The Internet/Network Layer

ICMP

• Many times we have said things like “drop the packet and send an error message back”
• So how are the messages sent?
• We only have packets, so the message must be in a packet
• A normal IP packet, with particular contents
• An Internet Control Message Protocol packet
The Internet/Network Layer

ICMP

- Used for general of the Internet, in particular errors
- Layered on top of IP, but considered to be part of the Internet layer
The Internet/Network Layer

ICMP

- Type: kind of message, e.g., “TTL expired”, “destination unreachable”
- Code: additional information, e.g., “destination unreachable” has “network unreachable” and “host unreachable” codes
The Internet/Network Layer

ICMP

- Checksum
- A field that has varying purposes for different types
The Internet/Network Layer

ICMP

- A general data field if needed
The Internet/Network Layer

ICMP

- ICMP packets are IP packets and so can be lost, delayed, duplicated or otherwise corrupted.
- ICMP errors can be generated for ICMP packets, with certain reservations.
- ICMP messages are classed as either a query or an error.
- E.g., ICMP “echo request” (ping) is a query, but “TTL expired” is an error.
The Internet/Network Layer

ICMP

- ICMP errors are not generated for
  - ICMP errors (e.g., TTL expires on an ICMP packet)
  - a packet whose destination is a broadcast/multicast
  - a packet whose source is a broadcast/multicast
  - a packet whose link-layer address is a broadcast
  - any fragment other than the first
The Internet/Network Layer

ICMP

• This is to prevent *broadcast storms*, where a single error is multiplied up into many ICMP packets

• Non-initial fragments don't contain enough identifying information for the OS to do anything useful with them, so don't bother with them (see later)
The Internet/Network Layer

ICMP

<table>
<thead>
<tr>
<th>Type</th>
<th>Err</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>ECHOREPLY</td>
<td></td>
<td>reply from a ping</td>
</tr>
<tr>
<td>DEST_UNREACH</td>
<td>e</td>
<td>network unreachable</td>
</tr>
<tr>
<td></td>
<td>e</td>
<td>host unreachable</td>
</tr>
<tr>
<td></td>
<td>e</td>
<td>port unreachable</td>
</tr>
<tr>
<td></td>
<td>e</td>
<td>fragmentation wanted but DF set</td>
</tr>
<tr>
<td>REDIRECT</td>
<td>e</td>
<td>routing redirect for a network</td>
</tr>
<tr>
<td></td>
<td>e</td>
<td>routing redirect for a host</td>
</tr>
<tr>
<td>ECHO</td>
<td></td>
<td>ping</td>
</tr>
<tr>
<td>TIME_EXCEEDED</td>
<td>e</td>
<td>TTL reached 0</td>
</tr>
<tr>
<td></td>
<td>e</td>
<td>fragment reassembly time exceeded</td>
</tr>
</tbody>
</table>
The Internet/Network Layer

ICMP Ping

- ICMP has many other uses than simply running IP
- Discover if a machine is up and running using ICMP ping
- This sends an ICMP “echo request” (usually called a “ping”) packet, waits a second, then repeats
The Internet/Network Layer

ICMP Ping

- ICMP type 0, code 0, with some random data
- A functioning host that gets a ping should return a “echo reply”
The Internet/Network Layer

ICMP Ping

- ICMP type 8, code 0, and a copy of the identifier, sequence and data
- The identifier field allows the originator to match up replies with requests
The Internet/Network Layer

ICMP Ping

<table>
<thead>
<tr>
<th>8 bit type</th>
<th>8 bit code</th>
<th>16 bit checksum</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0 or 8)</td>
<td>(0)</td>
<td></td>
</tr>
</tbody>
</table>

Identifier    Sequence number

Optional data

- The sequence starts at 0 and increases by 1 for each ping sent
- This allows us to spot lost, duplicated or reordered packets
The Internet/Network Layer

ICMP Ping

% ping www.yahoo.co.uk
PING homerc.europe.yahoo.com: 56 data bytes
64 bytes from rc3.europe.yahoo.com (194.237.109.72): icmp_seq=0. time=160. ms
64 bytes from rc3.europe.yahoo.com (194.237.109.72): icmp_seq=1. time=154. ms
64 bytes from rc3.europe.yahoo.com (194.237.109.72): icmp_seq=2. time=176. ms
64 bytes from rc3.europe.yahoo.com (194.237.109.72): icmp_seq=3. time=159. ms
64 bytes from rc3.europe.yahoo.com (194.237.109.72): icmp_seq=4. time=161. ms

^C

----homerc.europe.yahoo.com PING Statistics----
5 packets transmitted, 5 packets received, 0% packet loss
round-trip (ms)  min/avg/max = 154/162/176

• “ping” command also keeps track of *round trip time* (RTT), the time between sending a request and getting the corresponding reply
The Internet/Network Layer

ICMP Ping

% ping www.yahoo.co.uk
PING homerc.europe.yahoo.com: 56 data bytes
64 bytes from rc3.europe.yahoo.com (194.237.109.72): icmp_seq=0. time=160. ms
64 bytes from rc3.europe.yahoo.com (194.237.109.72): icmp_seq=1. time=154. ms
64 bytes from rc3.europe.yahoo.com (194.237.109.72): icmp_seq=2. time=176. ms
64 bytes from rc3.europe.yahoo.com (194.237.109.72): icmp_seq=3. time=159. ms
64 bytes from rc3.europe.yahoo.com (194.237.109.72): icmp_seq=4. time=161. ms
^C

----homerc.europe.yahoo.com PING Statistics----
5 packets transmitted, 5 packets received, 0% packet loss
round-trip (ms) min/avg/max = 154/162/176

• Note lots of variance in the RTT: this is typical
The Internet/Network Layer

ICMP Ping

% ping -R www.yahoo.co.uk
PING homerc.europe.yahoo.com: 56 data bytes
64 bytes from rc3.europe.yahoo.com (194.237.109.72): icmp_seq=0. time=168. ms
  IP options: <record route> 138.38.29.254, bath-gw-1.bwe.net.uk
          (194.82.125.198), man-gw-2.bwe.net.uk (194.82.125.210), bristol.
          bweman.site.ja.net (146.97.252.102), south-east-gw.bristol-core.j
          a.net (146.97.252.62), south-east-gw.ja.net (193.63.94.50), 212.1
          .192.150, se-uk.uk.ten-155.net (212.1.192.110), se-aucs.se.ten-155
          .net (212.1.194.25)

• Some versions of ping can enable the IP header option record route: this makes IP save
  the address of each intermediate router in the header
The Internet/Network Layer

ICMP Ping

% ping -R www.yahoo.co.uk
PING homerc.europe.yahoo.com: 56 data bytes
64 bytes from rc3.europe.yahoo.com (194.237.109.72): icmp_seq=0. time=168. ms
  IP options: <record route> 138.38.29.254, bath-gw-1.bwe.net.uk
  (194.82.125.198), man-gw-2.bwe.net.uk (194.82.125.210), bristol.
  bweman.site.ja.net (146.97.252.102), south-east-gw.bristol-core.j
  a.net (146.97.252.62), south-east-gw.ja.net (193.63.94.50), 212.1
  .192.150, se-uk.uk.ten-155.net (212.1.192.110), se-aucs.se.ten-155
  .net (212.1.194.25)

• But only 60 bytes of options, giving space for up
to 9 addresses (overheads of option header
and other bits and pieces), so only 9 addresses
are recorded
The Internet/Network Layer

Traceroute

• There are lots of routes of more than 9 hops, so using ping to discover a route is limited

• Traceroute is a clever way to find routes by deliberately generating errors and looking at the ICMP messages that result

• It sends a packet to the intended destination, but with an artificially small time-to-live
The Internet/Network Layer

Traceroute

• When the TTL drops to zero en route, the packet is dropped and an ICMP “TTL exceeded” is returned

• The source address on the ICMP error tells us where the packet had got to

• Repeat for other values of TTL to get the entire route
The Internet/Network Layer

Traceroute

0. Send a packet to the desired destination but with TTL of 1

1. This reaches the first gateway/router, the TTL is decremented to 0. The router drops the packet and returns a “TTL exceeded”

2. This reaches the source, which records where the packets came from, namely the router
The Internet/Network Layer

Traceroute

3. Send a packet with TTL 2. This gets to the next router before the TTL drops to 0, and the ICMP response identifies the second router.

4. Send a packet with TTL 3, then 4, and so on until a packet reaches the destination. At each stage we get an ICMP reply telling us the router the packet reached.
The Internet/Network Layer

Traceroute

5. When a packet reaches the destination, it will be rejected with an ICMP “port unreachable”. This is the sign we can stop
% traceroute mary.bath.ac.uk
traceroute to mary.bath.ac.uk (138.38.32.14), 30 hops max, 46 byte packets

<table>
<thead>
<tr>
<th>Hop</th>
<th>IP Address</th>
<th>Time 1</th>
<th>Time 2</th>
<th>Time 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>136.159.7.1 (136.159.7.1)</td>
<td>0.779 ms</td>
<td>1.131 ms</td>
<td>0.642 ms</td>
</tr>
<tr>
<td>2</td>
<td>136.159.28.1 (136.159.28.1)</td>
<td>1.369 ms</td>
<td>0.910 ms</td>
<td>1.489 ms</td>
</tr>
<tr>
<td>3</td>
<td>136.159.30.1 (136.159.30.1)</td>
<td>2.339 ms</td>
<td>1.937 ms</td>
<td>0.988 ms</td>
</tr>
<tr>
<td>4</td>
<td>136.159.251.2 (136.159.251.2)</td>
<td>1.458 ms</td>
<td>1.071 ms</td>
<td>1.831 ms</td>
</tr>
<tr>
<td>5</td>
<td>192.168.47.1 (192.168.47.1)</td>
<td>1.434 ms</td>
<td>1.554 ms</td>
<td>1.008 ms</td>
</tr>
<tr>
<td>6</td>
<td>192.168.3.25 (192.168.3.25)</td>
<td>29.192 ms</td>
<td>30.094 ms</td>
<td>25.374 ms</td>
</tr>
<tr>
<td>7</td>
<td>REGIONAL2.tac.net (205.233.111.67)</td>
<td>25.413 ms</td>
<td>33.002 ms</td>
<td>32.677 ms</td>
</tr>
<tr>
<td>8</td>
<td>* * *</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>117.ATM3-0.XR2.CHI6.ALTER.NET (146.188.209.182)</td>
<td>82.403 ms</td>
<td>58.747 ms</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>190.ATM11-0-0.GW4.CHI6.ALTER.NET (146.188.209.149)</td>
<td>56.376 ms</td>
<td>67.898 ms</td>
<td>73.462 ms</td>
</tr>
<tr>
<td>11</td>
<td>if-4-0-1-1.bb1.Chicago2.Teleglobe.net (207.45.193.9)</td>
<td>66.853 ms</td>
<td>46.089 ms</td>
<td>44.670 ms</td>
</tr>
<tr>
<td>12</td>
<td>if-0-0.core1.Chicago3.Teleglobe.net (207.45.222.213)</td>
<td>48.817 ms</td>
<td>* 75.093 ms</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>if-8-1.core1.NewYork.Teleglobe.net (207.45.222.209)</td>
<td>106.198 ms</td>
<td>94.249 ms</td>
<td>73.375 ms</td>
</tr>
<tr>
<td>14</td>
<td>ix-5-3.core1.NewYork.Teleglobe.net (207.45.202.30)</td>
<td>75.286 ms</td>
<td>89.873 ms</td>
<td>98.789 ms</td>
</tr>
<tr>
<td>15</td>
<td>us-gw.ja.net (193.62.157.13)</td>
<td>143.686 ms</td>
<td>159.212 ms</td>
<td>166.020 ms</td>
</tr>
<tr>
<td>16</td>
<td>external-gw.ja.net (193.63.94.40)</td>
<td>172.803 ms</td>
<td>189.216 ms</td>
<td>191.260 ms</td>
</tr>
<tr>
<td>17</td>
<td>external-gw.bristol-core.ja.net (146.97.252.58)</td>
<td>206.403 ms</td>
<td>185.438 ms</td>
<td>192.989 ms</td>
</tr>
<tr>
<td>18</td>
<td>bristol.bweman.site.ja.net (146.97.252.102)</td>
<td>196.685 ms</td>
<td>206.221 ms</td>
<td>183.763 ms</td>
</tr>
<tr>
<td>19</td>
<td>man-gw-2.bwe.net.uk (194.82.125.210)</td>
<td>197.968 ms</td>
<td>* 174.809 ms</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>bath-gw-1.bwe.net.uk (194.82.125.198)</td>
<td>209.307 ms</td>
<td>221.512 ms</td>
<td>199.168 ms</td>
</tr>
<tr>
<td>21</td>
<td>* * *</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>mary.bath.ac.uk (138.38.32.14)</td>
<td>250.670 ms</td>
<td>* 186.400 ms</td>
<td></td>
</tr>
</tbody>
</table>
The Internet/Network Layer

Traceroute

- The traceroute command sends *three* probes for each stage

6. ***

No error packet was received for this TTL. Many possible reasons. E.g., some implementations return an ICMP with the same TTL as was left in the original packet. This is guaranteed not to reach us
The Internet/Network Layer

Traceroute

• If the last half of the routes are *s, the destination has this bug. The TTLs are ramped up until double the path length before we get the ICMP replies

• Also possible on a long route is the router is setting a TTL too small to reach us

• A common possibility is that the router refuses to send ICMP errors for TTL exceeded in a misguided attempt at security
The Internet/Network Layer

Traceroute

8. * 117.ATM3-0.XR2.CHI6.ALTER.NET (146.188.209.182) 82.403 ms 58.747 ms

• A * before the name means the name lookup took so long traceroute decided to stop waiting and carry on. The name subsequently turned up

• Sometimes the same line is repeated: this is because some routers forward packets with TTL 0. This is a bug
The Internet/Network Layer

ICMP

- There are many bugs out there in the real world!
- ICMP errors must contain the IP header and at least 8 bytes of the original data in the packet that caused the problem. This is so the source machine can match up the ICMP packet with the original packet. Eight bytes is just enough to contain the interesting parts of the next layer headers (UDP and TCP)