IK1350 Protocols in Computer Networks/ Protokoll i datornätverk Spring 2008, Period 3

Module 4: IP, ICMP, and Tools

Lecture notes of G. Q. Maguire Jr.

KTH Information and Communication Technology

For use in conjunction with *TCP/IP Protocol Suite*, by Behrouz A. Forouzan, 3rd Edition, McGraw-Hill, 2006.

For this lecture: Chapters 8-9

IP, ICMP, and Tools Outline

- IP
- ICMP
- Useful Diagnostic Tools
  - Ping
  - Traceroute
  - tcpdump and ethereal
  - sock

Internet Protocol version 4 (IPv4) (RFC 791)

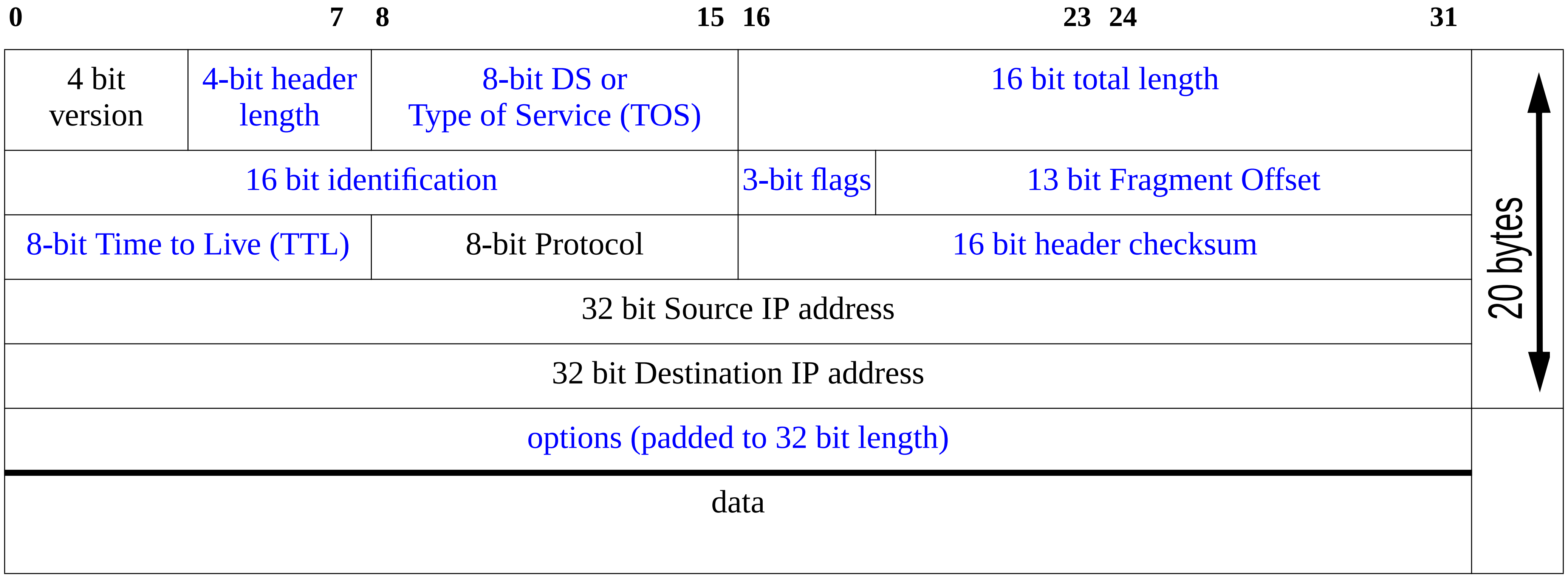


Figure 44: IP header (see Stevens, Vol. 1, figure 3.1, pg. 34)

Note: We examined the Version, Protocol, and IP address fields in the previous lecture. Today we will examine the length fields, TOS, identification, flags, offset, checksum, and options fields.

Length Fields

- Header Length (4 bits)
    - Size of IPv4 header including IP options
    - Expressed in number of 32-bit words (4-byte words)
    - Minimum is 5 words (i.e., 20 bytes)
    - Maximum is 15 words (i.e., 60 bytes)
        - limited size ⇒ limited use
- Total Length (16 bits)
    - Total length of datagram **including** header
    - If datagram is fragmented: length of this fragment
    - Expressed in bytes
    - Hosts **only** have to accept packets up to 576 bytes in size
    - Maximum: 65,535 bytes
        - Most modern systems accept slightly larger than 8,196 + header bytes (to provide efficient file service for 8 Kbyte blocks)
        - Note: Some systems **only** accept this much!

MTU≡Maximum Transmission Unit

MTU is a characteristic of the link layer

Typical MTUS

MTU	**Network**	
65535	Official maximum MTU	
17914	16Mbps IBM Token Ring	
8166	IEEE 802.4	
4464	IEEE 802.5 (4Mbps max)	
4352	FDDI (Revised)	
2048	Wideband Network	
2002	IEEE 802.5 (4Mb recommended)	
1536	Experimental Ethernet Nets	
1500	Ethernet Networks	
1500	Point-to-Point (default)	
1492	IEEE 802.3	
1006	SLIP	simply a logical limit for interactive response
1006	ARPANET	
576	X.25 Networks	⇐ we will see this number again!
544	DEC IP Portal	
512	NETBIOS	
508	IEEE 802/Source-Route Bridge	
296	Point-to-Point (low delay)	
68	Official minimum MTU	

Fragmentation

- If an IP datagram is larger than the MTU of the link layer, it must be divided into several pieces ⇒ fragmentation
- Fragmentation may occur multiple times
  - as a fragment might need to go across a link with an even smaller MTU!
- Both hosts and routers may fragment
  - **However, only** destination host reassemble!
    - as fragments only need to come together at the final host - thus the fragments can take different paths though the network
    - also reassembly requires waiting for the other fragments - so doing this earlier in the network could add unnecessary delay
  - Each fragment is routed separately (i.e., as independent datagram)
- TCP uses either 576 byte MTU or path MTU discovery

Fields relevant to Fragmentation

- Identification (16 bits)
  - Identification + source IP address *uniquely* identifies each datagram sent by a host $\Rightarrow$ Identification field is copied to all fragments of a datagram upon fragmentation (since they are all part of the same original datagram)
- Flags: 3 bits
  - Reserved Fragment (RF) - set to 0
  - Don’t Fragment (DF)
    - Set to 1 if datagram should **not** be fragmented
    - If set and fragmentation needed $\Rightarrow$ datagram will be **discarded** and an **error message** will be returned to the sender
  - More Fragments (MF)
    - Set to 1 for all fragments, **except** the last
- Fragmentation Offset (13 bits)
  - 8-byte units: (i.e., the byte offset is ip_frag << 3)
  - indicates relative position of a fragment with respect to the whole datagram

Fragments can overlap - the receiver simply assembles what it receives (ignoring duplicate parts).

If there are gaps - then at some point there will be a re-assembly error.

Path MTU

- Each link in path from source to destination can have a different MTU
- to avoid fragmentation you have to find the minimum of these
- RFC 1191: Path MTU discovery[27] uses:
  - “good” guesses (i.e., likely values)
  - By setting Don’t Fragment (DF) bit in IP datagram $\Rightarrow$ change size while you get ICMP messages saying “Destination Unreachable” with a code saying fragmentation needed

Serial line throughput

At 9,000 bits/sec, 8 bits per byte, plus 1 start and 1 stop bit, i.e., 960 bytes/sec, then transferring 1024 byte packets would take 1066 ms

- too long for interactive limits; since the average wait would be 533 ms

∴ shorten the MTU to 296 bytes ⇒ 266 ms/frame or ~133 ms average wait

With 5 bytes of CSLIP header and 256 bytes of data (in the 261 byte frame) ⇒

- 98.1% utilization of link for data and
- 1.9% for header

For single bytes of interactive traffic, the round trip-time is 12.5 ms

Caveats:

- assumes that you give interactive traffic priority
- error correcting and compression in the modem can complicate the calculations - since the modem has to delay traffic to have more to compress and compression takes time

Differentiated Services (DS) & Type of Service

Type of Service (TOS): 8 bits

Bits 0-2:	Precedence	
Bit 3:	0 = normal Delay	1 = Low Delay
Bit 4:	0 = normal Throughput	1 = High Throughput
Bit 5:	0 = normal Relibility	1 = High Relibility
Bit 6:	0 = normal monetary Cost	1 = minimize monetary Cost.

0	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Precedence			DELAY	T	R	C	Reserved

- Few applications set the TOS field (in fact most implementations will not let you set these bits!) However, 4.3BSD Reno and later - do support these bits.
- Differentiated Services (diffserv) proposes to use 6 of these bits to provide 64 priority levels - calling it the Differentiated Service (DS) field [RFC2474] (using bits 0..5 as Differentiated Services CodePoint (DSCP))
- SLIP guesses by looking at the **protocol** field and then checks the source and destination **port** numbers.

There has been a lot of experimentation with this field, both for TOS and more recently for Early Congestion Notification (ECN): RFC 3168 [26] using bits 6 and 7 {ECN Capable Transport (ECT) and Congestion Experienced (CE)}.

Recommended Value for TOS Field

Application		Minimum delay	Maximize throughput	Maximize reliability	Minimize monitary cost	Hex value[a]
telnet/rlogin		1	0	0	0	0x10
FTP	control	1	0	0	0	0x10
	data	0	1	0	0	0x08
	any bulk data	0	1	0	0	0x08
TFTP		1	0	0	0	0x10
SMTP	command phase	1	0	0	0	0x10
	data phase	0	1	0	0	0x08
DNS	UDP query	1	0	0	0	0x10
	TCP query	0	0	0	0	0x00
	zone transfer	0	1	0	0	0x08
ICMP	error	0	0	0	0	0x00
	query	0	0	0	0	0x00
any IGP		0	0	1	0	0x04
SNMP		0	0	1	0	0x04
BOOTP		0	0	0	0	0x00
NNTP		0	0	0	1	0x02

a. Note that this is the hex value as see in the TOS/DS byte.

See also Table 8.2 on page 182 of Forouzan.

Precedence

Precedence values are defined but are largely ignored, few applications use them.

111	Network Control
110	Internetwork Control
101	CRITIC/ECP
100	Flash Override
011	Flash
010	Immediate
001	Priority
000	Routine

In the original ARPANET there were two priority levels defined (in order to support low delay services and regular traffic).

Problems with precedence

- As soon as people found that high priority meant something $\Rightarrow$ all traffic was sent with this bit set!

So unless there is a added cost/policy check/… associated with usage of a precedence level - it is very likely going to be abused.

Precendence and telephony systems

Similar precedence systems exist in most national telephony systems.

Q: What are the A, B, C and D touch tone keys used for? …

A: These are extensions to the standard touch-tones (0-9, *, #) which originated with the U.S. military's Autovon phone network. The original names of these keys were FO (Flash Override), F (Flash), I (Immediate), and P (Priority). The various priority levels established calls with varying degrees of immediacy, terminating other conversations on the network if necessary. FO was the greatest priority, normally reserved for the President or very high ranking officials. P had a lesser priority, but still took precedence over calls that were placed without any priority established.

-- from TELECOM Digest - Frequently Asked Questions - v.8, 8 February 1997

Differentiated services

If bits 3, 4, and 5 are all zero (i.e., XXX000) ⇒ treat the bits 1, 2, 3 as the traditional precedence bits, else the 6 bits define 64 services:

- Category 1: numbers 0, 2, 4, … 62 - defined by IETF
- Category 2: numbers 3, 7, 11, 15, … 63 defined by local authorities
- Category 3: numbers 1, 5, 9, … 61 are for temporary/experimental use

The numbering makes more sense when you see them as bit patterns:

Category	codepoint	Assigning authority
1	XXXXX0	IETF
2	XXXX11	local
3	XXXX01	temporary/experimental

The big problems occur at gateways where the intepretation of local DS values is different on the incoming and outgoing links!

TTL field

Time To Live (TTL) (8 bits):

- Limits the lifetime of a datagram, to avoid infinite loops
- A router receiving a packet with TTL>1 decrements the TTL field and forwards the packet
- If TTL <= 1 shall not be forwarded
  ⇒ an ICMP time exceeded error is returned to the sender {we will cover ICMP shortly}
- Recommended value is 64
- Should really be called Hop Limit (as in IPv6)

Historically: Every router holding a datagram for more than 1 **second** was expected to decrement the TTL by the *number of seconds* the datagram resided in the router.

Header Checksum

- Ensures integrity of header fields
  - Hop-by-hop (not end-to-end)
  - Header fields must be correct for proper and safe processing of IP!
  - Payload is **not** covered
- Other checksums
  - Hop-by-hop: using link-layer CRC
    - IP assumes a strong link layer checksum/CRC - as the IP checksum is weak
  - End-to-end: Transport layer checksums, e.g., TCP & UDP checksums, cover **payload**
- Internet Checksum Algorithm, RFC 1071
  - Treat headers as sequence of 16-bit integers
  - Add them together
  - Take the one’s complement of the result

Note that recent work concerning IP over wireless links assume that the payload can have errors and will still be received (see work concerning selective coverage of UDP checksum).

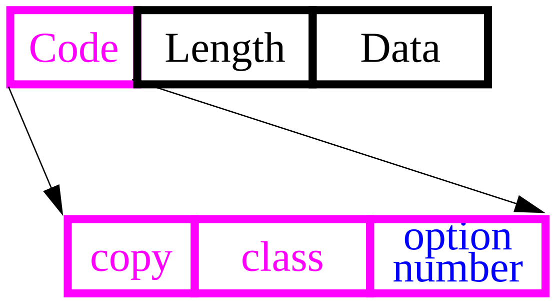
IPv4 Options

- IPv4 options were intended for network testing & debugging
- Options are variable sized and follow the fixed header
- Contiguous (i.e., no separators)
- Not required fields, but all IP implementations **must** include processing of options
  - Unfortunately, many implementations do not!
- Maxium of 40 bytes available $\Rightarrow$ very limited use
  - Since the maximum header length is 60 bytes and the fixed part is 20 bytes - there is very little space left!

IP Options Encoding

Two styles:

- Single byte (only code)
- Multiple byte



Option Code: 1 byte

- Copy (to fragments) (1 bit)
  - 0: copy only to the first fragment
  - 1: copy the option to all fragements
- Class (2 bits)
  - 0 (00): Datagram or network control
  - 2 (10): Debugging and measurement
  - 1 (01) and 3 (11) reserved
- Option Number (5 bits)

Option Length: 1 byte, defines total length of option

Data: option specific

Categories of IP Options

- Single byte (only code)
  - No operation (Option Number=0)
  - End of operation (Option Number=1)
- Multiple byte
  - Loose Source Route (Option Number=3)
    - Path includes these router, but there can be multiple hops between the specified addresses
  - Time stamp (Option Number=4)
    - Like record route (below), but adds a timestamp at each of the routers (upto the space available - after this an overflow field is incremented - but it is only 4 bits)
  - Record Route (Option Number=7)
  - Strict Source Route (Option Number=9)
    - The exact path is specified

However, due to the very limited space available for the options - these options are of little practical value in todays internet. (Consider the diameter of today’s internet versus the number of IP addresses or timestamps that could be in the options field; i.e., record route can only store 9 IP addresses!)

Internet Control Message Protocol (ICMP)

ICMP [28] is part of the same level as IP, but uses IP for transfers! ICMP is used by layer 3 entities to communicate with each other.

- ICMP PDU: type (8 bits); code (8 bits); checksum (16 bits); parameters (n*32 bits); information (variable length)
  for errors: the information field always includes the first 64 bits of the data field of the original datagram which caused the ICMP message
- ICMP messages include:
  - Destination Unreachable (Network/Host/Protocol/Port/…)
  - Time Exceeded (TTL expired)
  - Parameter problem - IP header error
  - Source Quench (requests source to decrease its data rate)
  - Redirect - tell source to send its messages to a “better address”
  - Echo Request/ Echo reply - for testing (e.g., “ping” program sends an Echo request)
  - Timestamp Request/ Timestamp reply
  - Information Request / Information reply
  - Address Mask Request / Reply
  - Traceroute
  - Datagram conversion error
  - Mobile Host Redirect/Registration Request/Registration Reply
  - IPv6 Where-Are-You/I-Am-Here

ICMP Port Unreachable Error

Example: (Stevens, Vol. 1, Section 6.5, pp. 77-78)

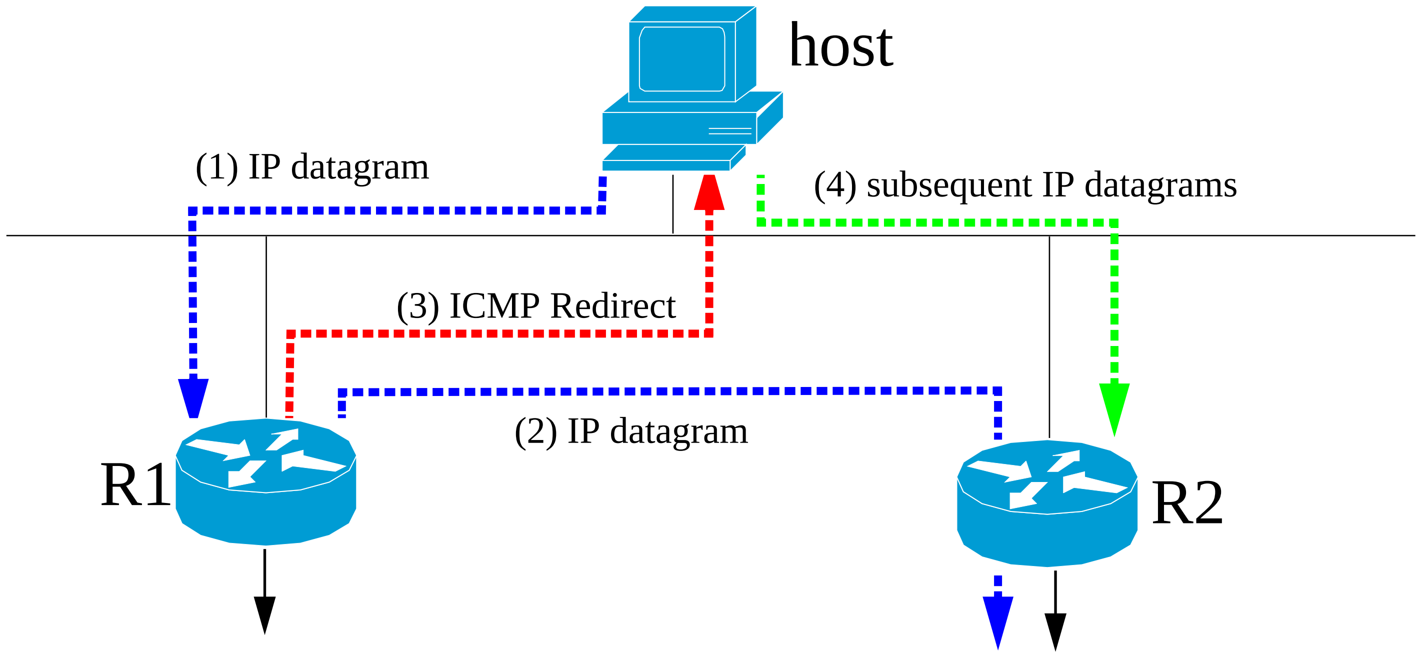
```
bsdi% tftp
tftp> connect svr4 888  specify host and port number
tftp> get temp.foo      try to fetch a file
Transfer times out.     about 25s later
tftp> quit
```

tcpdump output

```
1     0.0                     arp who-has svr4 tell bsdi
2     0.002050     (0.0020)   arp reply svr4 is-at 0:0:c0:c2:9b:26
3     0.002723     (0.0007)   bsdi.2924 > svr4.8888 udp 20
4     0.006399     (0.0037)   svr4 > bsdi: icmp: svr4 udp port 8888 unreachable
5     5.000776     (4.9944)   bsdi.2924 > svr4.8888 udp 20
6     5.004304     (0.0035)   svr4 > bsdi: icmp: svr4 udp port 8888 unreachable
…                             repeats every 5 seconds
11    20.001177    (4.9966)   bsdi.2924 > svr4.8888 udp 20
12    20.004759    (0.0036)   svr4 > bsdi: icmp: svr4 udp port 8888 unreachable
```

ICMP Redirect

ICMP Redirect message is sent by a router (R1) to the sender of an IP datagram (host) when the datagram should have been sent to a different router (R2).



PING: Packet InterNet Groper or sonar echo

Ping was written by Mike Muuss[1] to test host reach-ability. Uses ICMP,most IP implementations support Ping server. Sends an ICMP echo request to a host

Format of ICMP message for Echo request/reply (see Stevens, Vol. 1, figure 7.1, pg. 86)

Type (0 or 8)	code (0)	16 bit checksum
16 bit identifier		16 bit sequence number
Optional data		

Look at ping across different connections[2]:

- LAN
- WAN
- Hardwired SLIP
- Dialup SLIP - extra delay due to the modems and the correction/compression

With IP record route (RR) option tracing the route of the ping datagram.

1. Mike Muuss was killed in an automobile accident on November 20, 2000. *http://ftp.arl.mil/~mike/*

2. For examples, see Stevens, Vol. 1, Chapter 7, pp. 86-90.

PING examples

On a Solaris machine:

```
bash-2.03$ /usr/sbin/ping cyklop.nada.kth.se
```

from a machine at IMIT

```
   cyklop.nada.kth.se is alive
bash-2.03$ /usr/sbin/ping -s cyklop.nada.kth.se
   PING cyklop.nada.kth.se: 56 data bytes
   64 bytes from cyklop.nada.kth.se (130.237.222.71): icmp_seq=0. time=3. ms
   64 bytes from cyklop.nada.kth.se (130.237.222.71): icmp_seq=1. time=1. ms
   64 bytes from cyklop.nada.kth.se (130.237.222.71): icmp_seq=2. time=1. ms
   ^C
   ----cyklop.nada.kth.se PING Statistics----
   3 packets transmitted, 3 packets received, 0% packet loss
   round-trip (ms)  min/avg/max = 1/1/3
```

Why did the first ping take longer?

On a HP-UX 11.0 machine:

ping -ov www.kth.se *from a machine on Telia’s ADSL network*

```
PING www.kth.se: 64 byte packets
64 bytes from 130.237.32.51: icmp_seq=0. time=54. ms
64 bytes from 130.237.32.51: icmp_seq=1. time=38. ms
64 bytes from 130.237.32.51: icmp_seq=2. time=11. ms
64 bytes from 130.237.32.51: icmp_seq=3. time=11. ms
64 bytes from 130.237.32.51: icmp_seq=4. time=11. ms
^C
----www.kth.se PING Statistics----
5 packets transmitted, 5 packets received, 0% packet loss
round-trip (ms)  min/avg/max = 11/25/54
5 packets sent via:
    217.208.194.247 - fls31o268.telia.com
    213.64.62.150   - fre-d4-geth6-0.se.telia.net
    213.64.62.154   - fre-c3-geth6-0.se.telia.net
    195.67.220.1    - fre-b1-pos0-1.se.telia.net
    130.242.94.4    - STK-PR-2-SRP5.sunet.se
    130.242.204.130 - STK-BB-2-POS4-3.sunet.se
    130.242.204.121 - stockholm-1-FE1-1-0.sunet.se
    130.237.32.3    - [ name lookup failed ]
    130.237.32.51   - oberon.admin.kth.se
```

this is based on the record route information (caused by “-ov”)

Ping with record route option

$ ping -R www.kth.se

```
PING www.kth.se (130.237.32.51) 56(124) bytes of data.
64 bytes from oberon.admin.kth.se (130.237.32.51): icmp_seq=1
ttl=253 time=2.50ms
RR:     ccsser2 (130.237.15.248)
        ke4-ea4-p2p.gw.kth.se (130.237.211.50)
        kthlan-gw-32-2.admin.kth.se (130.237.32.2)
        oberon.admin.kth.se (130.237.32.51)
        oberon.admin.kth.se (130.237.32.51)
        ea4-ke4-p2p.gw.kth.se (130.237.211.49)
        130.237.15.194
        ccsser2 (130.237.15.248)

64 bytes from oberon.admin.kth.se (130.237.32.51): icmp_seq=2
ttl=253 time=1.73ms      (same route)
64 bytes from oberon.admin.kth.se (130.237.32.51): icmp_seq=3
ttl=253 time=1.80ms      (same route)
64 bytes from oberon.admin.kth.se (130.237.32.51): icmp_seq=4
ttl=253 time=1.90ms      (same route)
```

Useful Tool: Traceroute Programs

Developed by Van Jacobson to see the route that IP datagrams follow from one host to another. Traceroute uses ICMP, TTL field, and an *unreachable UDP port*.

```
svr % traceroute slip
traceroute to slip (140.252.13.65), 30 hops max, 40 byte packets
1 bsdi (140.252.13.35) 20 ms 10 ms 10 ms         20 ms due to ARP
2 slip (140.252.13.65) 120 ms 120 ms 120 ms
```

tcpdump output

1	0.0		arp who-has bsdi tell svr4
2	0.000586	(0.0006)	arp reply bsdi is-at 0:0:c0:6f:2d:40
3	0.003067	(0.0025)	svr4.42804 > slip.33435 udp 12 [ttl 1]
4	0.004325	(0.0013)	bsdi > svr4: icmp: time exceeded in-transit
5	0.069810	(0.0655)	svr4.42804 > slip.33436 udp 12 [ttl 1]
6	0.071149	(0.0013)	bsdi > svr4: icmp: time exceeded in-transit
7	0.085162	(0.0140)	svr4.42804 > slip.33437 udp 12 [ttl 1]
8	0.086375	(0.0012)	bsdi > svr4: icmp: time exceeded in-transit
9	0.118608	(0.0322)	svr4.42804 > slip.33438 udp 12 *ttl=2*
10	0.226464	(0.1079)	slip > svr4: icmp: slip udp port 33438 unreachable
11	0.287296	(0.0608)	svr4.42804 > slip.33439 udp 12 *ttl=2*
12	0.395230	(0.1079)	slip > svr4: icmp: slip udp port 33439 unreachable
13	0.409504	(0.0608)	svr4.42804 > slip.33440 udp 12 *ttl=2*
14	0.517430	(0.1079)	slip > svr4: icmp: slip udp port 33440 unreachable

ICMP Summary

- Destination (Network/Host/Protocol/Port/...) Unreachable
- Time Exceeded - i.e., TTL expired
  - Used to implement traceroute
- Parameter problem - IP header error
- Source Quench- asks source to decrease its sending rate
- Redirect - tells the source to send packets to a “better” address
- Echo Request/Echo reply - for testing
  - ping: sends an Echo Request, then measures the time until the matching reply is received
- Timestamp Request/Reply
  - Round Trip Time (RTT) computation
  - Clock synchronization
- Address Mask Request/Reply
  - Allows diskless systems to learn their subnet mask
- Router Solicitation and Advertisment
  - Hosts query routers
  - Routers advertise presence and routes

The above is a partial summary of ICMP’s uses.

Summary

This lecture we have discussed:

- IP
- ICMP
- tools: Ping, Traceroute

References

[26] K. Ramakrishnan, S. Floyd, and D. Black, “The Addition of Explicit Congestion Notification (ECN) to IP”, IETF RFC 3168, September 2001.

[27] J. Mogul and S. Deering, “Path MTU Discovery”, IETF, RFC 1191, November 1990.

[28] J. Postel, Internet Control Message Protocol, IETF, RFC 792, September 1981 *http://www.ietf.org/rfc/rfc0792.txt*