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

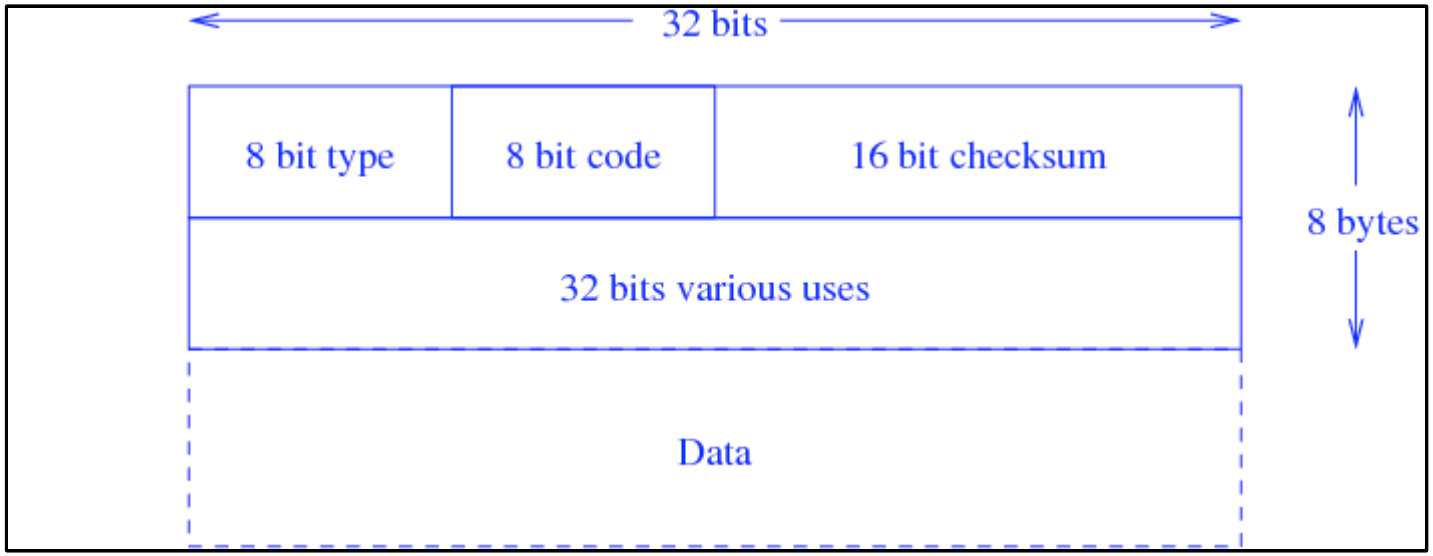
The Internet/Network Layer

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

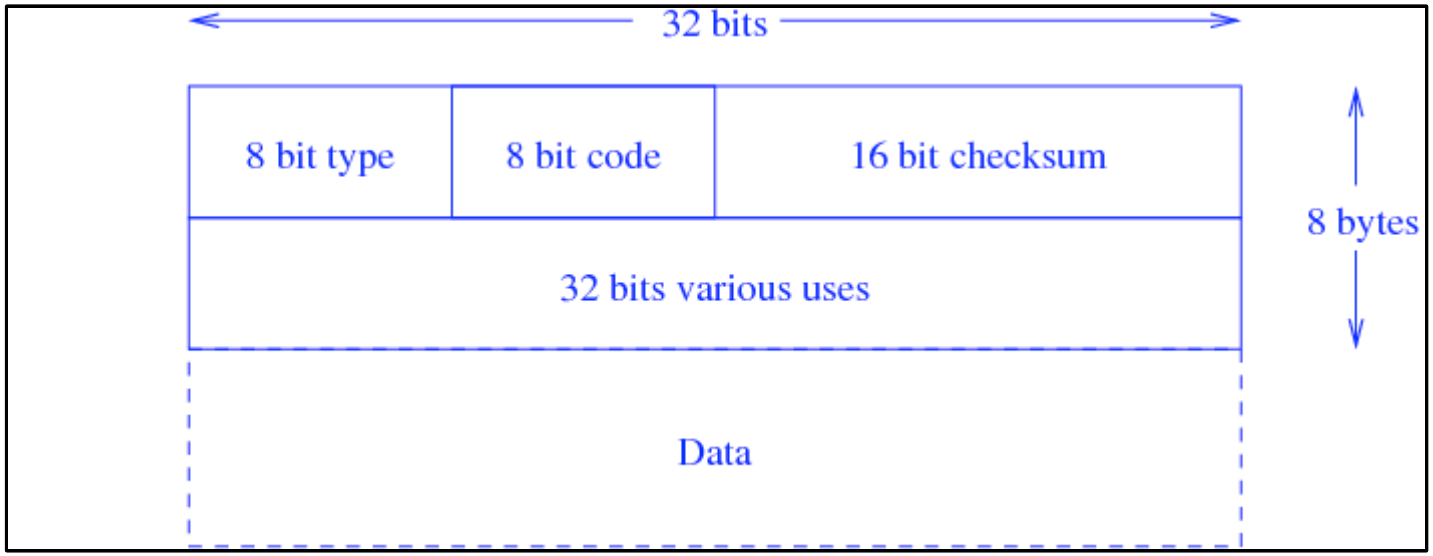
ICMP



- 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

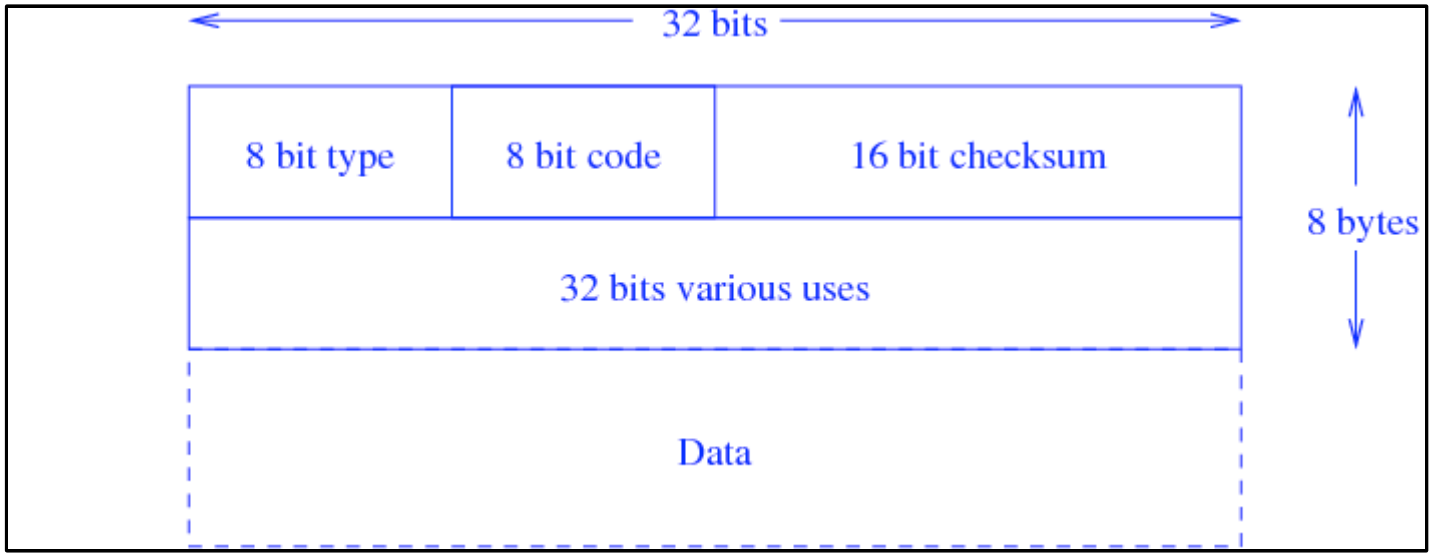
ICMP



- 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 a ICMP packet)
  - a packet whose destination is a broad/multicast
  - a packet whose source is a broad/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

Type	Err	Code
ECHOREPLY		reply from a ping
DEST_UNREACH	e	network unreachable
	e	host unreachable
	e	port unreachable
	e	fragmentation wanted but DF set
REDIRECT	e	routing redirect for a network
	e	routing redirect for a host
ECHO		ping
TIME_EXCEEDED	e	TTL reached 0
	e	fragment reassembly time exceeded

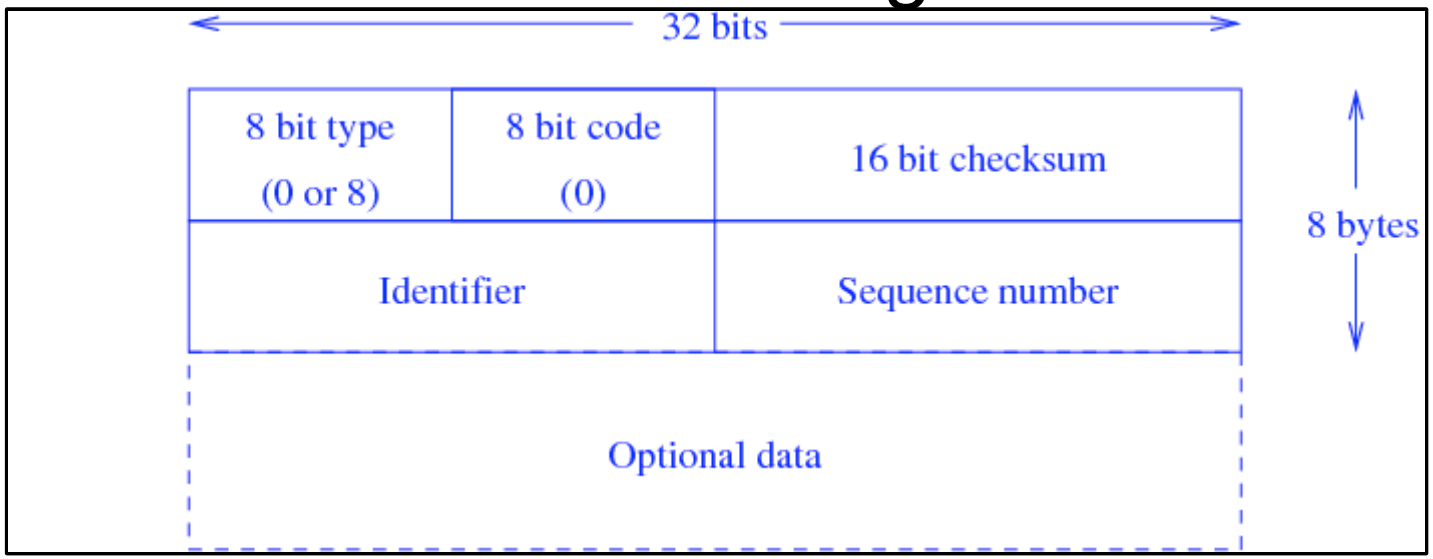
The Internet/Network Layer

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

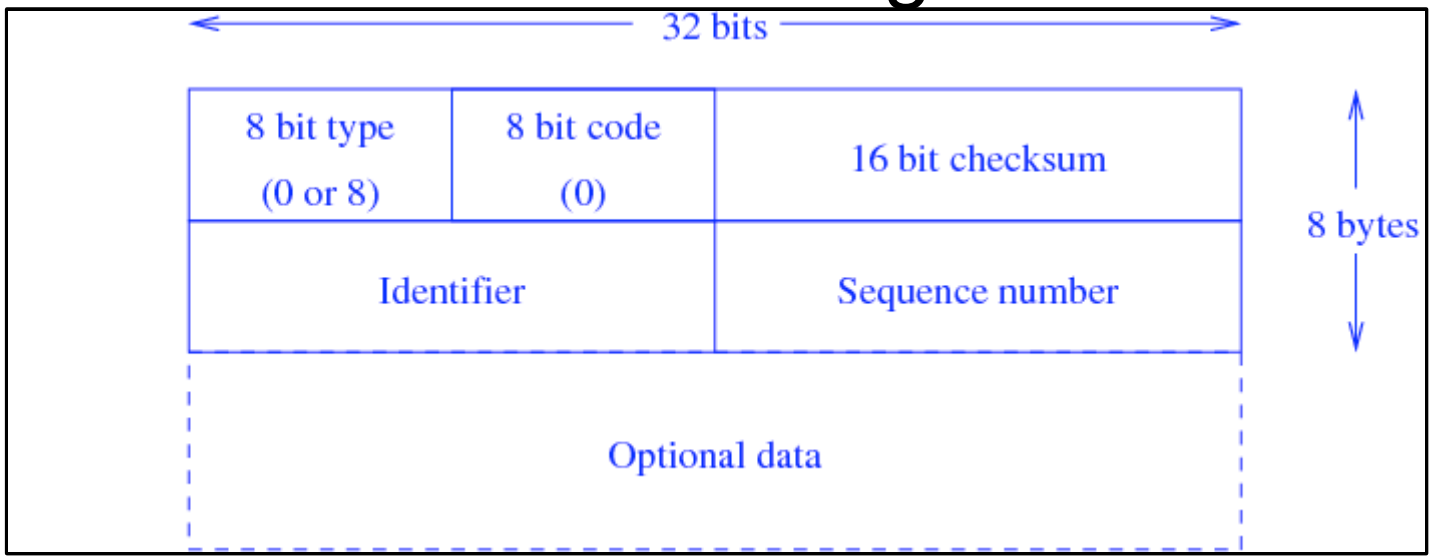
ICMP Ping



- 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

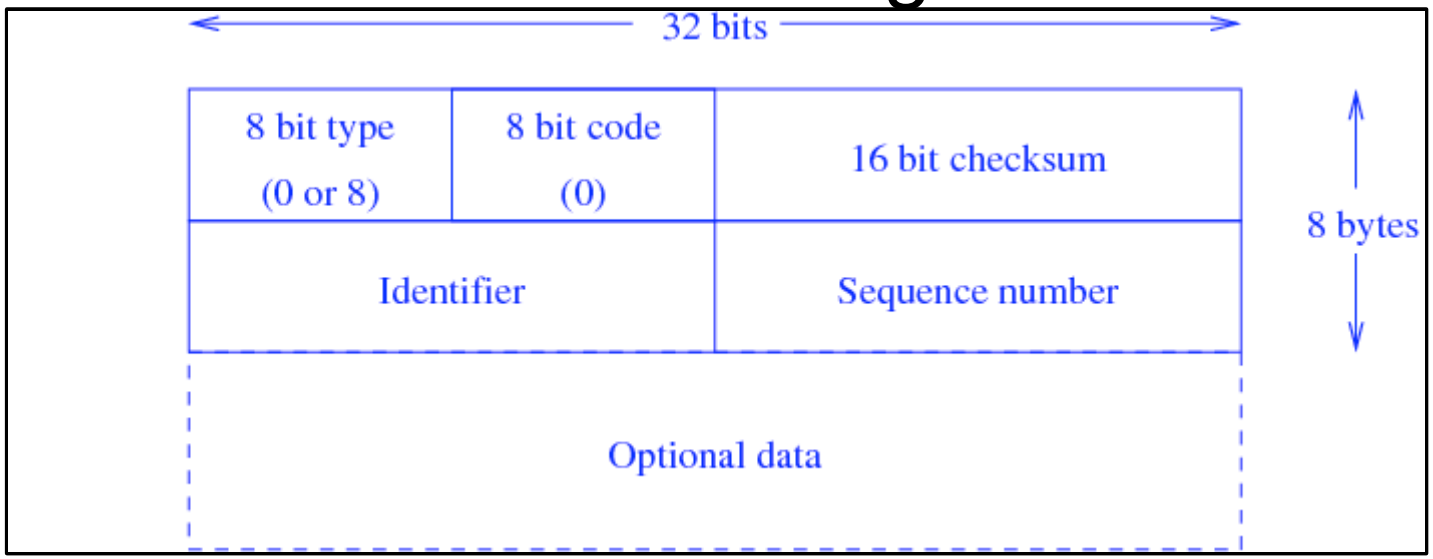
ICMP Ping



- 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



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

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)