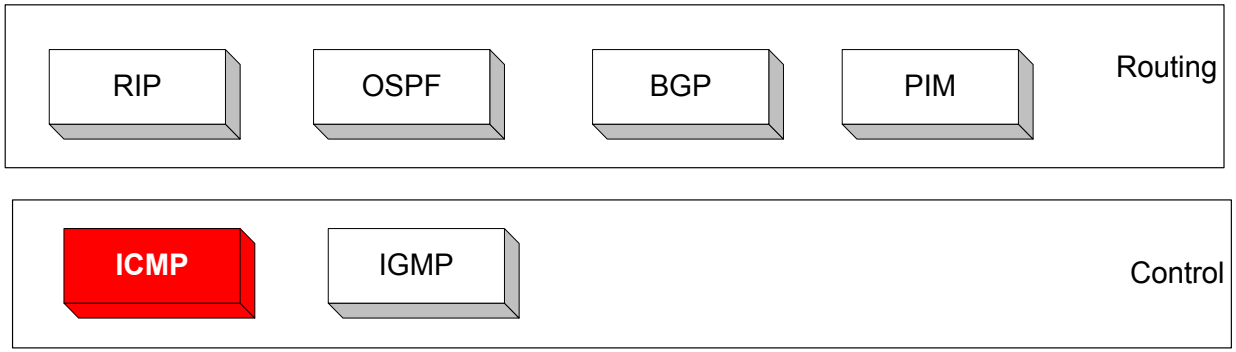
Internet Control Message Protocol (ICMP), RFC 792

Prof. Lin Weiguo

Oct. 2013

Overview

- The IP (Internet Protocol) relies on several other protocols to perform necessary control and routing functions:
  - Control functions (ICMP)
  - Multicast signaling (IGMP)
  - Setting up routing tables (RIP, OSPF, BGP, PIM, …)

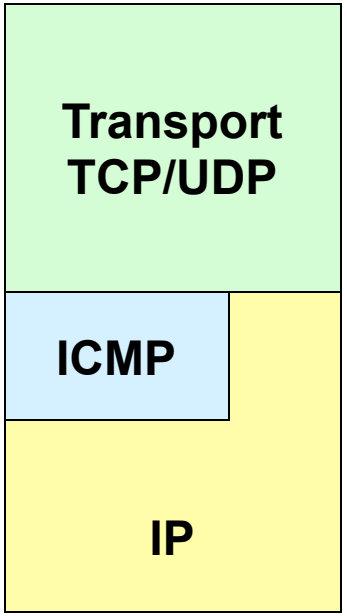


Purpose of ICMP

- The **Internet Control Message Protocol (ICMP)** is a helper protocol that supports IP with facility for
  - Error reporting
  - Simple queries

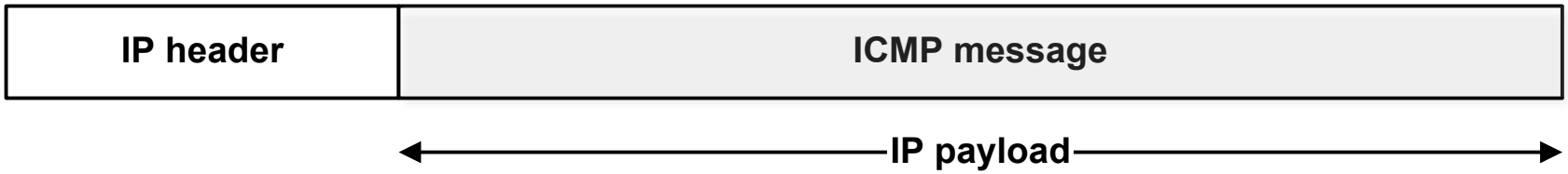
Layering view

- From a layering point of view, ICMP is a separate protocol that sits above IP and uses IP to transport messages.
- In practice, ICMP is an integral part of IP and all IP modules must support the ICMP protocol.
- ICMP datagrams are encapsulated within IP datagrams and processed by IP in the same way as TCP and UDP datagrams;



Message Encapsulation

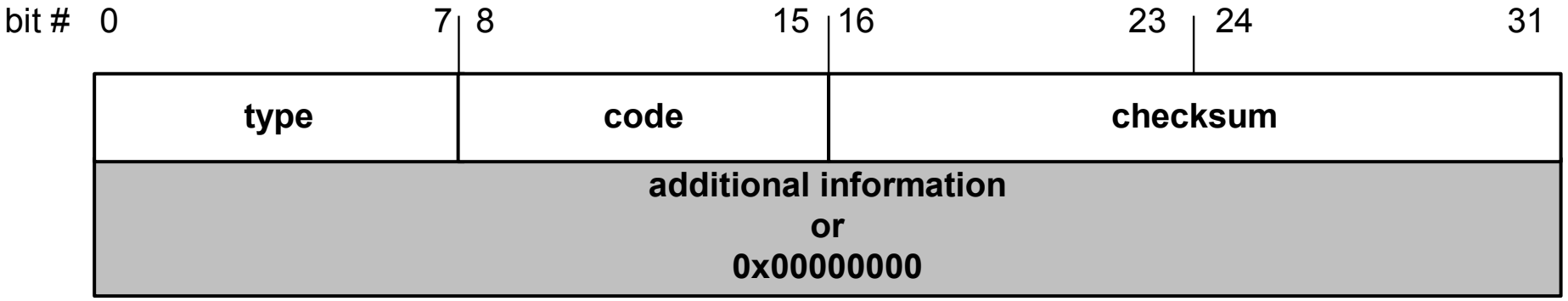
- ICMP messages are encapsulated as IP datagrams:



Protocol field:

ICMP:00000001
IGMP:00000010
TCP: 00000110
UDP: 00010001

ICMP message format



4 byte header:

- Type (1 byte): type of ICMP message
- Code (1 byte): subtype of ICMP message
- Checksum (2 bytes): similar to IP header checksum. Checksum is calculated over entire ICMP message

If there is no additional data, there are 4 bytes set to zero.

→ each ICMP messages is at least 8 bytes long

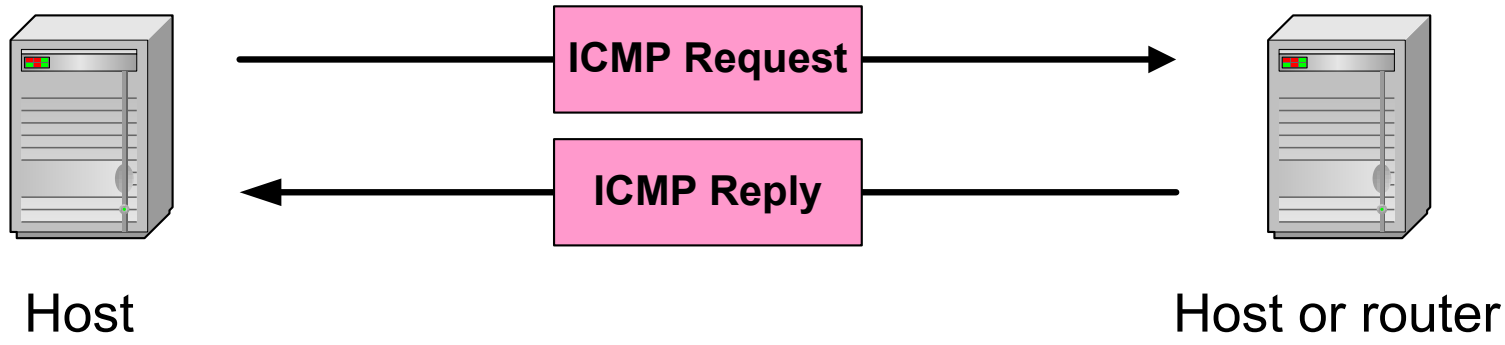
Types of ICMP Msges

- Two general types of ICMP messages:
  - Information messages, where a sender sends a query to another machine (either host or gateway) and expects an answer. For example, a host might want to know if a gateway is alive.
  - Error indication messages, where the IP software on a host or gateway has encountered a problem processing an IP datagram. For example, it may be unable to route a datagram to its destination, or it may have had to drop a frame.

ICMP messages type/code

Type	Code	description
0	0	echo reply (ping)
3	0	dest network unreachable
3	1	dest host unreachable
3	2	dest protocol unreachable
3	3	dest port unreachable
3	6	dest network unknown
3	7	dest host unknown
4	0	source quench (congestion control - not used)
8	0	echo request (ping)
9	0	route advertisement
10	0	router discovery
11	0	TTL expired
12	0	bad IP header

ICMP Query message



- **ICMP query:**
  - Request sent by host to a router or host
  - Reply sent back to querying host

Example of ICMP Queries

Type/Code	Description
8/0	Echo Request
0/0	Echo Reply
13/0	Timestamp Request
14/0	Timestamp Reply
10/0	Router Solicitation
9/0	Router Advertisement

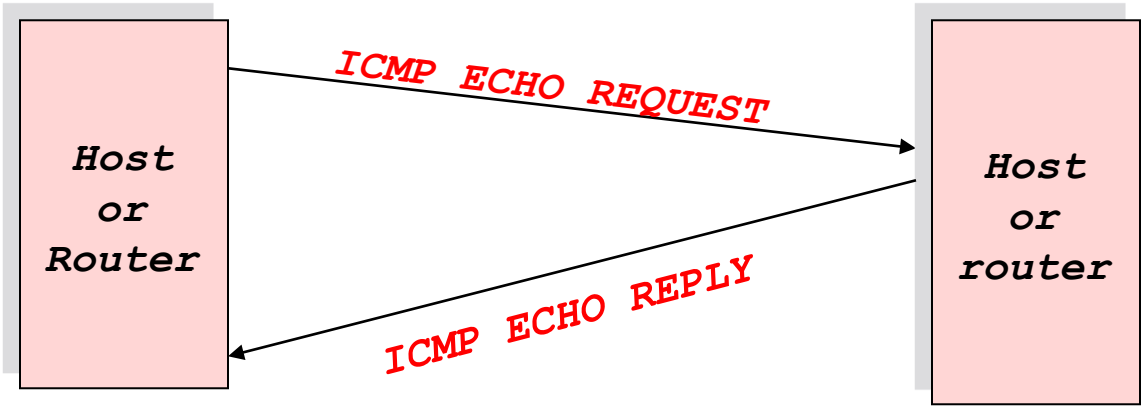
} The ping command uses Echo Request/ Echo Reply

Purpose of Req/Reply

- The ICMP echo request and echo reply messages are useful for network debugging.
  - If machine A sends an echo request message to machine B, machine B is required to respond with an ICMP echo reply.
  - Most systems supply an application program that sends and receives ICMP echo messages.
  - In UNIX, the program ping allows a user to check whether a machine is reachable and functioning.
  - Because ICMP messages are handled just like other IP datagrams, ICMP echo messages test the reachability of any host. Also, because ICMP is an integral part of IP, all hosts and gateways must implement ICMP.

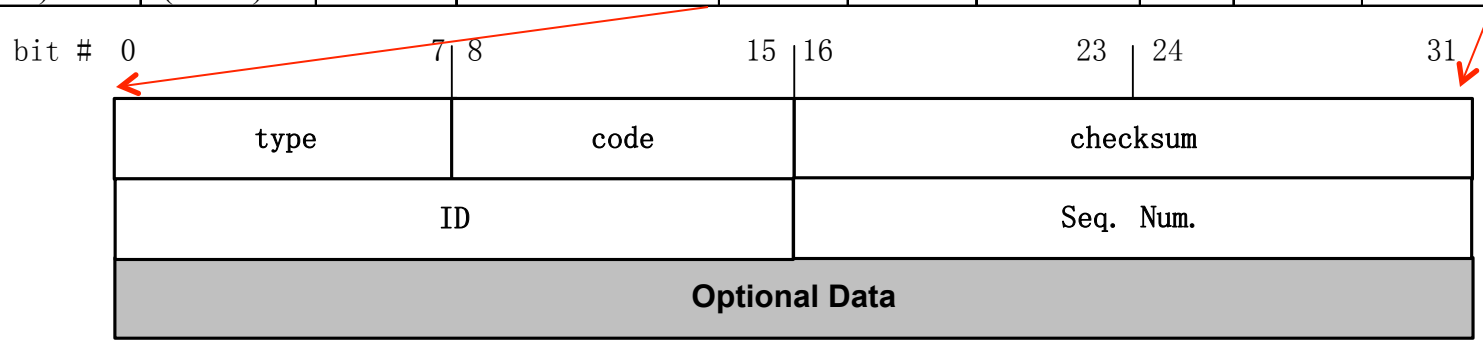
Ping: Echo Request and Reply

- Format
  - **ping** *ip address* (or **ping** <cr> for extended ping with CISCO IOS)
  - **ping** *172.30.1.25*
- Pings are handled directly by the kernel
  - Each Ping is translated into an ICMP Echo Request
  - The Pinged host responds with an ICMP Echo Reply



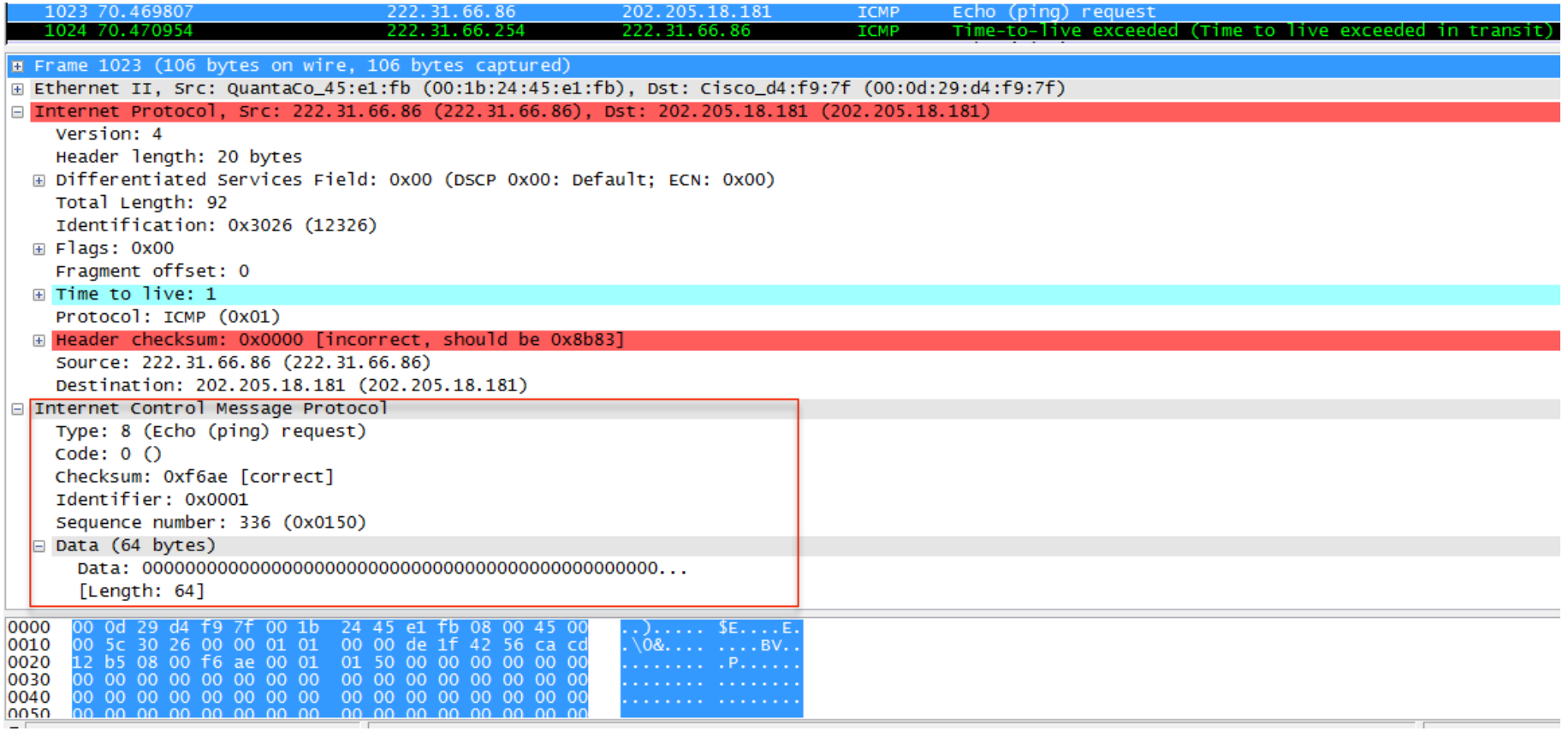
Ping Frame Format

Ethernet Header (Layer 2)			**IP Header (Layer 3)**	**ICMP Message (Layer 3)**						**Ether. Tr.**
Ethernet Destination Address (MAC)	Ethernet Source Address (MAC)	Frame Type	Source IP Add. Dest. IP Add. Protocol field:1	Type 8 or 0	Code 0	Check-sum	ID	Seq. Num.	Data	FCS

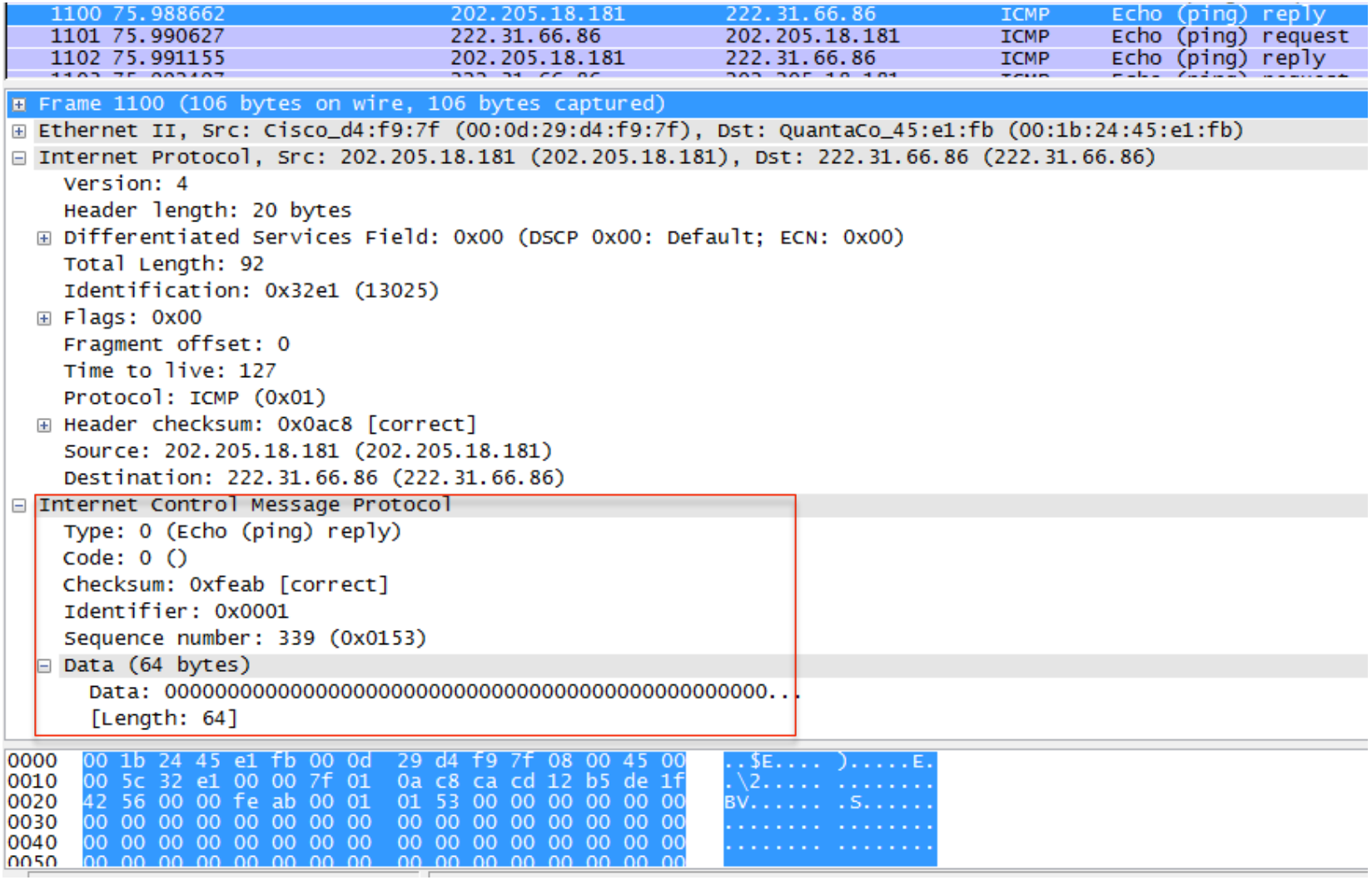


- Uses ICMP message within an IP Packet, Protocol field = 1
- Both are layer 3 protocols. (ICMP is considered as a network layer protocol.)
- Does not use TCP or UDP, but may be acted upon by the receiver using TCP or UDP.

ICMP Echo(Ping) Request Message



ICMP Echo(Ping) Reply Message



Ping Q&A

Q: Are pings forwarded by routers?

A: Yes! This is why you can ping devices all over the Internet.

Q: Do all devices forward or respond to pings?

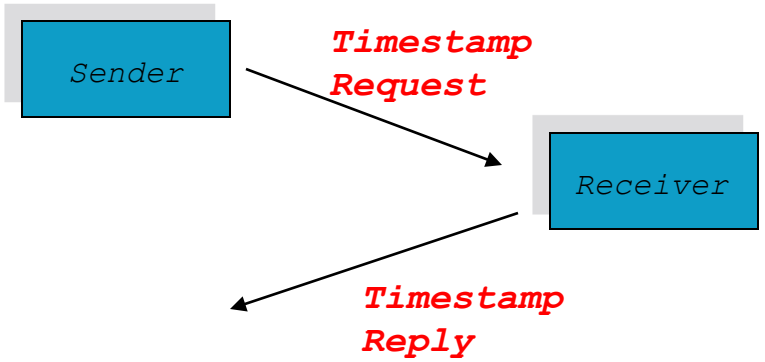
A: No, this is up to the network administrator of the device. Devices, including routers, can be configured not to reply to pings (ICMP echo requests). This is why you may not always be able to ping a device. Also, routers can be configured not to forward pings destined for other devices.

Timestamp Messages

- ICMP timestamp messages are used to estimate the transmission delays between machines and to synchronize clocks:
  - Including both the receive and transmit timestamp allows the sending host to determine the fraction of time spent transmitting vs. processing the request.
  - By averaging the measurements of several messages, the sender can estimate the offset between its local clock and that on the remote machine.
  - Note: it is quite feasible to synchronize the clocks of all machines on a LAN to within several milliseconds of each other.

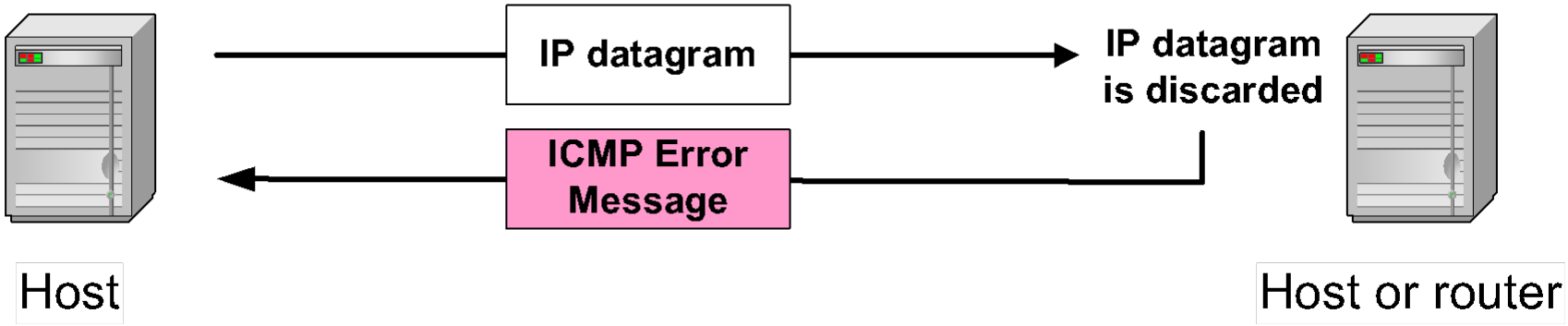
Example of a Query: ICMP Timestamp

- A system (host or router) asks another system for the current time.
- Time is measured in milliseconds after midnight UTC (Universal Coordinated Time) of the current day
- Sender sends a request, receiver responds with reply



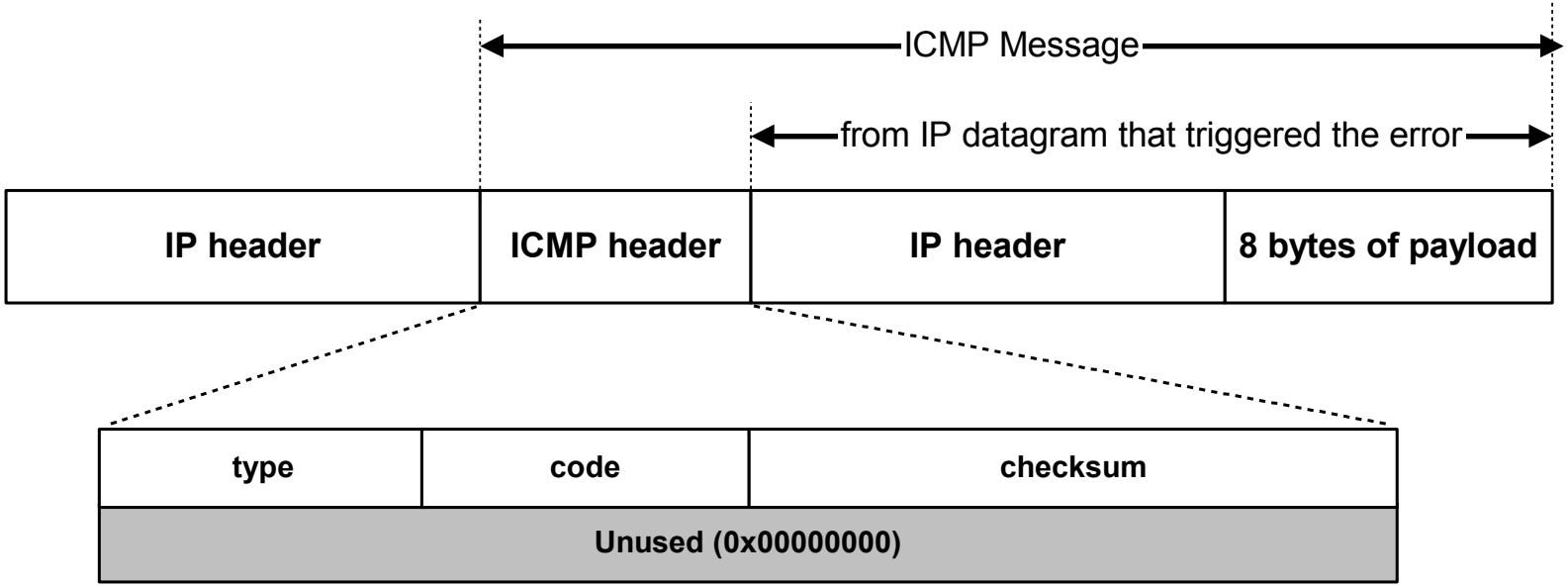
Type (= 17 or 18)	Code (=0)	Checksum
identifier		sequence number
32-bit sender timestamp		
32-bit receive timestamp		
32-bit transmit timestamp		

ICMP Error message



- **ICMP error messages report error conditions**
- **Typically sent when a datagram is discarded**
- **Error message is often passed from ICMP to the application program**

ICMP Error message



- ICMP error messages include the complete IP header and the first 8 bytes of the payload (typically: UDP, TCP)

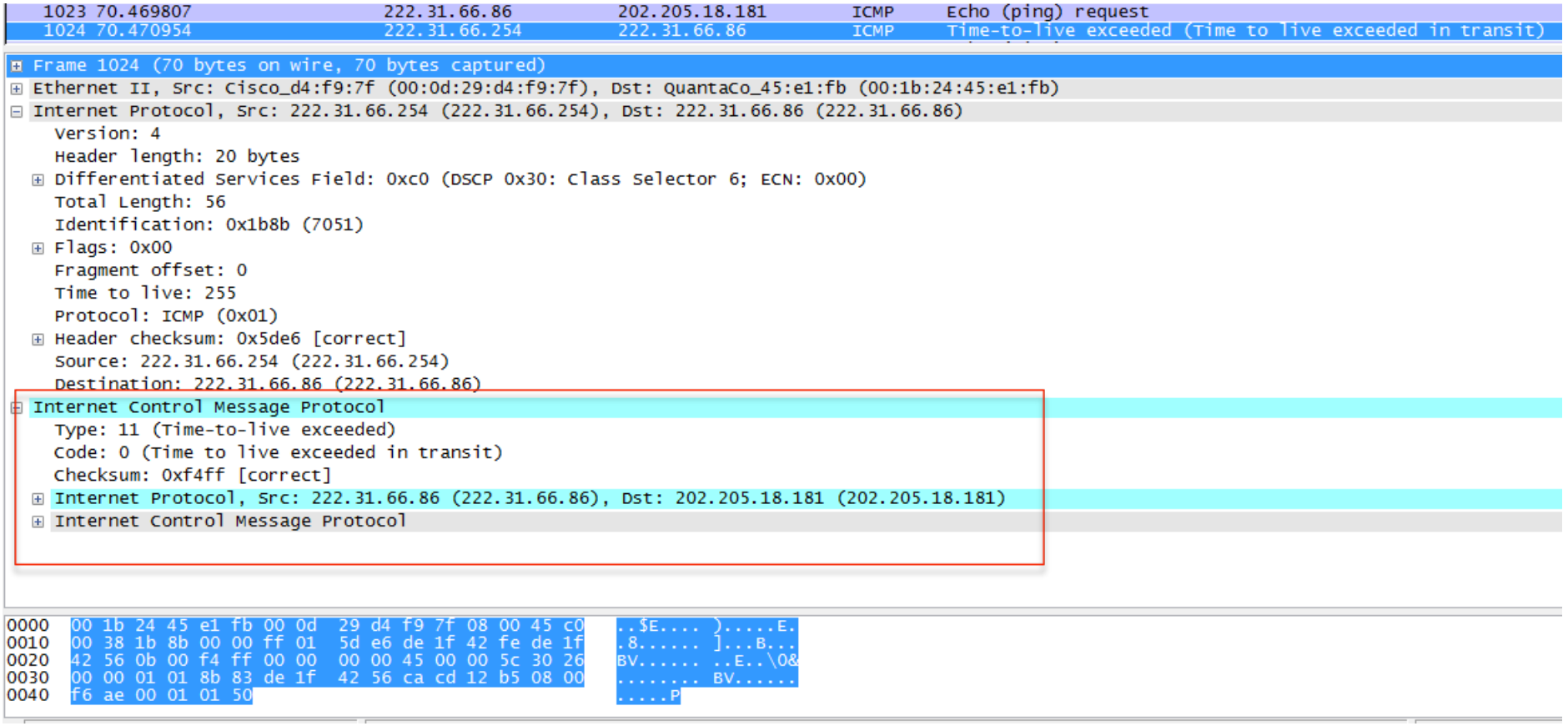
Frequent ICMP Error message

Type	Code	Description	
3	0–15	Destination unreachable	Notification that an IP datagram could not be forwarded and was dropped. The code field contains an explanation.
5	0–3	Redirect	Informs about an alternative route for the datagram and should result in a routing table update. The code field explains the reason for the route change.
11	0, 1	Time exceeded	Sent when the TTL field has reached zero (Code 0) or when there is a timeout for the reassembly of segments (Code 1)
12	0, 1	Parameter problem	Sent when the IP header is invalid (Code 0) or when an IP header option is missing (Code 1)

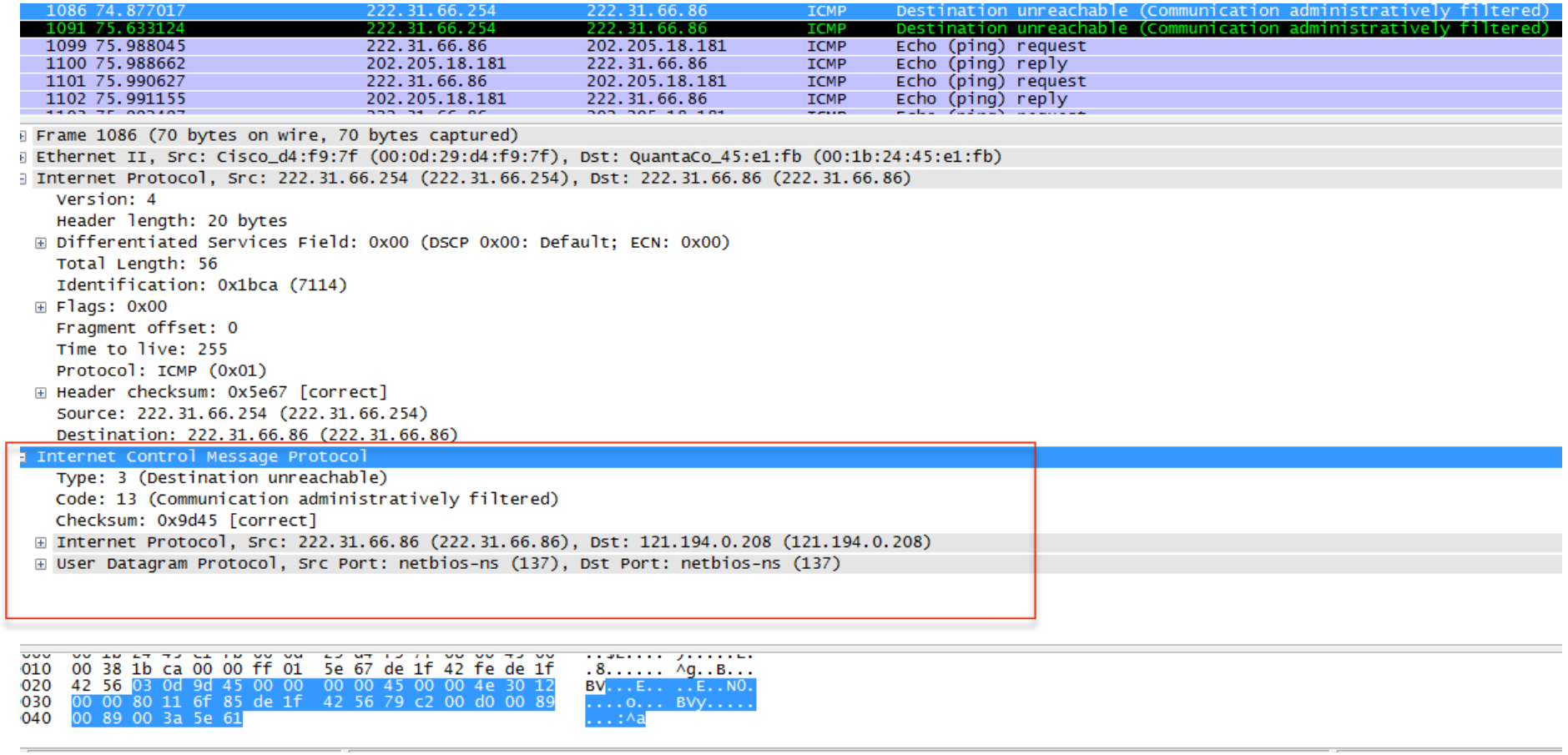
Some subtypes of the "Destination Unreachable"

Code	Description	Reason for Sending
0	Network Unreachable	No routing table entry is available for the destination network.
1	Host Unreachable	Destination host should be directly reachable, but does not respond to ARP Requests.
2	Protocol Unreachable	The protocol in the protocol field of the IP header is not supported at the destination.
3	Port Unreachable	The transport protocol at the destination host cannot pass the datagram to an application.
4	Fragmentation Needed and DF Bit Set	IP datagram must be fragmented, but the DF bit in the IP header is set.
13	Communication Administratively Prohibited	Generated if a router cannot forward a packet due to administrative filtering;

ICMP TTL Exceeded Message

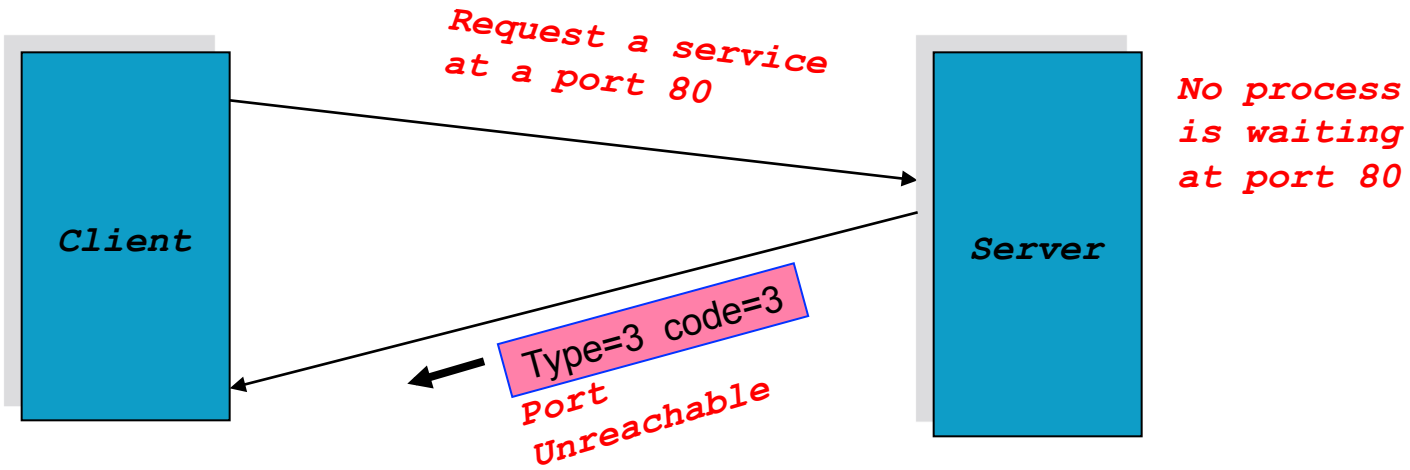


ICMP Dest Unreachable (Filtered)

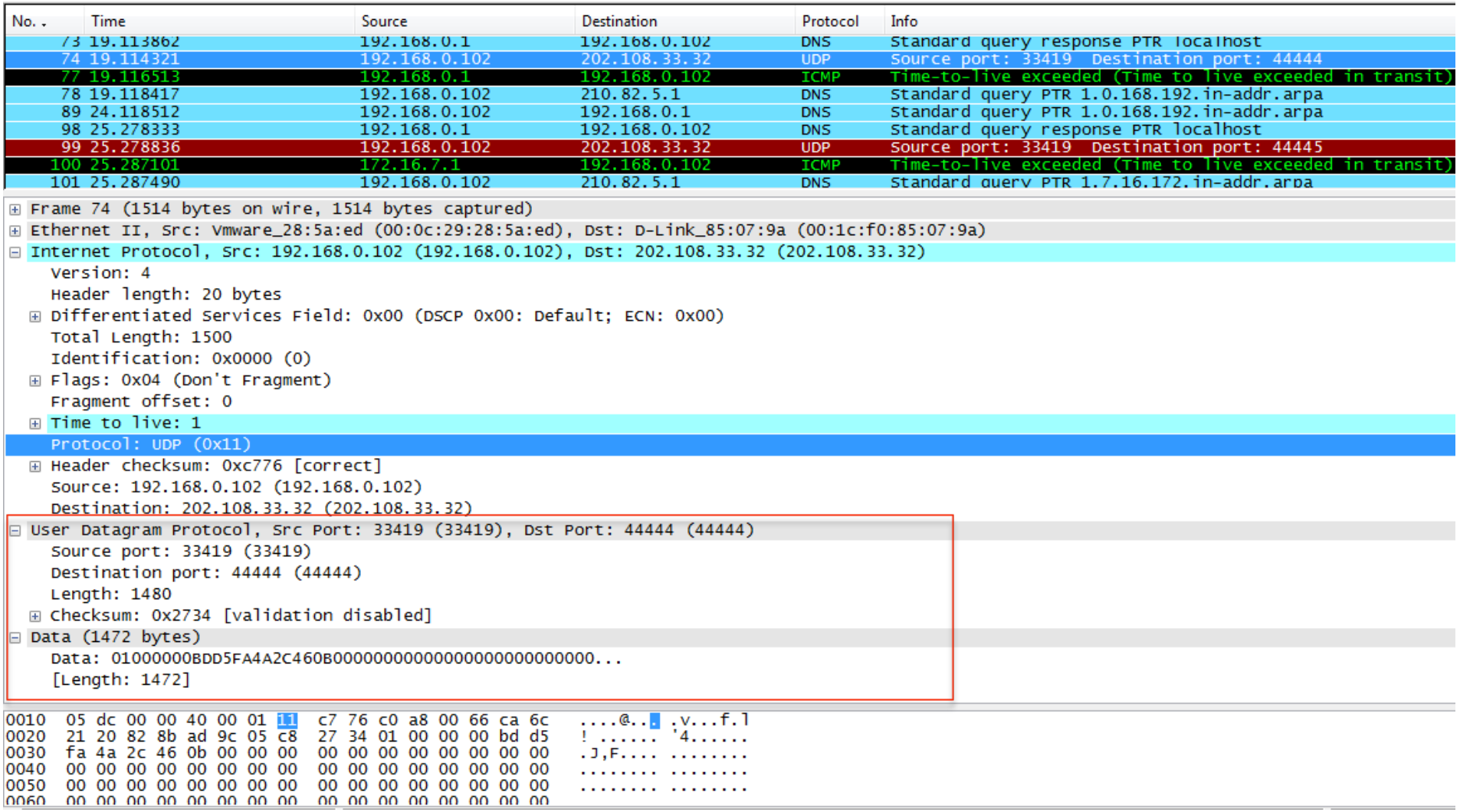


Example: ICMP Port Unreachable

- RFC 792:
  - If, in the destination host, the IP module cannot deliver the datagram because the indicated protocol module or process port is not active, the destination host may send a destination unreachable message to the source host.
- Scenario:



UDP Datagram on Linux

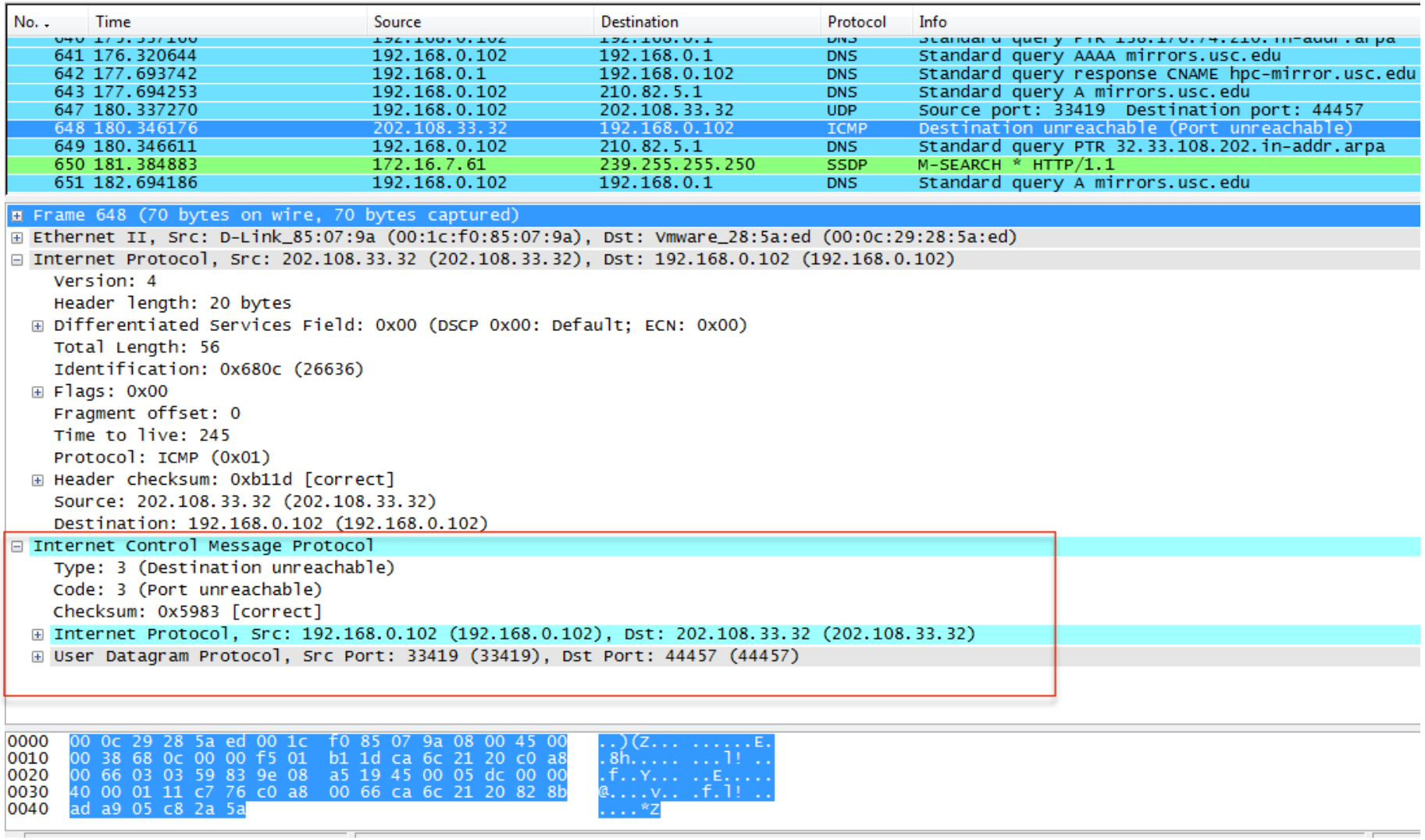


No.	Time	Source	Destination	Protocol	Info
73	19.113862	192.168.0.1	192.168.0.102	DNS	Standard query response PTR localhost
74	19.114321	192.168.0.102	202.108.33.32	UDP	Source port: 33419 Destination port: 44444
77	19.116513	192.168.0.1	192.168.0.102	ICMP	Time-to-live exceeded (Time to live exceeded in transit)
78	19.118417	192.168.0.102	210.82.5.1	DNS	Standard query PTR 1.0.168.192.in-addr.arpa
89	24.118512	192.168.0.102	192.168.0.1	DNS	Standard query PTR 1.0.168.192.in-addr.arpa
98	25.278333	192.168.0.1	192.168.0.102	DNS	Standard query response PTR localhost
99	25.278836	192.168.0.102	202.108.33.32	UDP	Source port: 33419 Destination port: 44445
100	25.287101	172.16.7.1	192.168.0.102	ICMP	Time-to-live exceeded (Time to live exceeded in transit)
101	25.287490	192.168.0.102	210.82.5.1	DNS	Standard query PTR 1.7.16.172.in-addr.arpa

```
⊞ Frame 74 (1514 bytes on wire, 1514 bytes captured)
⊞ Ethernet II, Src: Vmware_28:5a:ed (00:0c:29:28:5a:ed), Dst: D-Link_85:07:9a (00:1c:f0:85:07:9a)
⊟ Internet Protocol, Src: 192.168.0.102 (192.168.0.102), Dst: 202.108.33.32 (202.108.33.32)
    Version: 4
    Header length: 20 bytes
  ⊞ Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
    Total Length: 1500
    Identification: 0x0000 (0)
  ⊞ Flags: 0x04 (Don't Fragment)
    Fragment offset: 0
  ⊞ Time to live: 1
    Protocol: UDP (0x11)
  ⊞ Header checksum: 0xc776 [correct]
    Source: 192.168.0.102 (192.168.0.102)
    Destination: 202.108.33.32 (202.108.33.32)
⊟ User Datagram Protocol, Src Port: 33419 (33419), Dst Port: 44444 (44444)
    Source port: 33419 (33419)
    Destination port: 44444 (44444)
    Length: 1480
  ⊞ Checksum: 0x2734 [validation disabled]
⊟ Data (1472 bytes)
    Data: 01000000BDD5FA4A2C460B0000000000000000000000000000...
    [Length: 1472]
```

```
0010  05 dc 00 00 40 00 01 11  c7 76 c0 a8 00 66 ca 6c   ....@... .v...f.l
0020  21 20 82 8b ad 9c 05 c8  27 34 01 00 00 00 bd d5   ! ...... '4......
0030  fa 4a 2c 46 0b 00 00 00  00 00 00 00 00 00 00 00   .J,F.... ........
0040  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0050  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0060  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
```

ICMP Port Unreachable Error Message



No.	Time	Source	Destination	Protocol	Info
641	176.320644	192.168.0.102	192.168.0.1	DNS	Standard query AAAA mirrors.usc.edu
642	177.693742	192.168.0.1	192.168.0.102	DNS	Standard query response CNAME hpc-mirror.usc.edu
643	177.694253	192.168.0.102	210.82.5.1	DNS	Standard query A mirrors.usc.edu
647	180.337270	192.168.0.102	202.108.33.32	UDP	Source port: 33419 Destination port: 44457
648	180.346176	202.108.33.32	192.168.0.102	ICMP	Destination unreachable (Port unreachable)
649	180.346611	192.168.0.102	210.82.5.1	DNS	Standard query PTR 32.33.108.202.in-addr.arpa
650	181.384883	172.16.7.61	239.255.255.250	SSDP	M-SEARCH * HTTP/1.1
651	182.694186	192.168.0.102	192.168.0.1	DNS	Standard query A mirrors.usc.edu

```
Frame 648 (70 bytes on wire, 70 bytes captured)
Ethernet II, Src: D-Link_85:07:9a (00:1c:f0:85:07:9a), Dst: Vmware_28:5a:ed (00:0c:29:28:5a:ed)
Internet Protocol, Src: 202.108.33.32 (202.108.33.32), Dst: 192.168.0.102 (192.168.0.102)
    Version: 4
    Header length: 20 bytes
  Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
    Total Length: 56
    Identification: 0x680c (26636)
  Flags: 0x00
    Fragment offset: 0
    Time to live: 245
    Protocol: ICMP (0x01)
  Header checksum: 0xb11d [correct]
    Source: 202.108.33.32 (202.108.33.32)
    Destination: 192.168.0.102 (192.168.0.102)
Internet Control Message Protocol
    Type: 3 (Destination unreachable)
    Code: 3 (Port unreachable)
    Checksum: 0x5983 [correct]
  Internet Protocol, Src: 192.168.0.102 (192.168.0.102), Dst: 202.108.33.32 (202.108.33.32)
  User Datagram Protocol, Src Port: 33419 (33419), Dst Port: 44457 (44457)
```

```
0000  00 0c 29 28 5a ed 00 1c  f0 85 07 9a 08 00 45 00   ..)(Z... ......E.
0010  00 38 68 0c 00 00 f5 01  b1 1d ca 6c 21 20 c0 a8   .8h..... ...l! ..
0020  00 66 03 03 59 83 9e 08  a5 19 45 00 05 dc 00 00   .f..Y... ..E.....
0030  40 00 01 11 c7 76 c0 a8  00 66 ca 6c 21 20 82 8b   @....v.. .f.l! ..
0040  ad a9 05 c8 2a 5a                                 ....*Z
```

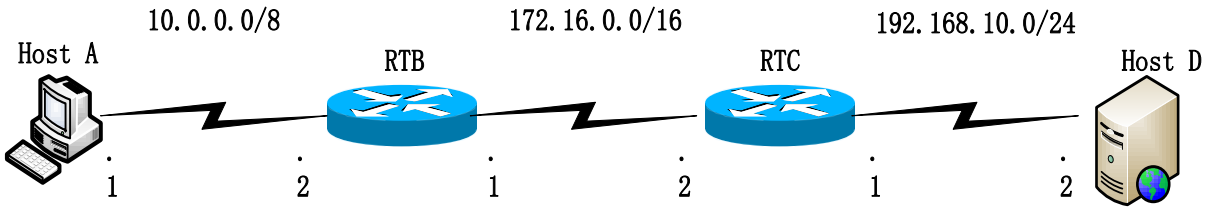
Trace*route*

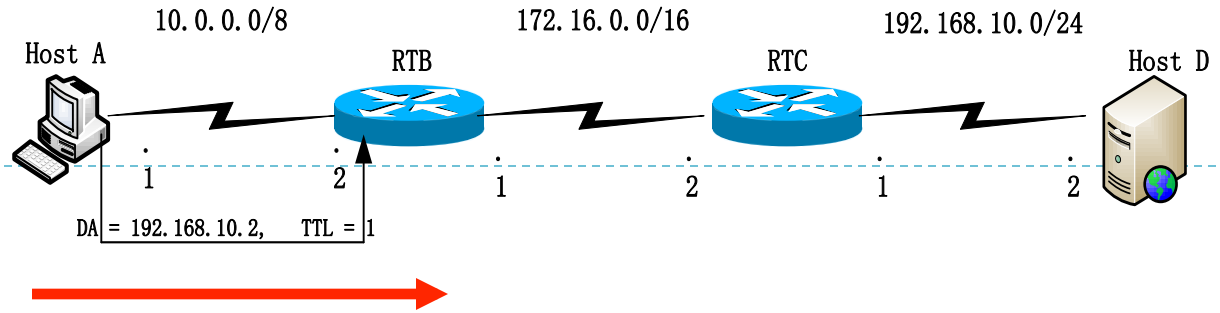
- Trace (traceroute, tracert, tracepath…) is used to trace the *probable* path a packet takes between source and destination.
- *Probable*, because IP is a connectionless protocol, and different packets may take different paths between the same source and destination networks, although this is not usually the case.
- Trace will show the path the packet takes to the destination, but the return path may be different.
  - This is more likely the case in the Internet, and less likely within your own autonomous system.
- Uses ICMP message within an IP Packet (on Windows)
- Uses UDP in the transport layer (on Unix/Linux/Cisco IOS).

Example(on Windows)

- HostA> Trace*RT* *ip_address*

```
HostA> TraceRT 192.168.10.2
```

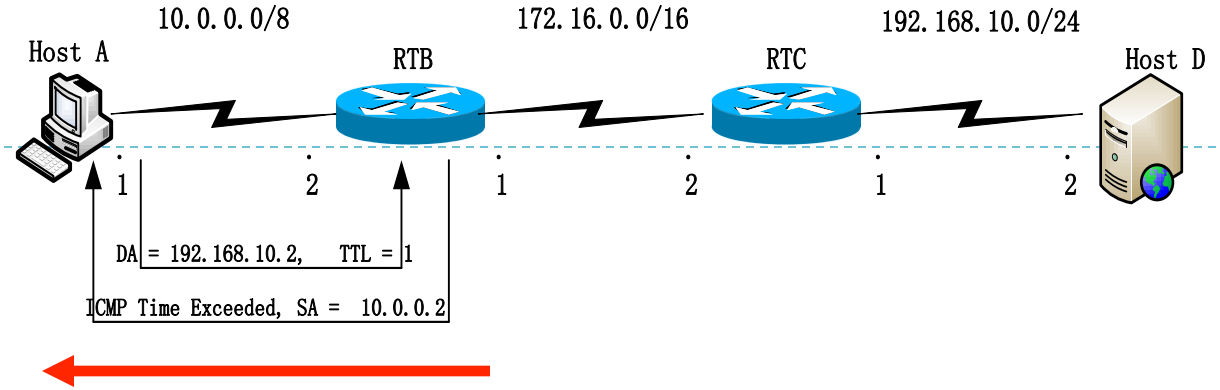




Data Link Header (Layer 2)			IP Header (Layer 3)	ICMP Message - Echo Request (trace)					DataLink Tr.
Data Link Destination Address	Data Link Source Address	……	**Source IP Add.** 10.0.0.1 **Dest. IP Add.** 192.168.10.2 **Protocol field** 1 **TTL** 1	Type 8 Code 0	Chk sum	ID	Seq. Num	Data	FCS

How it works - Fooling the routers & host!

- Traceroute uses ping (echo requests)
- Traceroute sets the TTL (Time To Live) field in the IP Header, initially to "1"



Data Link Header (Layer 2)			IP Header (Layer 3)	ICMP Message - Time Exceeded					DataLink Tr.
Data Link Destination Address	Data Link Source Address	….	**Source IP Add.** 10.0.0.2 **Dest. IP Add.** 10.0.0.1 **Protocol field** 1	**Type** 11 **Code** 0	Chk sum	0	0	Data	FCS

RTB - TTL:

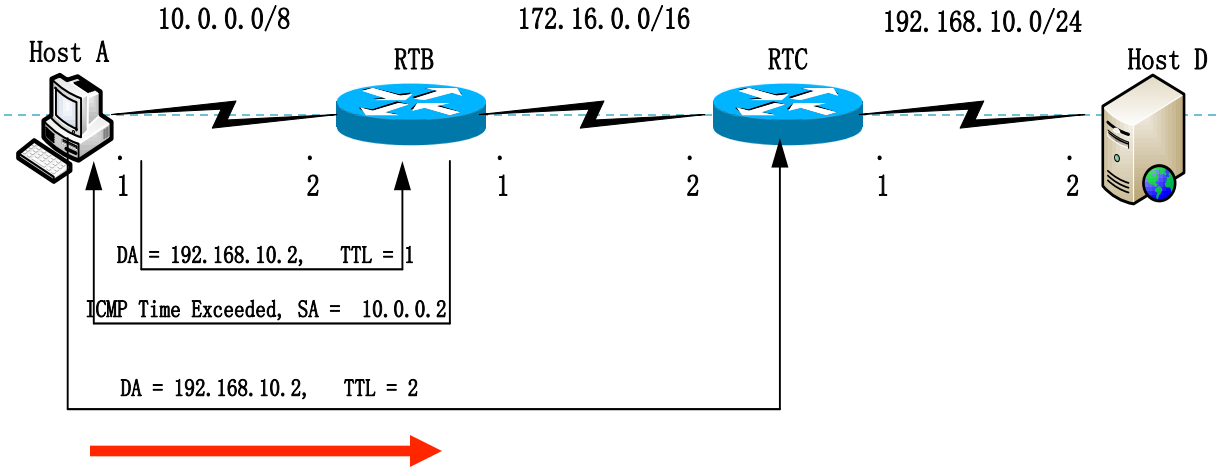
- When a router receives an IP Packet, it decrements the TTL by 1.
- If the TTL is 0, it will not forward the IP Packet, and send back to the source an ICMP "time exceeded" message.
  - using its IP header and first 8 bytes of ICMP header as Data
- ICMP Message: Type = 11, Code = 0

TraceRT output -1

HostA, Sending Host

- The traceroute program of the sending host (Host A) will use the source IP address of this ICMP Time Exceeded packet to display at the first hop.

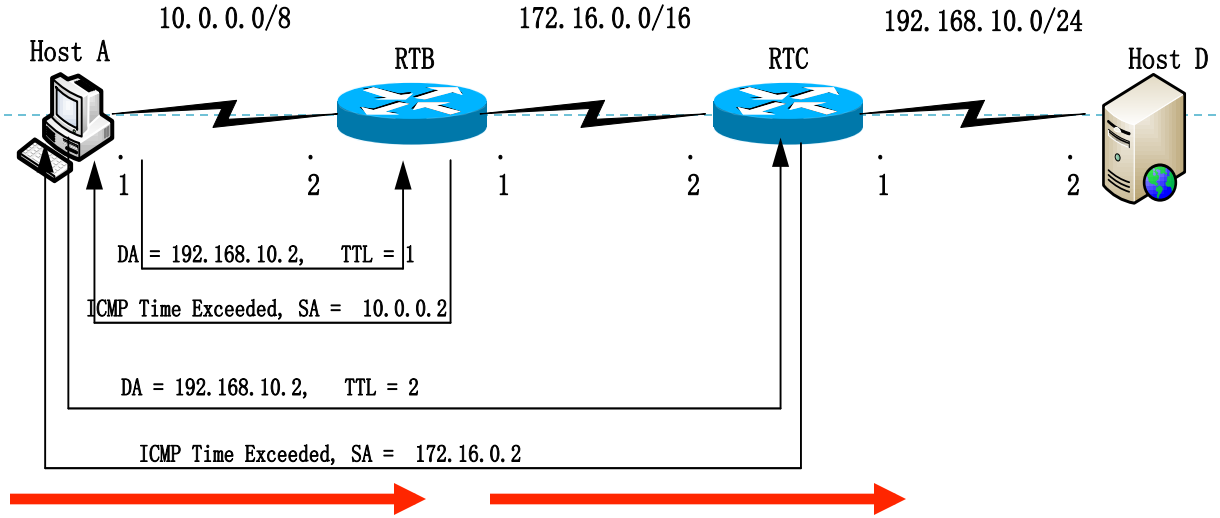
```
HostA> tracert 192.168.10.2
Type escape sequence to abort.
Tracing the route to 192.168.10.2
  1 10.0.0.2 4 msec 4 msec 4 msec
```



Data Link Header (Layer 2)			**IP Header (Layer 3)**	**ICMP Message - Echo Request (trace)**					**DataLink Tr.**
Data Link Destination Address	Data Link Source Address	……	**Source IP Add.** 10.0.0.1 **Dest. IP Add.** 192.168.10.2 **Protocol field** 1 **TTL** 2	Type 8 Code 0	Chk sum	ID	Seq. Num	Data	FCS

HostA

- The traceroute program increments the TTL by 1 (now 2) and resends the ICMP Echo Request packet.

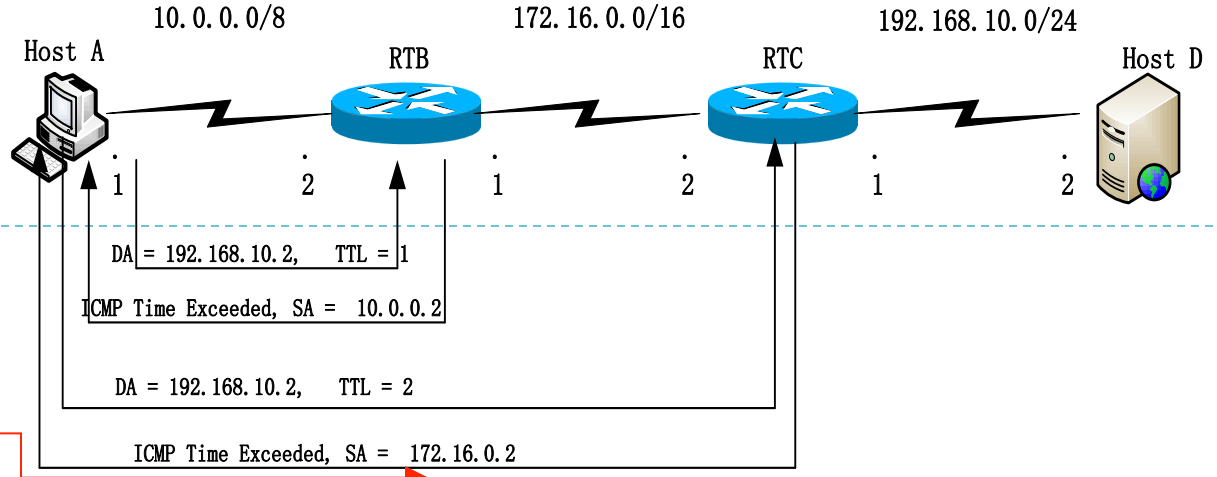


RTB

- This time RTB decrements the TTL by 1 and it is NOT 0. (It is 1.)
- So it looks up the destination ip address in its routing table and forwards it on to the next router.

RTC

- RTC however decrements the TTL by 1 and it is 0.
- RTC notices the TTL is 0 and sends back the ICMP Time Exceeded message back to the source.
- RTC's IP header includes its own IP address (source IP) and the sending host's IP address (destination IP address of RTA).
- The sending host, RTA, will use the source IP address of this ICMP Time Exceeded message to display at the second hop.



HostA to RTB

Data Link Header (Layer 2)			**IP Header (Layer 3)**	**ICMP Message - Echo Request (trace)**					**DataLink Tr.**
Data Link Destination Address	Data Link Source Address	……	**Source IP Add.** 10.0.0.1 **Dest. IP Add.** 192.168.10.2 **Protocol field** 1 **TTL** 2	Type 8 Code 0	Chk sum	ID	Seq. Num	Data	FCS

RTB to RTC

Data Link Header (Layer 2)			**IP Header (Layer 3)**	**ICMP Message - Echo Request (trace)**					**DataLink Tr.**
Data Link Destination Address	Data Link Source Address	……	**Source IP Add.** 10.0.0.1 **Dest. IP Add.** 192.168.10.2 **Protocol field** 1 **TTL** 1	Type 8 Code 0	Chk sum	ID	Seq. Num	Data	FCS

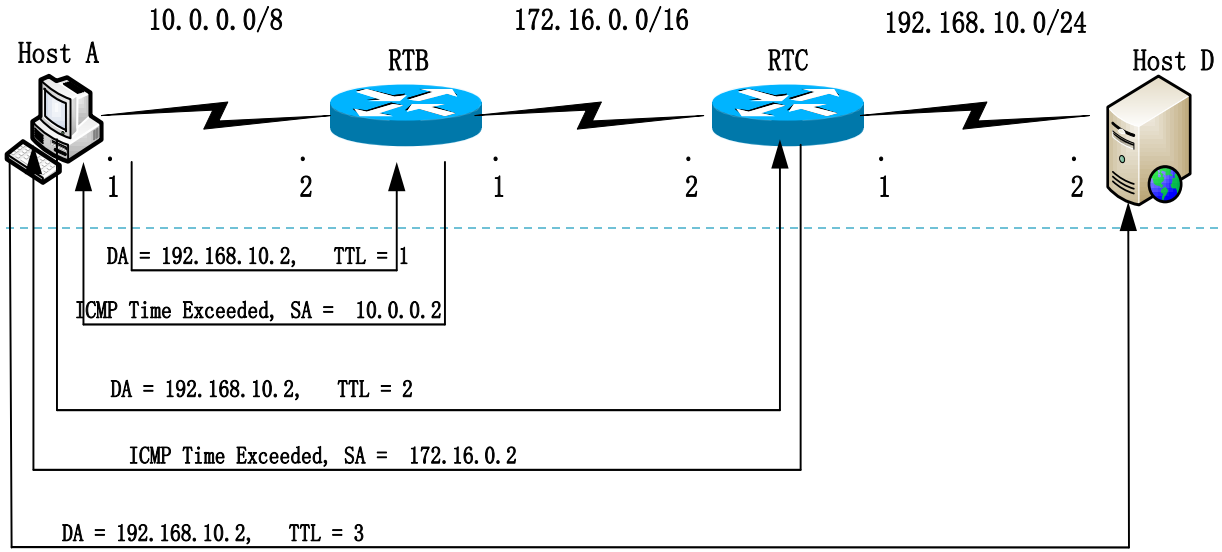
Data Link Header (Layer 2)			**IP Header (Layer 3)**	**ICMP Message - Time Exceeded**					**DataLink Tr.**
Data Link Destination Address	Data Link Source Address	….	**Source IP Add.** 172.16.0.2 **Dest. IP Add.** 10.0.0.1 **Protocol field** 1	**Type** 11 **Code** 0	Chk sum	0	0	Data	FCS

TraceRT output -2

The sending host, Host A:

- The traceroute program uses this information (Source IP Address) and displays the second hop.

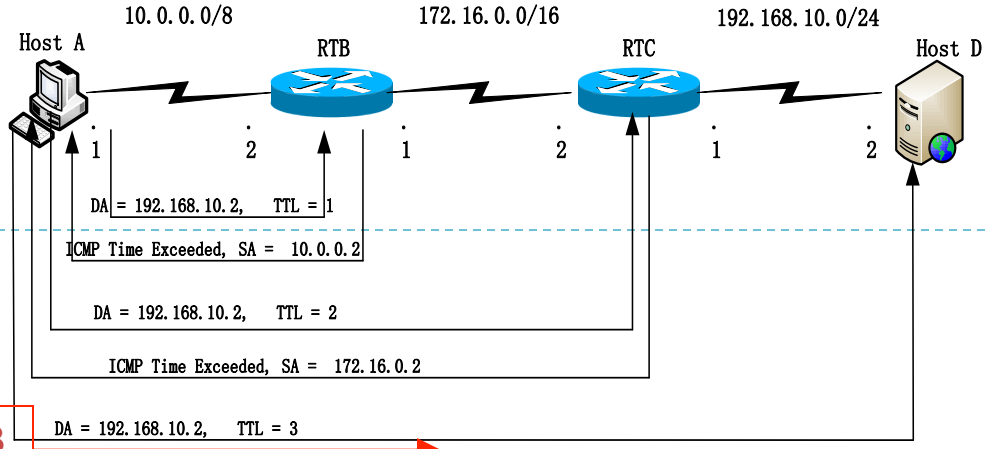
```
HostA> tracert 192.168.10.2
Type escape sequence to abort.
Tracing the route to 192.168.10.2

  1 10.0.0.2 4 msec 4 msec 4 msec
  2 172.16.0.2 20 msec 16 msec 16 msec
```



Data Link Header (Layer 2)			**IP Header (Layer 3)**	**ICMP Message - Echo Request (trace)**					**DataLink Tr.**
Data Link Destination Address	Data Link Source Address	……	**Source IP Add.** 10.0.0.1 **Dest. IP Add.** 192.168.10.2 **Protocol field** 1 **TTL** 3	Type 8 Code 0	Chk sum	ID	Seq. Num	Data	FCS

The sending host, HostA:

- The traceroute program increments the TTL by 1 (now 3) and resends the Packet.



HostA to RTB

Data Link Header (Layer 2)			**IP Header (Layer 3)**	**ICMP Message - Echo Request (trace)**					**DataLink Tr.**
Data Link Destination Address	Data Link Source Address	……	**Source IP Add.** 10.0.0.1 **Dest. IP Add.** 192.168.10.2 **Protocol field** 1 **TTL** 3	Type 8 Code 0	Chk sum	ID	Seq. Num	Data	FCS

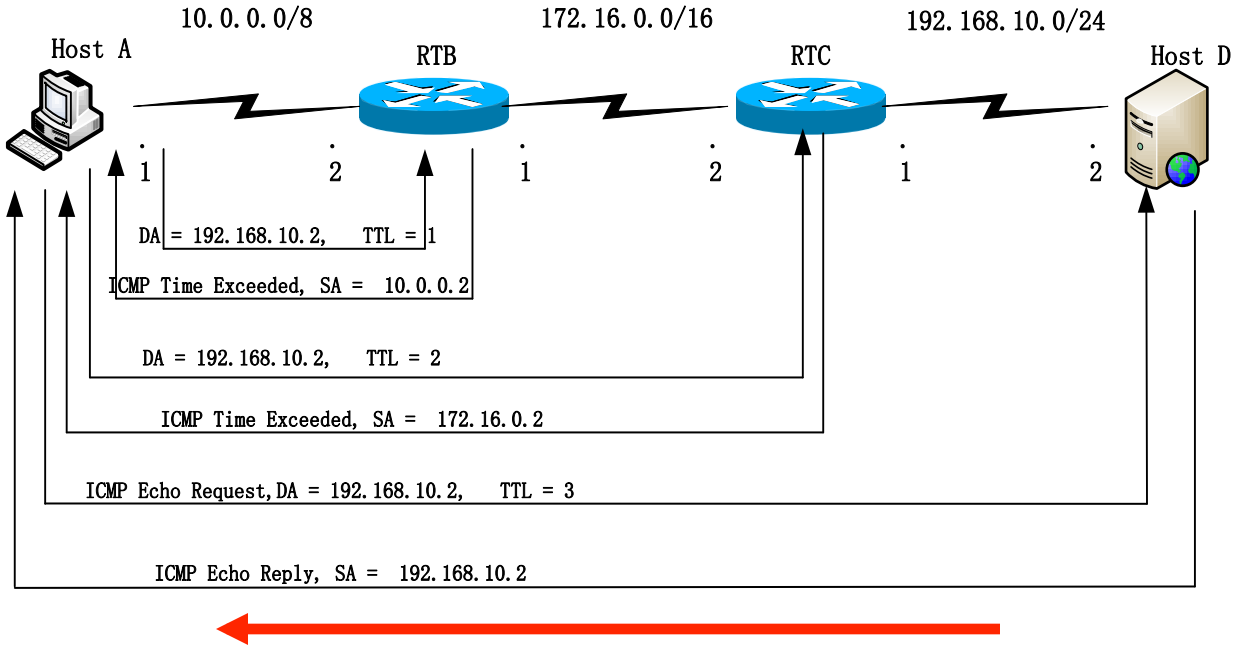
RTB to RTC

Data Link Header (Layer 2)			**IP Header (Layer 3)**	**ICMP Message - Echo Request (trace)**					**DataLink Tr.**
Data Link Destination Address	Data Link Source Address	……	**Source IP Add.** 10.0.0.1 **Dest. IP Add.** 192.168.10.2 **Protocol field** 1 **TTL** 2	Type 8 Code 0	Chk sum	ID	Seq. Num	Data	FCS

RTC to HostD

Data Link Header (Layer 2)			**IP Header (Layer 3)**	**ICMP Message - Echo Request (trace)**					**DataLink Tr.**
Data Link Destination Address	Data Link Source Address	……	**Source IP Add.** 10.0.0.1 **Dest. IP Add.** 192.168.10.2 **Protocol field** 1 **TTL** 1	Type 8 Code 0	Chk sum	ID	Seq. Num	Data	FCS

HostD → HostA



Data Link Header (Layer 2)			**IP Header (Layer 3)**	**ICMP Message – Echo Reply**					**DataLink Tr.**
Data Link Destination Address	Data Link Source Address	….	**Source IP Add.** 192.168.10.2 **Dest. IP Add.** 10.0.0.1 **Protocol field** 1	**Type** 0 **Code** 0	Chk sum	ID	Seq. Num	Data	FCS

Tracing to HostD

RTB

- This time RTB decrements the TTL by 1 and it is NOT 0. (It is 2.)
- So it looks up the destination ip address in its routing table and forwards it on to the next router.

RTC

- This time RTC decrements the TTL by 1 and it is NOT 0. (It is 1.)
- So it looks up the destination ip address in its routing table and forwards it on to the next router.

HostD

- HostD however decrements the TTL by 1 and it is 0.
- However, HostD notices that the Destination IP Address of 192.168.0.2 is it's own interface.
- Since it does not need to forward the packet, the TTL of 0 has no affect.
- HostD sends the ICMP Echo Reply message to HostA.

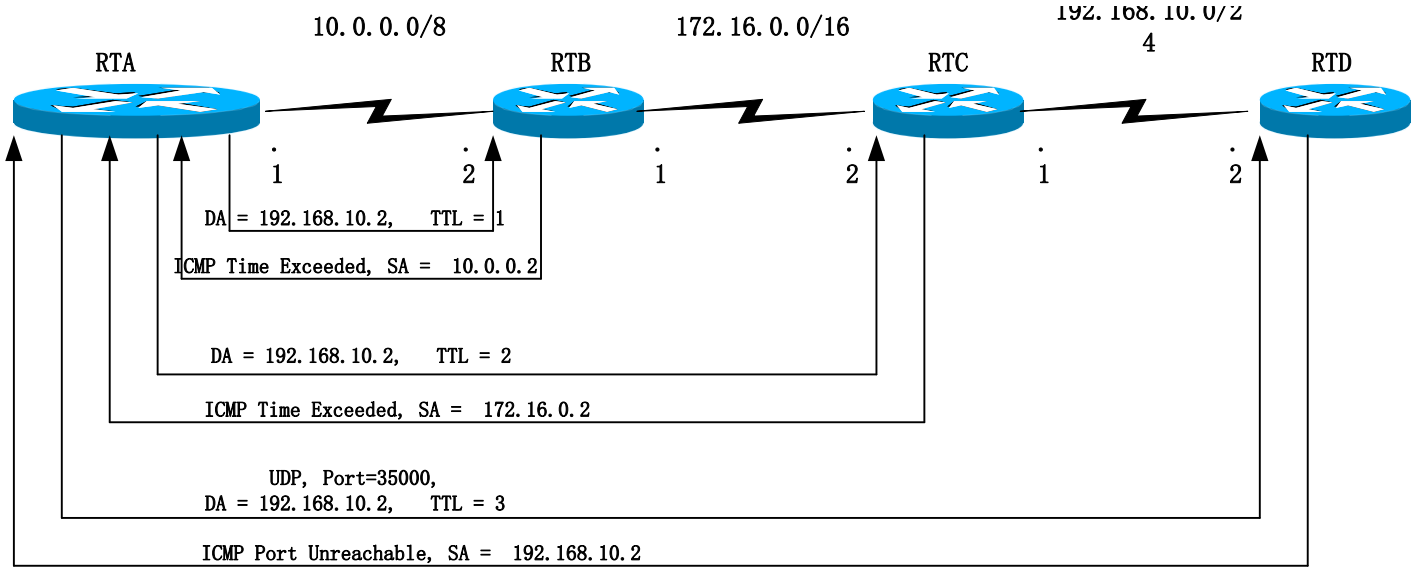
TraceRT output -3

Sending host, HostA

- HostA receives the ICMP Echo Reply message.
- The traceroute program uses this information (Source IP Address) and displays the third hop.
- The traceroute program also recognizes this **ICMP Echo Reply** as meaning this is the destination it was tracing (it knows this is the final hop and does not send any more echo requests).
- HostA, the sending host, now displays the third hop.

```
HostA> tracert 192.168.10.2
Type escape sequence to abort.
Tracing the route to 192.168.10.2

  1 10.0.0.2 4 msec 4 msec 4 msec
  2 172.16.0.2 20 msec 16 msec 16 msec
  3 192.168.10.2 16 msec 16 msec 16 msec
```

UDP tracing on Linux/Cisco IOS



ICMP Port Unreachable on RTD

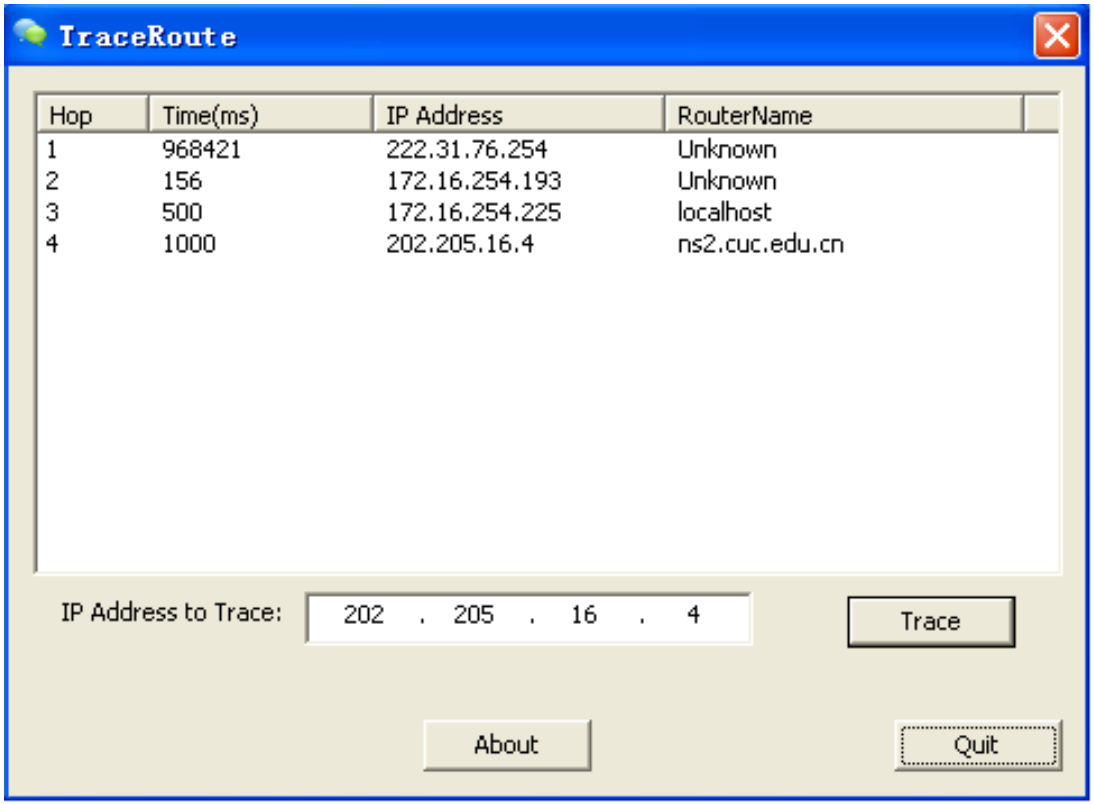
Data Link Header (Layer 2)			**IP Header (Layer 3)**	**UDP (Layer 4)**	**DataLink Tr.**
Data Link Destination Address	Data Link Source Address	……	**Source IP Add.** 10.0.0.1 **Dest. IP Add.** 192.168.10.2 **Protocol field** 0x11 **TTL** 1	**DestPort** 35,000	FCS

RTD

Data Link Header (Layer 2)			**IP Header (Layer 3)**	**ICMP Message – Port Unreachable**					**DataLink Tr.**
Data Link Destination Address	Data Link Source Address	….	**Source IP Add.** 192.168.10.2 **Dest. IP Add.** 10.0.0.1 **Protocol field** 1	**Type** 3 **Code** 3	Chk sum	0	0	Data	FCS

- RTD sends the packet to the UDP process.
- UDP examines the unrecognizable port number of 35,000 and sends back an **ICMP Port Unreachable message** to the sender, RTA, using Type 3 and Code 3.

TraceRoute


TraceRoute
Hop
Time(ms)
IP Address
RouterName
1
968421
222.31.76.254
Unknown
2
156
172.16.254.193
Unknown
3
500
172.16.254.225
localhost
4
1000
202.205.16.4
ns2.cuc.edu.cn
IP Address to Trace:
202 . 205 . 16 . 4
Trace
About
Quit

References

- RFC 792 - Internet Control Message Protocol
- RFC1393 - Traceroute Using an IP Option
- Internet Control Message Protocol (ICMP) Parameters: http://www.iana.org/assignments/icmp-parameters/icmp-parameters.xml
- RFC1574 - Essential Tools for the OSI Internet
- http://en.wikipedia.org/wiki/Traceroute
- http://www.cs.virginia.edu/~itlab/book/
- CISCO: Understanding the Ping and Traceroute Commands: http://www.cisco.com/en/US/products/sw/iosswrel/ps1831/products_tech_note09186a00800a6057.shtml