168.1.12 : 5523  | 44.25.80.3 : 1500

Src = Dst =

NAT box

Src = Dst =

External source
IP=X, port=Y

CSE 461 University of Washington
How NAT Works (4)

• Need to enter translations in the table for it to work
  • Create external name when host makes a TCP connection

<table>
<thead>
<tr>
<th>Internal source</th>
<th></th>
<th>External destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>Src = Internal IP:port</td>
<td></td>
<td>External IP:port</td>
</tr>
<tr>
<td>192.168.1.12:5523</td>
<td></td>
<td>IP=X, port=Y</td>
</tr>
<tr>
<td>Dst =</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
NAT in action

```
Ratuls-MacBook-Pro:19wi ratuls$ ifconfig en0
en0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    ether f0:18:98:a5:f9:cc
    inet6 fe80::e511:c06f:78f9:en0 prefixlen 64 secured scopeid 0xa
    inet 192.168.88.14 netmask 0xffffff00 broadcast 192.168.88.255
    nd6 options=201<PERFORMNUD,DAD>
    media: autoselect
    status: active
```
NAT Downsides

• Connectivity has been broken!
  • Can only send incoming packets after an outgoing connection is set up
  • Difficult to run servers or peer-to-peer apps (Skype)
• Doesn’t work if return traffic by passes the NAT
• Breaks apps that expose their IP addresses (FTP)
NAT Upsides

• Relieves much IP address pressure
  • Many home hosts behind NATs
• Easy to deploy
  • Rapidly, and by you alone
• Useful functionality
  • Firewall, helps with privacy
• Kinks will get worked out eventually
  • “NAT Traversal” for incoming traffic