# CS 305 Lab Tutorial
# Lab11 IP & ICMP

Dept. Computer Science and Engineering
Southern University of Science and Technology

SUSTech
Southern University
of Science and Technology

Thanks to Wei WANG

# Topic

- IPv4
  - Best effort, IP address, IP fragment and assemble
- ICMP
  - Detect and report
- IPv6
  - The difference between IPv4 and IPv6

# Part A. IPv4

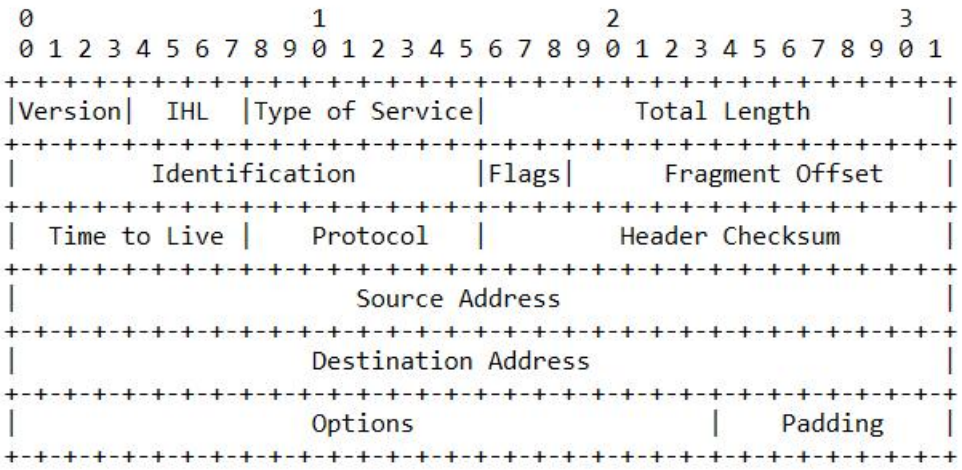
- **Best effort** : NO connection, NO flow control, NO congestion control, NO retransmission...

- The internet protocol implements two basic functions: **addressing** and **fragmentation.**
  - The internet modules use the addresses carried in the internet header to transmit internet datagrams toward their destinations. The selection of a path for transmission is called **routing**.
  - The internet modules use fields in the internet header to **fragment** and **reassemble** internet datagrams when necessary for transmission through "small packet" networks. The model of operation is that an internet module resides in each host engaged in internet communication and in each gateway that interconnects networks.



https://tools.ietf.org/html/rfc791

# IPv4 Datagram




Example Internet Datagram Header

- **Type of Service:**

The major choice is a three way tradeoff between low-delay, high-reliability, and high-throughput.

- **Time to Live (TTL):**

an indication of an upper bound on the lifetime of an internet datagram. **It is set by the sender of the datagram and reduced at the points along the route where it is processed.** An IP datagram with zero TTL will be dropped.

- **Header Checksum:**

provides a verification that the information used in processing internet datagram has been transmitted correctly. The data may contain errors. If the header checksum fails, the internet datagram is discarded at once by the entity which detects the error.

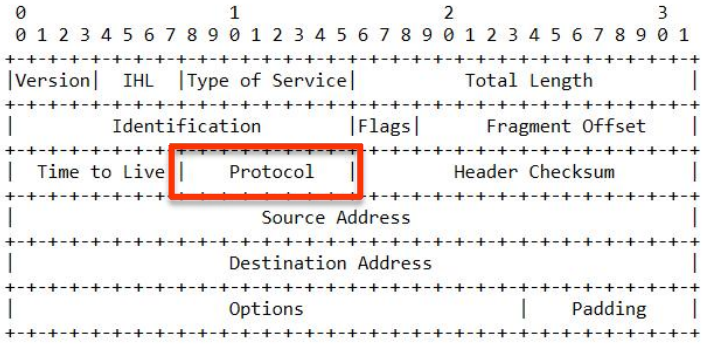
- **Options:**

provide for control functions needed or useful in some situations but unnecessary for the most common communications. The options include provisions for timestamps, security, and special routing.



https://tools.ietf.org/html/rfc791

# “Protocol” Field




Example Internet Datagram Header

```
Internet Protocol Version 4, Src: 192.168.2.104 (192.168.2.104), Dst
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 1020
    Identification: 0x0a9a (2714)
  > Flags: 0x00b9
    Time to live: 6
    Protocol: ICMP (1)
    Header checksum: 0x8493 [validation
    [Header checksum status: Unverified
    Source: 192.168.2.104 (192.168.2.10
    Destination: 116.7.234.3 (116.7.234
  > [2 IPv4 Fragments (2480 bytes): #1(
Internet Control Message Protocol
```

```
Internet Protocol Version 4, Src: 192.168.2.104 (192.168.2.104), Dst: t
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 52
    Identification: 0x05ec (1516)
  > Flags: 0x4000, Don't fragment
    Time to live: 64
    Protocol: TCP (6)
    Header checksum: 0x0fda [vali
    [Header checksum status: Unve
    Source: 192.168.2.104 (192.16
    Destination: tg-in-f113.1e100
> Transmission Control Protocol,
```

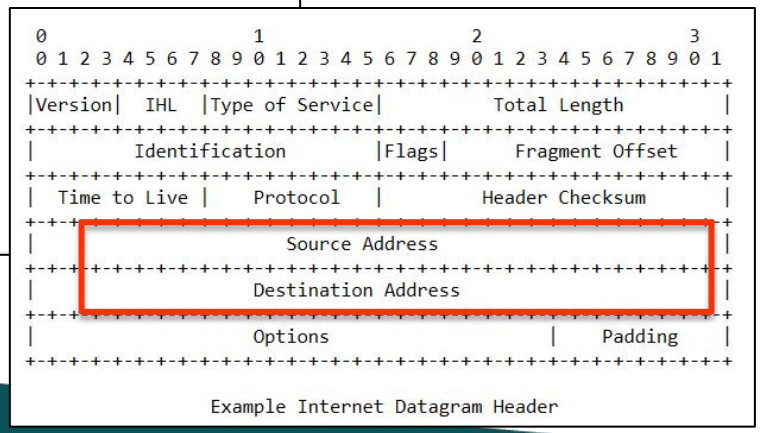
```
Internet Protocol Version 4, Src: tw.net-east.com (116.77.76.254), Dst
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 128
    Identification: 0x311d (12573)
  > Flags: 0x0000
    Time to live: 57
    Protocol: UDP (17)
    Header checksum: 0xcbf4 [validation disabled]
    [Header checksum status: Unverified]
    Source: tw.net-east.com (116.77.76.254)
    Destination: 192.168.2.104 (192.168.2.104)
> User Datagram Protocol, Src Port: domain (53), Dst Port: 61818 (61818)
> Domain Name System (response)
```

SUSTech
Southern University
of Science and Technology

# “Source” and “Destination” Field

```
> Frame 4: 216 bytes on wire (1728 bits), 216 bytes captured (1728 bits) on interface 0
> Ethernet II, Src: IntelCor_5c:69:58 (90:61:ae:5c:69:58), Dst: IPv4mcast_7f:ff:fa (01:00:5e:7f:ff:fa)
v Internet Protocol Version 4, Src: 192.168.2.104 (192.168.2.104), Dst: 239.255.255.250 (239.255.255.250)
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 202
    Identification: 0x7437 (29751)
  > Flags: 0x0000
    Time to live: 1
    Protocol: UDP (17)
    Header checksum: 0x91e1 [validation disabled]
    [Header checksum status: Unverified]
    Source: 192.168.2.104 (192.168.2.104)
    Destination: 239.255.255.250 (239.255.255.250)
> User Datagram Protocol, Src Port: 58806 (58806), Dst Port: ssdp (1900)
> Simple Service Discovery Protocol
```

```
v Internet Protocol Version 4, Src: 0.0.0.0 (0.0.0.0), Dst: 255.255.255.255 (255.255.255.255)
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 328
    Identification: 0xb310 (45840)
  > Flags: 0x0000
    Time to live: 128
    Protocol: UDP (17)
    Header checksum: 0x8695 [validation disabled]
    [Header checksum status: Unverified]
    Source: 0.0.0.0 (0.0.0.0)
    Destination: 255.255.255.255 (255.255.255.255)
```

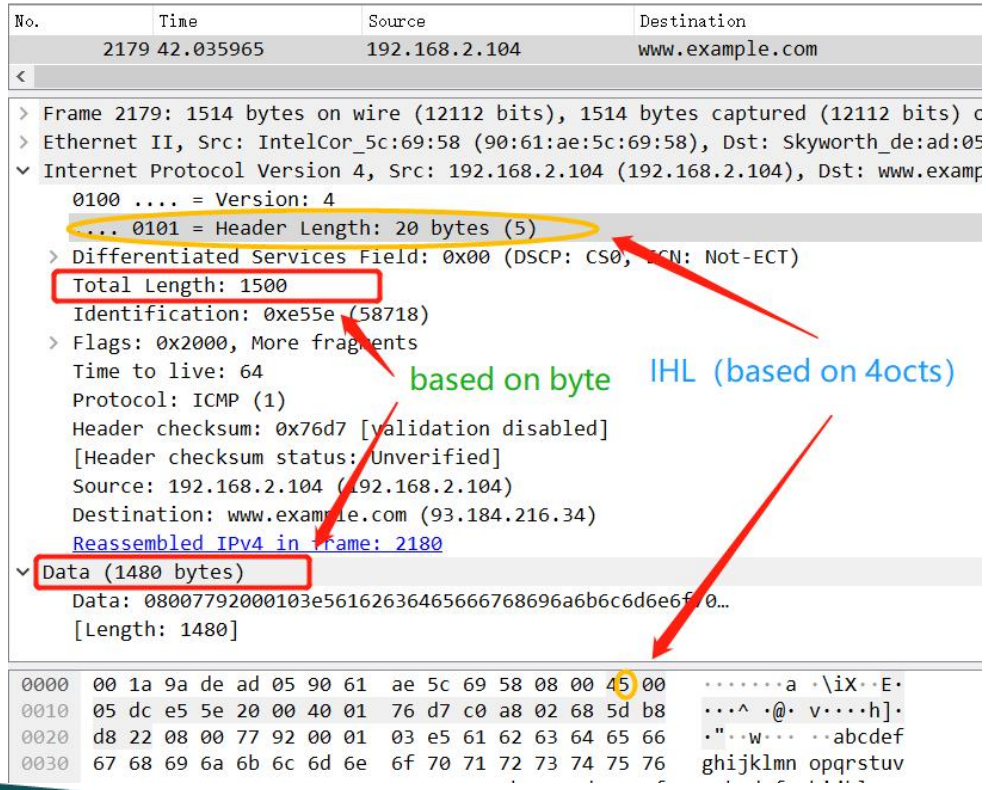


Example Internet Datagram Header

SUSTech
Southern University of Science and Technology

# “IHL” and “Total Length” Field

Initial the session with following cmd command: ping www.example.com –l 2000

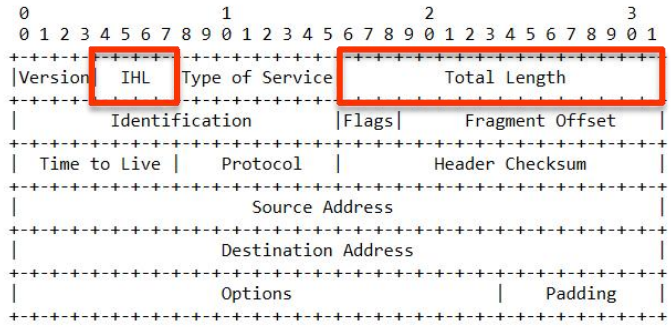



**IHL**: 4 bits

**I**nternet **H**eader **L**ength is the length of the internet header in 32 bit words, and thus points to the beginning of the data. Note that the minimum value for a correct header is 5.

**Total Length**: 16 bits

the length of the datagram, measured in octets, including internet header and data.

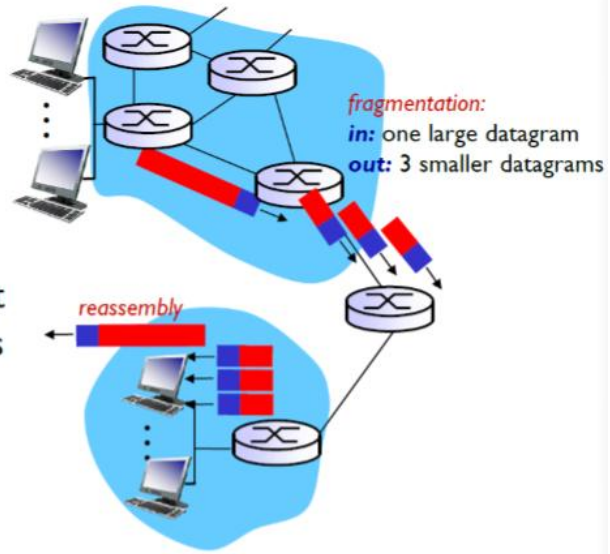



SUSTech
Southern University of Science and Technology

# IP Fragmentation and Reassembly(1)

## IP fragmentation, reassembly

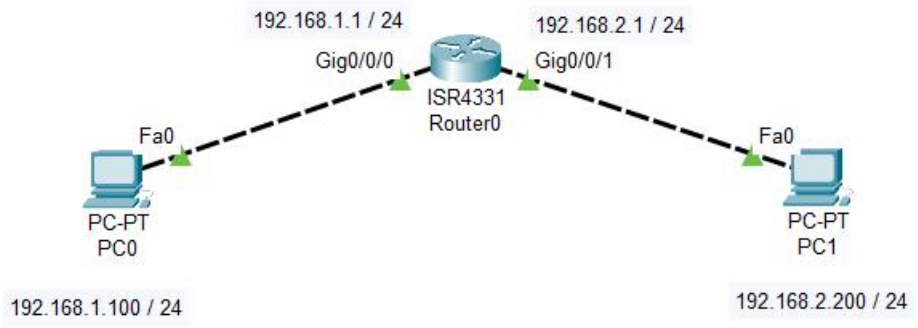
- network links have MTU (max.transfer size) - largest possible link-level frame
  - different link types, different MTUs
- large IP datagram divided ("fragmented") within net
  - one datagram becomes several datagrams
  - "reassembled" only at final destination
  - IP header bits used to identify, order related fragments



# IP Fragmentation and Reassembly(2)

Set the **MTU** of the interface **Gig0/0/0** in Router0 as 70, keep the MUT of the interface Gig0/0/1 in Router0 as 1500.

Q:While PC0 send a packet to PC1, and then PC1 reply a packet to PC0, **which device** would do the **ip fragmentation** and which device would do the **ip reassembly**?

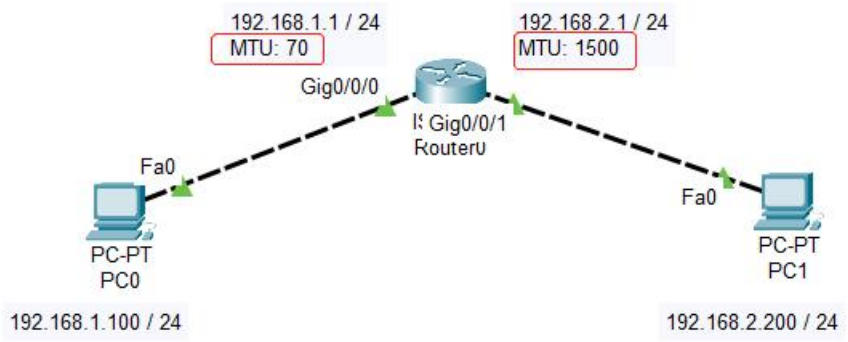



```
Router(config)#interface gigabitEthernet 0/0/0
Router(config-if)#mtu 70
%IPV6-3-MIN_MTU: The link MTU of GigabitEthernet0/0/0 is below the 1280-byte
minimum IPv6 link MTU. IPv6 may not work correctly on this interface.
Router(config-if)#
```

```
Router#show interfaces gigabitEthernet 0/0/0
GigabitEthernet0/0/0 is up, line protocol is up (connected)
  Hardware is ISR4331-3x1GE, address is 00d0.ffbb.db01 (bia 00d0.ffbb.db01)
  Internet address is 192.168.1.200/24
  MTU 70 bytes, BW 1000000 Kbit, DLY 100 usec,
     reliability 255/255, txload 1/255, rxload 1/255
```

```
Router#show interfaces gigabitEthernet 0/0/1
GigabitEthernet0/0/1 is up, line protocol is up (connected)
  Hardware is ISR4331-3x1GE, address is 000d.bd3b.d902 (bia 000d.bd3b.d902)
  Internet address is 192.168.2.1/24
  MTU 1500 bytes, BW 1000000 Kbit, DLY 100 usec,
     reliability 255/255, txload 1/255, rxload 1/255
```

# IP Fragmentation and Reassembly(3)




Q: While PC0 send a packet to PC1, and then PC1 reply a packet to PC0, **which device would do the IP fragmentation and which device would do the IP reassembly**?

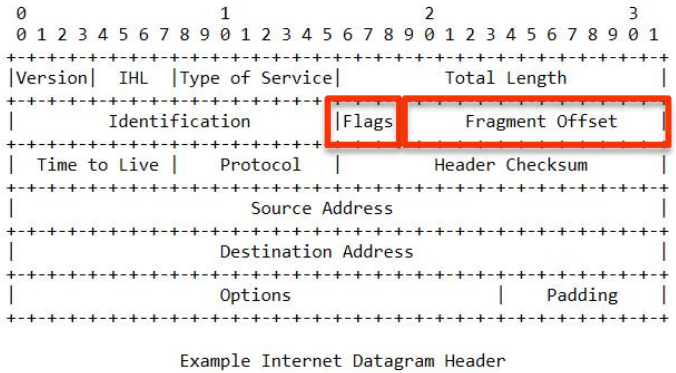
TIPS:
The simulation information would show you the answer

Simulation Panel

Event List

| Vis. | Time(sec) | Last Device | At Device | Type |
|---|---|---|---|---|
| | 0.000 | -- | PC0 | ICMP |
| | 0.001 | PC0 | Router0 | ICMP |
| | 0.002 | Router0 | PC1 | ICMP |
| | 0.003 | PC1 | Router0 | ICMP |
| | 0.003 | -- | Router0 | ICMP |
| | 0.003 | -- | Router0 | ICMP |
| | 0.004 | Router0 | PC0 | ICMP |
| | 0.004 | -- | Router0 | ICMP |
| | 0.005 | Router0 | PC0 | ICMP |
| | 0.005 | -- | Router0 | ICMP |
| | 0.006 | Router0 | PC0 | ICMP |
| | 1.007 | -- | PC0 | ICMP |
| | 1.008 | PC0 | Router0 | ICMP |
| | 1.009 | Router0 | PC1 | ICMP |

# IP Fragment(1)

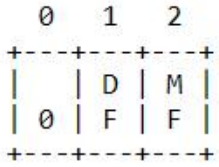



Example Internet Datagram Header

Flags: 3 bits

Various Control Flags.

Bit 0: reserved, must be zero
Bit 1: (DF) 0 = May Fragment, 1 = Don't Fragment.
Bit 2: (MF) 0 = Last Fragment, 1 = More Fragments.



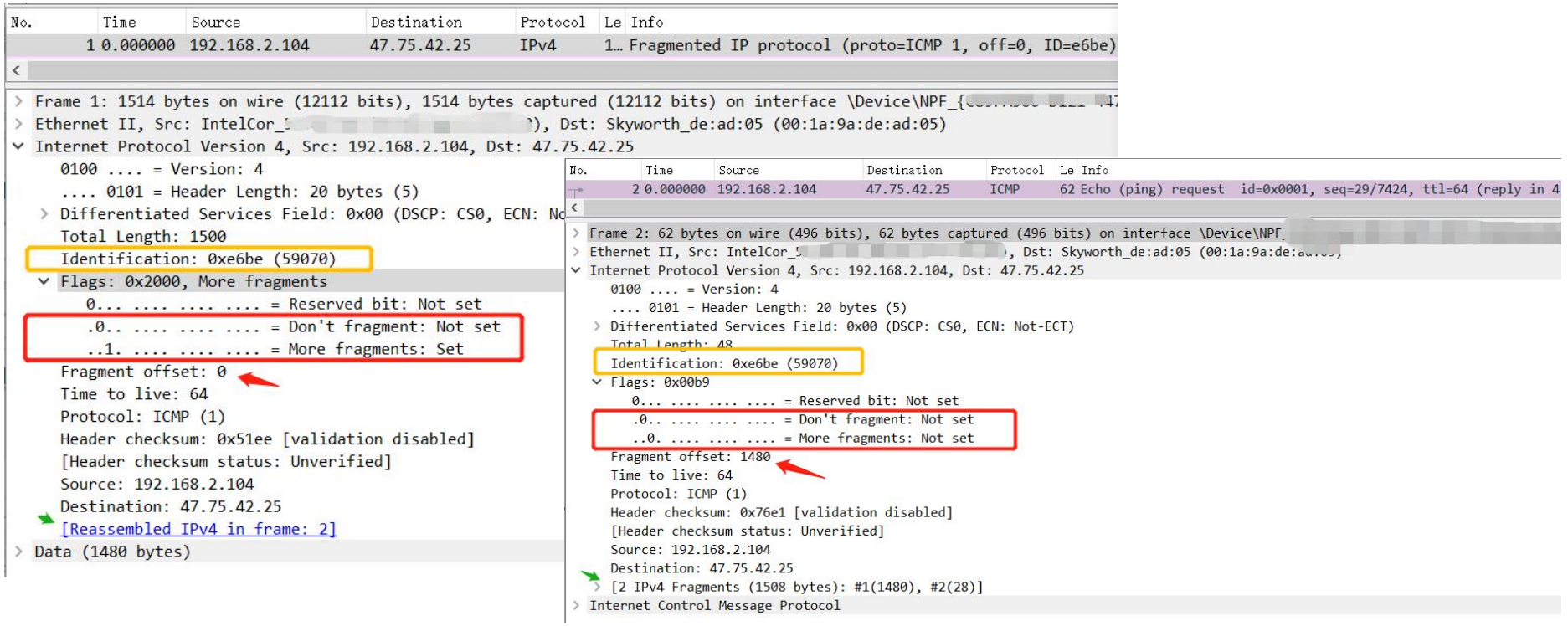

```
Internet Protocol Version 4, Src: 192.168.2.104 (192.168.2.104), Dst: 116.7.234.3 (116.7.234.3)
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
      0000 00.. = Differentiated Services Codepoint: Default (0)
      .... ..00 = Explicit Congestion Notification: Not ECN-Capable Transport (0)
  Total Length: 1020
  Identification: 0x0a9c (2716)
  Flags: 0x00b9
      0... .... .... .... = Reserved bit: Not set
      .0.. .... .... .... = Don't fragment: Not set
      ..0. .... .... .... = More fragments: Not set
      ...0 0000 1011 1001 = Fragment offset: 185
```

**Fragment Offset**: 13 bits
This field indicates where in the datagram this fragment belongs.
The fragment offset is measured **in units of 8 octets** (64 bits). The first fragment has offset zero.

*Tips in Wireshark : ip.flags.mf*

# IP Fragment(2)

Initial the session with following cmd command: ping www.example.cn –l _?_




Identification: An internet header field carrying the identifying value assigned by the sender to aid in assembling the fragments of a datagram.

*Tips in Wireshark : ip.id*

# Practise 11.1

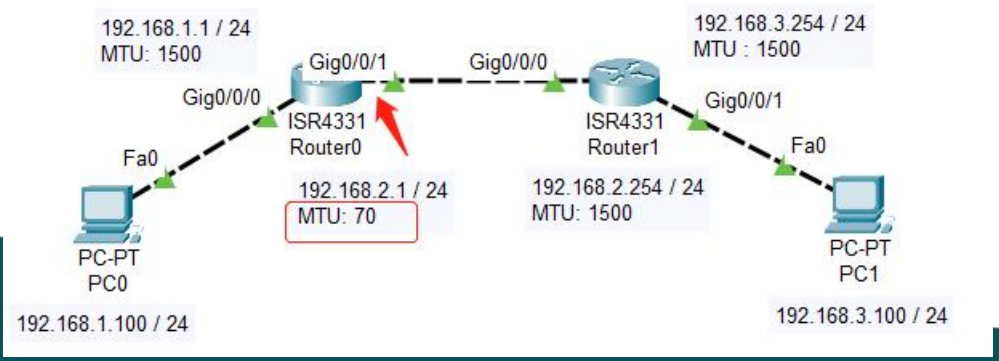
11.1-1) Initiates an ICMP session to test if www.example.com is reachable (setting the packet size to 2022B), and capture the packets.

- How to initiates an ICMP Echo request with 2022B length?
- Is there any fragmentation on the IP packets, how to find them?
- How many fragments are the 2022-Byte-length IP packet divided into?

- How to identify the ICMP Echo request and Echo reply?
- For the ICMP Echo request, which IP fragment is the first one, which is the last ? How to identify them?
- What's the length of each IP fragment? Is the sum of each fragment's length equal to the original IP packet ?

11.1-2) Build the network as the picture showed below, set the MUT of the interface Gig0/0/1 in Router0 as 70, keep the MUT of the interface Gig0/0/0 in Router0 as 1500

Do the test and answer: While PC0 send a packet to PC1, and then PC1 reply a packet to PC0, which device would do the IP fragmentation and which device would do the IP reassembly?




SUSTech
Southern University of Science and Technology

# Part B. ICMP

- **ICMP** is used from gateways to hosts and between hosts to **report errors and make routing suggestions.**

- ICMP and IP :
  - Internet protocol **errors may be reported** via the ICMP messages

  - **ICMP uses the basic support of IP** as if it were a higher level protocol, however, **ICMP is actually an integral part of IP, and must be implemented by every IP module.**

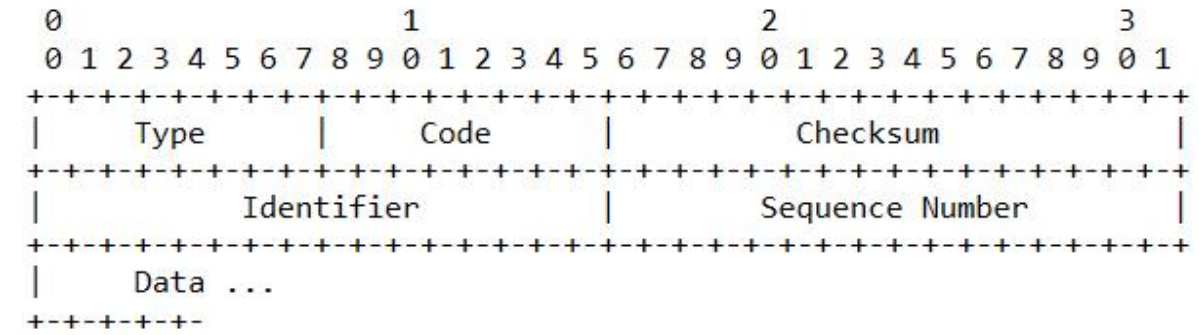


https://tools.ietf.org/html/rfc792

# ICMP (Echo and Echo Reply)

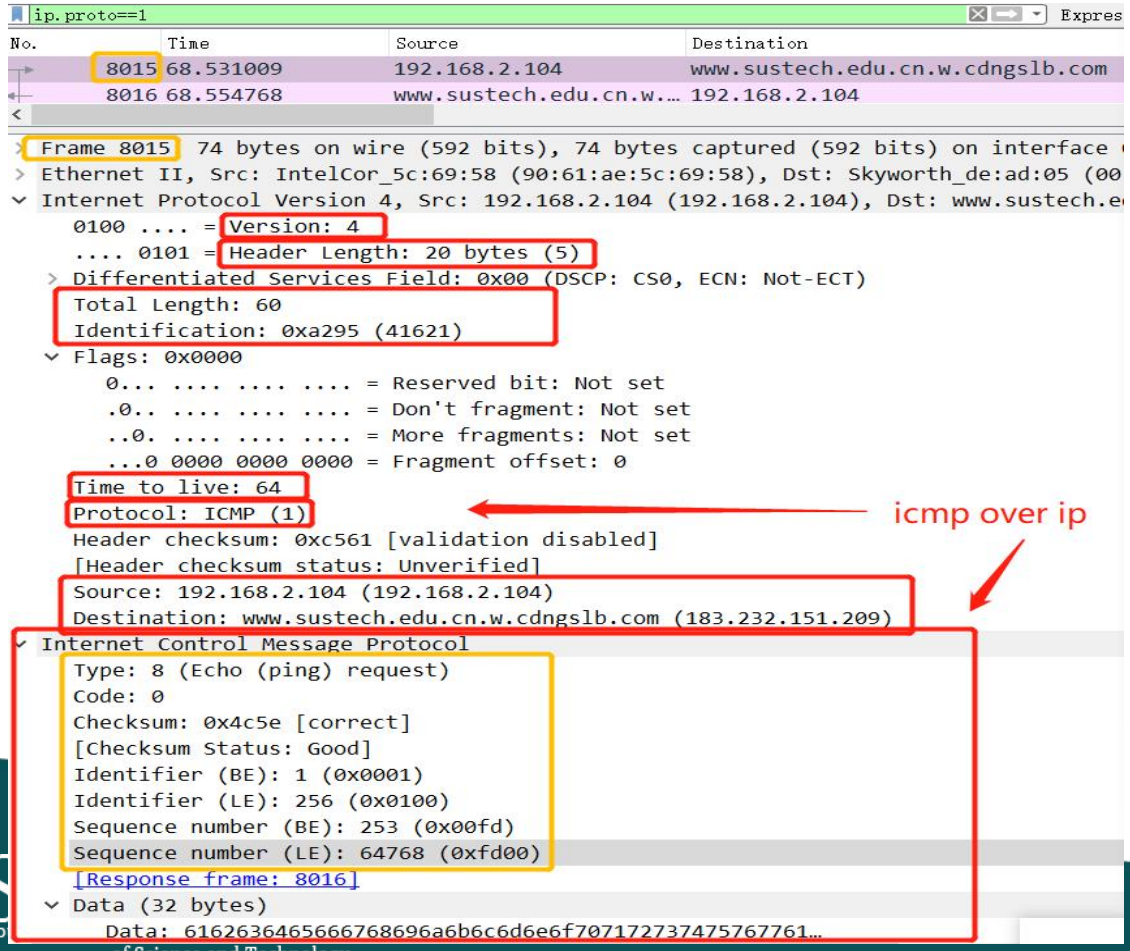
- The data received in the echo message must be returned in the echo reply message.
- **Type**
  - **8 for echo request message;**
  - **0 for echo reply message.**
- **Code**
  - **0**
- The identifier and sequence number may be used by the echo sender to aid in matching the replies with the echo requests. The echoer returns these same values in the echo reply.

Echo or Echo Reply Message



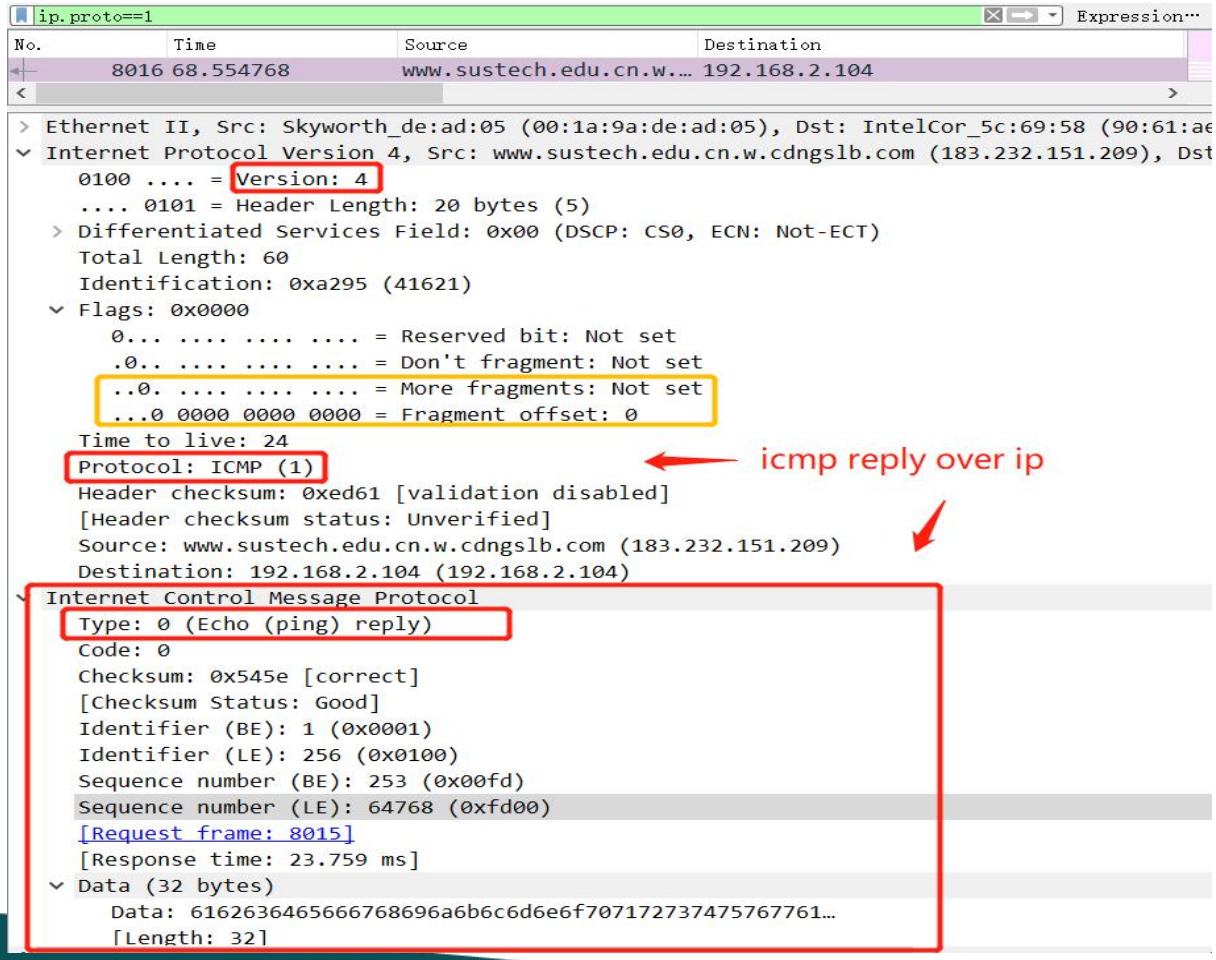
# ICMP Echo Request

Initial the session with following cmd command: ping www.sustech.edu.cn




*Tips in Wireshark :*
*ip.proto == 1 or*
*ICMP.type*

# ICMP Echo Reply

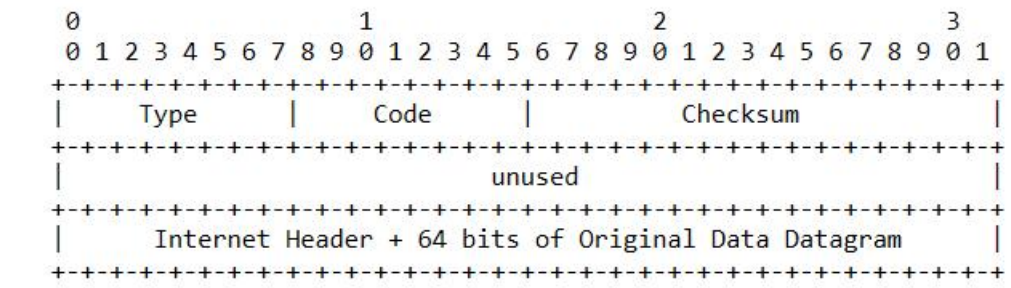



*Tips in Wireshark : ICMP.type*

SUSTech
Southern University
of Science and Technology

# ICMP： Time Exceeded(1)

Time Exceeded Message




**Type： 11**

**Code 0 = time to live exceeded in transit;**
**Code 1 = fragment reassembly time exceeded.**

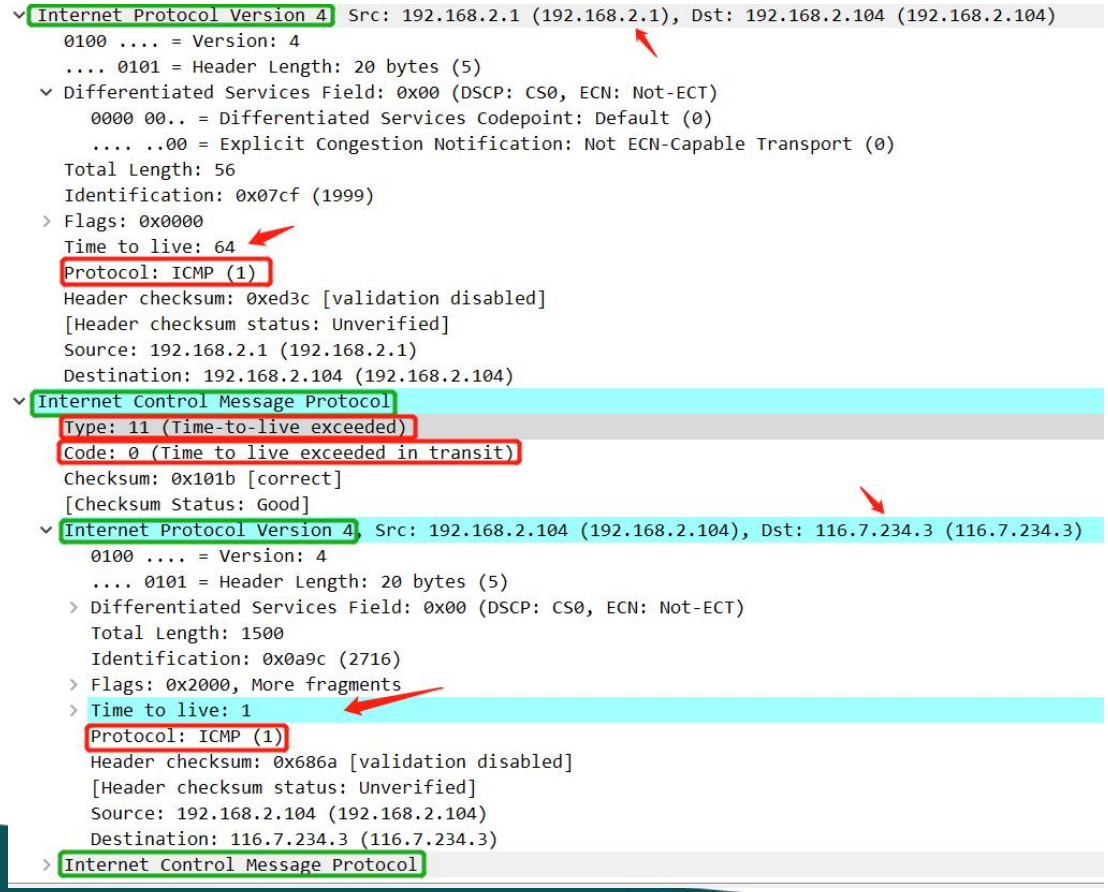
If the gateway processing a datagram finds the time to live field is zero it must discard the datagram. The gateway may also notify the source host via the time exceeded message.

If a host reassembling a fragmented datagram cannot complete the reassembly due to missing fragments within its time limit it discards the datagram, and it may send a time exceeded message.

**Code 0** may be received from a gateway. **Code 1** may be received from a host.

# ICMP： Time Exceeded(2)- time to live exceeded

Initial the session with following cmd: tracert / traceroute




Q:

1. Is the outside IP's src address same with the inside IP's dest address? Why?

2. Is the TTL of outside IP same with which in inside IP?
why?

*Tips in Wireshark : ICMP.type*

# Practise 11.2

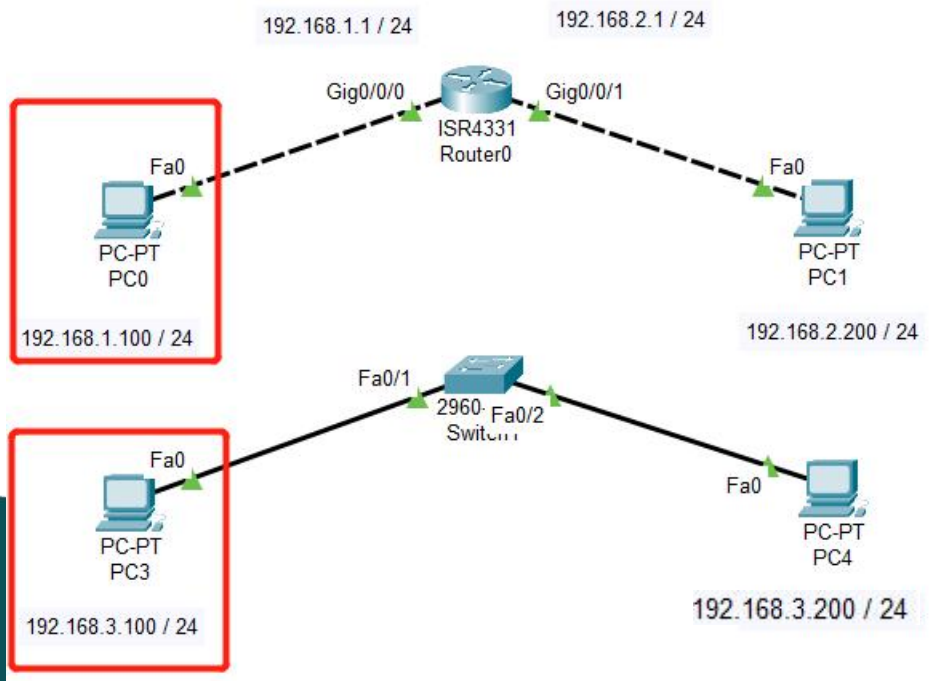
11.2-1) Use tracert(windows) / traceroute(linux or MacOS) to trace the route from your host to www.sustech.edu.cn, and capture the packets while tracing.

- Is there any 'Time-to-live exceeded' ICMP packets?
- What's the difference between these ICMP packets which are invoked by 'tracert' and ICMP echo request/replay packets which are invoked by 'ping' ? List at least 3 aspects.

11.2-2) Build the network as the picture showed bellow

- Invoke the icmp request on PC0 to test if PC3 is reachable
- Invoke the icmp request on PC3 to test if PC0 is reachable

Is the running result same for the both two test, if they are different, please tell the reason.



# Part C. IPv6(1)

- **IPv6** is a new version of the Internet Protocol, designed as the successor to IPv4. The changes from IPv4 to IPv6 fall primarily into the following categories:
- **Expanded Addressing Capabilities**: IPv6 increases the IP address size from 32 bits to 128 bits, to support more levels of addressing hierarchy, a much greater number of addressable nodes, and simpler auto-configuration of addresses. The scalability of multicast routing is improved by adding a "scope" field to multicast addresses. And a new type of address called an "anycast address" is defined, used to send a packet to any one of a group of nodes.
- **Header Format Simplification**: Some IPv4 header fields have been dropped or made optional, to reduce the common-case processing cost of packet handling and to limit the bandwidth cost of the IPv6 header.
- **Improved Support for Extensions and Options**: Changes in the way IP header options are encoded allows for more efficient forwarding, less stringent limits on the length of options, and greater flexibility for introducing new options in the future.
- **Flow Labeling Capability**: A new capability is added to enable the labeling of packets belonging to particular traffic "flows" for which the sender requests special handling, such as non-default quality of service or "real-time" service.
- **Authentication and Privacy Capabilities**: Extensions to support authentication, data integrity, and (optional) data confidentiality are specified for IPv6.



https://tools.ietf.org/html/rfc2460

# IPv6(2)

- **Version**

  **4-bit** Internet Protocol version number = 6.

- **Traffic Class**

  **8-bit** traffic class field.

- **Flow Label**

  **20-bit** flow label.

- **Payload Length**

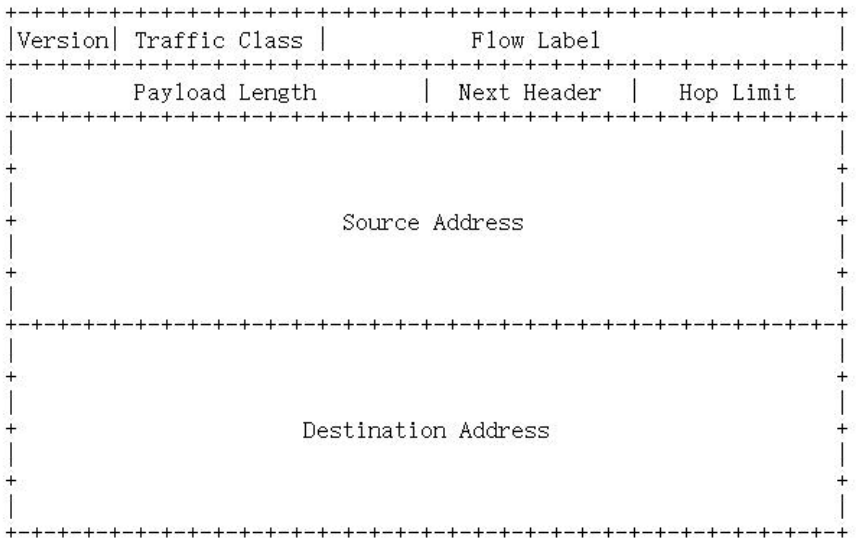
  **16-bit unsigned integer**. **Length of the IPv6 payload**, i.e., the rest of the packet following this IPv6 header, in octets. (Note that any extension headers present are considered part of the payload, i.e., included in the length count.)

- **Next Header**

  **8-bit selector**. Identifies the type of header immediately following the IPv6 header.

- **Hop Limit**

  **8-bit unsigned integer**. Decremented by 1 by each node that forwards the packet. The packet is discarded if Hop Limit is decremented to zero.



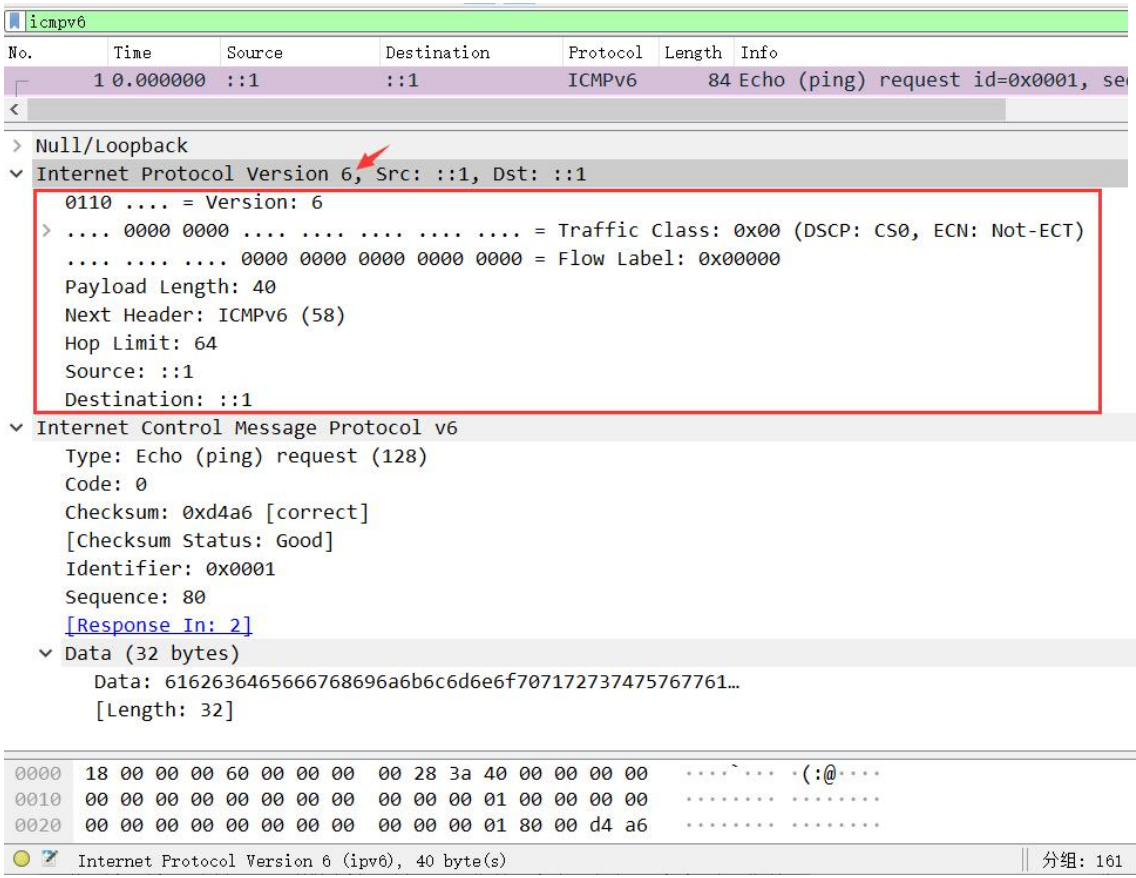

- **Source Address**

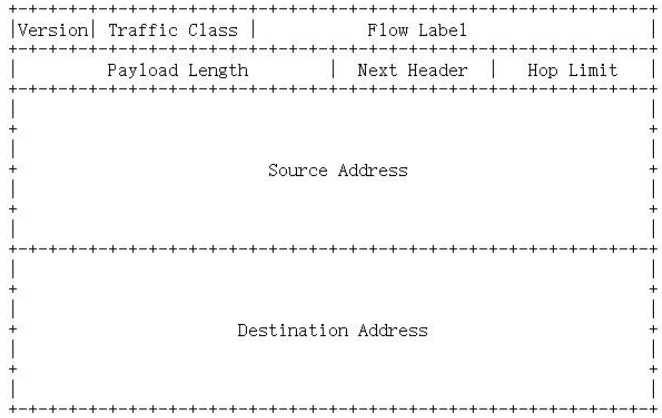
  **128-bit** address of the originator of the packet

- **Destination Address**

  **128-bit** address of the intended recipient of the packet. (possibly not the ultimate recipient, if a Routing header is present)

# IPv6(3)



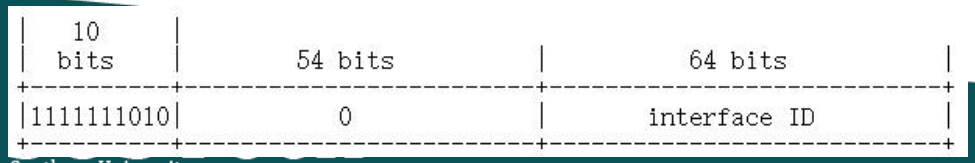




using 'ping -6 localhost' to invoke an ICMPv6 transaction.

# IPv6 Address

- Text Representation of Addresses
  - The preferred form is x:x:x:x:x:x:x:x, where the 'x's are the hexadecimal values of the eight 16-bit pieces of the address
  - In order to make writing addresses containing zero bits easier a special syntax is available to compress the zeros. The use of “::” indicates multiple groups of 16-bits of zeros. The "::" can only appear once in an address.
- Address Type Representation
  - The address 0:0:0:0:0:0:0:0 is called the unspecified address.
  - The unicast address 0:0:0:0:0:0:0:1 is called the loopback address.
  - Link-Local Unicast Addresses are designed to be used for addressing on a single link for purposes such as auto-address configuration, neighbor discovery, or when no routers are present.

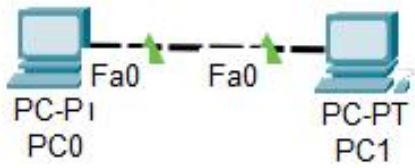
```
IPv6 地址 . . . . . . . . . . . . : 2409:8a55:3050:f6b0:48cf:2c49:a3fe:6381
临时 IPv6 地址. . . . . . . . . . : 2409:8a55:3050:f6b0:6cf9:d6fc:544:f4c7
本地链接 IPv6 地址. . . . . . . . : fe80::48cf:2c49:a3fe:6381%17
```






https://tools.ietf.org/html/rfc2373

# Practise 11.3




11.3-1). Use Packet-tracer to build a LAN with 2 PCs connected directely.

- What's link-local unicast IPv6 address of these 2 PCs? (Tips: You can check IPv6 configuration in IP configuration dialog.)
- Initiates an ICMPv6 session on PC0 to PC1, capture the packets.(Tips: You can use ping command in Command Prompt)
- What's the difference between IPv4 datagrame and IPv6 datagrame? List at least 3 aspects. (Tips: You can run ping command under simulation mode in Packet Tracer.)
- Does these two IPv6 addresses belong to the same sub-net, what is the sub-net ID of these two IPv6 addresses?

11.3-2). Buile the network as the picture described bellow. Set the IPv6 address on PC3, DNS Server and HTTP Sever. Suppose the Domain name of the HTTP Server is "www.sustech.edu.cn", finish the configureation to make the HTTP Server is accessable by its Domain name.