Internet Control Message Protocol (ICMP)

CSE 461

Ratul Mahajan
Topic

- 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

| Src=router, Dst=A  | Type=X, Code=Y | Src=A, Dst=B
| Protocol = 1      |              | XXXXXXXXXXXXXXXXXXXX |
|                   | ICMP header  | ICMP data            |
| IP header         |              |                      |
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’s success

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)

APNIC (Asia Pacific)

RIPE (Europe)

LACNIC (Latin Amer.)

AfriNIC (Africa)

ISPs

Companies

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

https://auctions.ipv4.global/prior-sales
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)
Layering Review

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

Sit “in the network” but do “more than IP” processing on packets to add new functionality

• NATs, Firewalls, Intrusion Detection Systems
Aside: 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
• Common scenario:
  • Home computers use “private” IP addresses
  • NAT (in AP/firewall) connects home to ISP using a single external IP address
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>

*Internal source

NAT box

External destination

IP=X, port=Y

Src =

Dst =

Src =

Dst =

CSE 461 University of Washington
How NAT Works (3)

• **External → Internal**
  - Look up and rewrite Destination 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 source
IP=\(X\), port=\(Y\)

Internal destination

Src =

Dst =

NAT box

Src =

Dst =
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 IP:port</th>
<th>External IP : port</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.168.1.12 : 5523</td>
<td></td>
</tr>
</tbody>
</table>

Source

NAT box

Destination
NAT in action

```bash
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::440: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
```

whatismyipaddress

About 166,000 results (0.45 seconds)

66.171.178.94
Your public IP address

CSE 461 University of Washington
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
IPv6
IP Version 6 to the Rescue

• Effort started by the IETF in 1994
  • Much larger addresses (128 bits)
  • Many sundry improvements

• Became an IETF standard in 1998
  • Nothing much happened for a decade
  • Hampered by deployment issues, and a lack of adoption incentives
  • Big push ~2011 as exhaustion looms
IPv6

- Features large addresses
  - 128 bits, most of header
- New notation
  - 8 groups of 4 hex digits (16 bits)
  - Omit leading zeros, groups of zeros

Ex: 2001:0db8:0000:0000:0000:ff00:0042:8329
→ 2001:db8::ff00:42:8329
IPv6 (2)

- Lots of other changes
  - Only public addresses
    - No more NAT!
  - Streamlined header processing
    - No checksum (why’s that faster?)
  - Flow label to group of packets
  - IPSec by default
  - Better fit with “advanced” features (mobility, multicasting, security)
IPv6 Stateless Autoconfiguration (SLAAC)

- Replaces DHCP (sorta...)
- Uses ICMPv6
- Process:
  - Send broadcast message
  - Get prefix from router
  - Attach MAC to router Prefix
IPv6 Transition

• The Big Problem:
  • How to deploy IPv6?
  • Fundamentally incompatible with IPv4

• Dozens of approaches proposed
  • Dual stack (speak IPv4 and IPv6)
  • Translators (convert packets)
  • Tunnels (carry IPv6 over IPv4)
Tunneling

- Native IPv6 islands connected via IPv4
- Tunnel carries IPv6 packets across IPv4 network
Tunneling (2)

• Tunnel acts as a single link across IPv4 network
Tunneling (3)

- Tunnel acts as a single link across IPv4 network
- Difficulty is to set up tunnel endpoints and routing