

# CS 305 Lab Tutorial

## Lab11 IP & ICMP

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

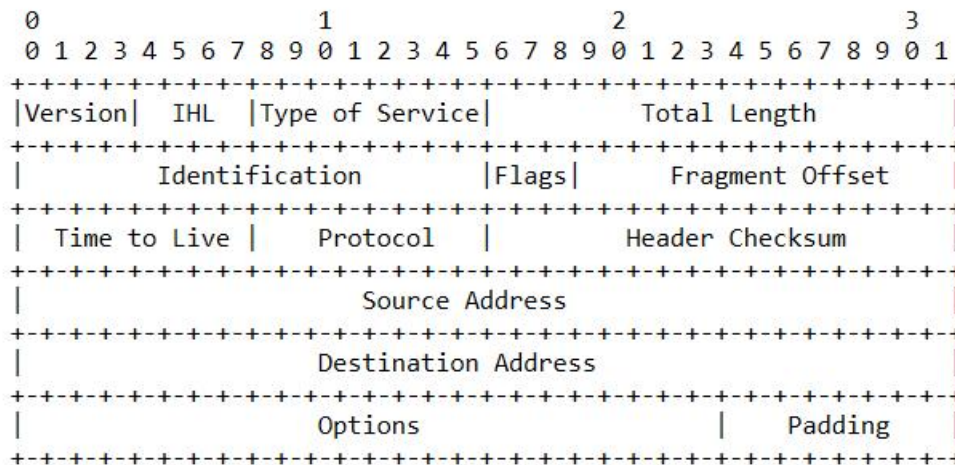
# Topic

- 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.

# 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.

# “Protocol” Field

0					1					2					3																																			
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1																			
Version					IHL					Type of Service					Total Length																																			
										Identification										Flags					Fragment Offset																									
										Time to Live															Protocol															Header Checksum										
Source Address																																																		
Destination Address																																																		
																				Options																				Padding										

### Example Internet Datagram Header

```

    Internet Protocol Version 4, Src: 192.168.2.104 (192.168.2.104), Dst: 192.168.2.104
      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)

```

```
Protocol: ICMP (1)
Header checksum: 0x8493 [validation
[Header checksum status: Unverified
Source: 192.168.2.104 (192.168.2.104)
Destination: 116.7.234.3 (116.7.234.3)
> [2 IPv4 Fragments (2480 bytes): #1(
v 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 [valid]
    [Header checksum status: Unverified]
    Source: 192.168.2.104 (192.168.2.104)
    Destination: tg-in-f113.1e100...
> Transmission Control Protocol

```

- > Transmission Control Protocol

```

v Internet Protocol Version 4, Src: tw.net-east.com (116.77.76.254), Dst: 192.168.2.104
  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)

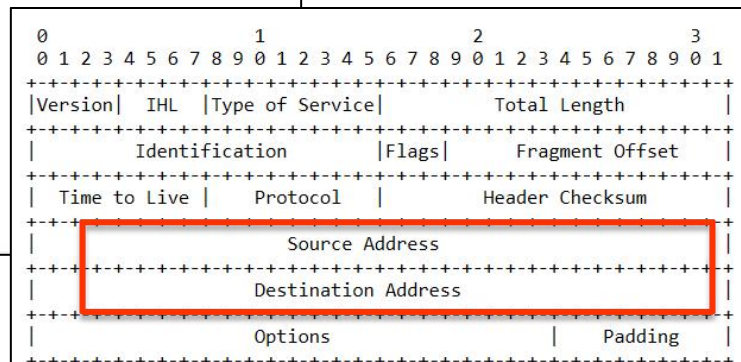
```

**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)
> 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
```

```
> 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



# “IHL” and “Total Length” Field

Initial the session with following cmd command: ping [www.example.com](http://www.example.com) -l 2000

No.	Time	Source	Destination
<	2179 42.035965	192.168.2.104	www.example.com
> Frame 2179: 1514 bytes on wire (12112 bits), 1514 bytes captured (12112 bits) on			
> Ethernet II, Src: IntelCor_5c:69:58 (90:61:ae:5c:69:58), Dst: Skyworth_de:ad:05			
v Internet Protocol Version 4, Src: 192.168.2.104 (192.168.2.104), Dst: www.examp			
0100 .... = Version: 4			
... 0101 = Header Length: 20 bytes (5)			
> Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)			
Total Length: 1500			
Identification: 0xe55e (58718)			
> Flags: 0x2000, More fragments			
Time to live: 64			
Protocol: ICMP (1)			
Header checksum: 0x76d7 [validation disabled]			
[Header checksum status: Unverified]			
Source: 192.168.2.104 (192.168.2.104)			
Destination: www.example.com (93.184.216.34)			
Reassembled IPv4 in frame: 2180			
v Data (1480 bytes)			
Data: 08007792000103e56162636465666768696a6b6c6d6e6f70...			
[Length: 1480]			

0000	00 1a 9a de ad 05 90 61 ae 5c 69 58 08 00 45 00	.....a \ix..E.
0010	05 dc e5 5e 20 00 40 01 76 d7 c0 a8 02 68 5d b8	...^.@ v...h].
0020	d8 22 08 00 77 92 00 01 03 e5 61 62 63 64 65 66	."w...abcdef
0030	67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76	ghijklmn opqrstuv

**IHL: 4 bits**

Internet Header Length 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.

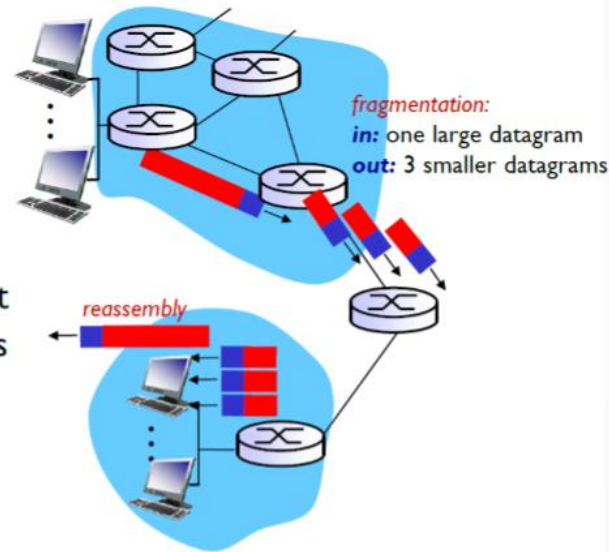
0	1																2																3															
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1																	
Version				IHL				Type of Service				Total Length																																				
Identification																Flags				Fragment Offset																												
Time to Live								Protocol								Header Checksum																																
Source Address																																																
Destination Address																																																
Options																								Padding																								

Example Internet Datagram Header

# 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

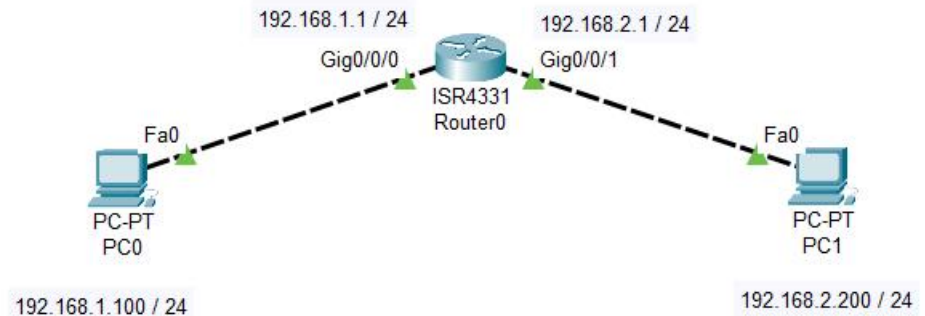




# IP Fragmentation and Reassembly(2)

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**?

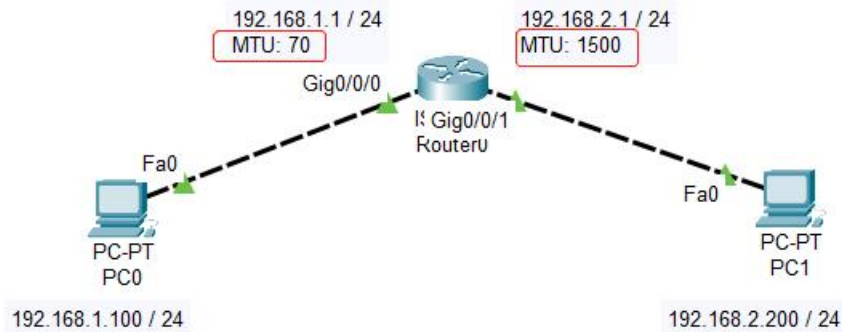


```
Router(config)#interface gigabitEthernet 0/0/0
Router(config-if)#mtu 70
%IPV6-3-MIN_MTU: The link MTU of GigabitEthernet0/0/0 is below the 1280-byte
minimum IPv6 link MTU. IPv6 may not work correctly on this interface.
Router(config-if)#
```

```
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)

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.

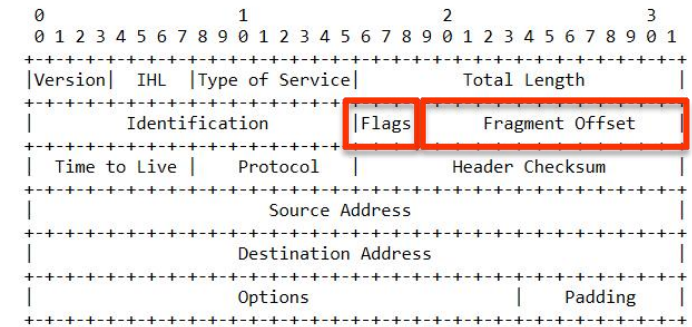
0	1	2
	D	M
0	F	F

**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*



Example Internet Datagram Header

- ✓ 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

# IP Fragment(2)

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

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.2.104	47.75.42.25	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=0, ID=e6be)
<						
> Frame 1: 1514 bytes on wire (12112 bits), 1514 bytes captured (12112 bits) on interface \Device\NPF_{...}						
> Ethernet II, Src: IntelCor_..., Dst: Skyworth_de:ad:05 (00:1a:9a:de:ad:05)						
v Internet Protocol Version 4, Src: 192.168.2.104, Dst: 47.75.42.25						
0100 .... = Version: 4						
.... 0101 = Header Length: 20 bytes (5)						
> Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)						
Total Length: 1500						
Identification: 0xe6be (59070)						
v Flags: 0x2000, More fragments						
0... .. = Reserved bit: Not set						
.0... .. = Don't fragment: Not set						
..1. .... = More fragments: Set						
Fragment offset: 0						
Time to live: 64						
Protocol: ICMP (1)						
Header checksum: 0x51ee [validation disabled]						
[Header checksum status: Unverified]						
Source: 192.168.2.104						
Destination: 47.75.42.25						
[Reassembled IPv4 in frame: 2]						
> Data (1480 bytes)						

No.	Time	Source	Destination	Protocol	Length	Info
2	0.000000	192.168.2.104	47.75.42.25	ICMP	62	Echo (ping) request id=0x0001, seq=29/7424, ttl=64 (reply in 4...)
<						
> Frame 2: 62 bytes on wire (496 bits), 62 bytes captured (496 bits) on interface \Device\NPF_{...}						
> Ethernet II, Src: IntelCor_..., Dst: Skyworth_de:ad:05 (00:1a:9a:de:ad:05)						
v Internet Protocol Version 4, Src: 192.168.2.104, Dst: 47.75.42.25						
0100 .... = Version: 4						
.... 0101 = Header Length: 20 bytes (5)						
> Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)						
Total Length: 48						
Identification: 0xe6be (59070)						
v Flags: 0x00b9						
0... .. = Reserved bit: Not set						
.0... .. = Don't fragment: Not set						
..0. .... = More fragments: Not set						
Fragment offset: 1480						
Time to live: 64						
Protocol: ICMP (1)						
Header checksum: 0x76e1 [validation disabled]						
[Header checksum status: Unverified]						
Source: 192.168.2.104						
Destination: 47.75.42.25						
[2 IPv4 Fragments (1508 bytes): #1(1480), #2(28)]						
> Internet Control Message Protocol						

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*



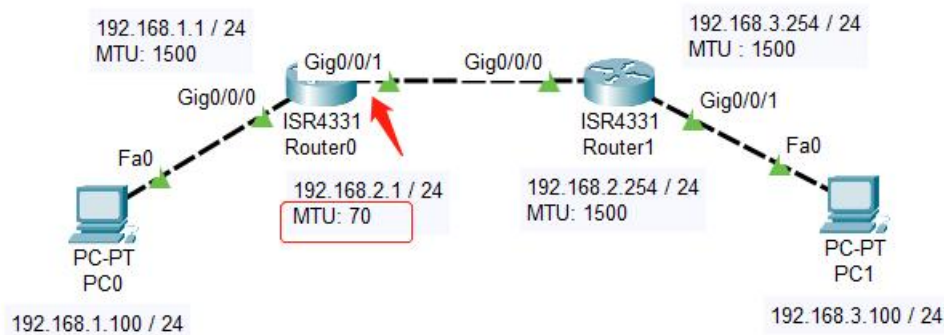
# Practise 11.1

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

- How to initiate 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 MTU of the interface Gig0/0/1 in Router0 as 70, keep the MTU 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?



MTU只和一个接口的出链路有关。且分片由路由器做，组装由PC做。因此，  
去的时候：分片：Router 0  
组装：PC1  
回来的时候：不用分片也不用组装



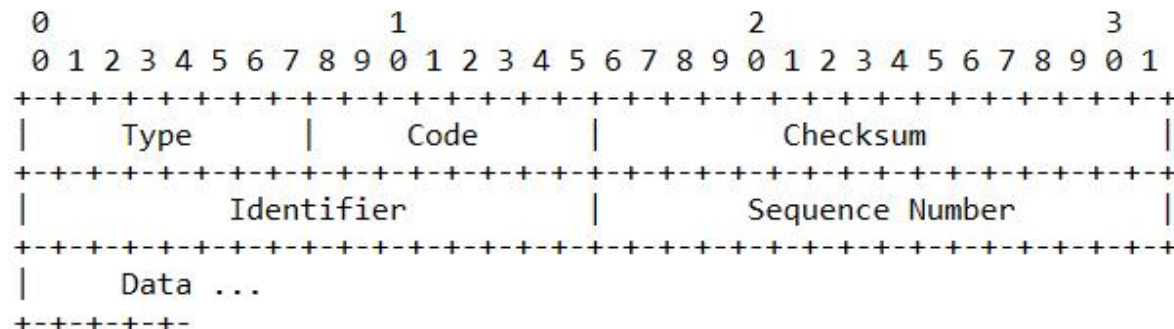
# Part B. ICMP

- **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**.

# 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`

ip.proto==1

No.	Time	Source	Destination
8015	68.531009	192.168.2.104	www.sustech.edu.cn.w.cdngslb.com
8016	68.554768	www.sustech.edu.cn.w...	192.168.2.104

Frame 8015: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0  
Ethernet II, Src: IntelCor\_5c:69:58 (90:61:ae:5c:69:58), Dst: Skyworth\_de:ad:05 (00:0c:29:ad:05:00)  
Internet Protocol Version 4, Src: 192.168.2.104 (192.168.2.104), Dst: www.sustech.edu.cn.w.cdngslb.com (183.232.151.209)

0100 .... = Version: 4  
.... 0101 = Header Length: 20 bytes (5)  
Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)  
Total Length: 60  
Identification: 0xa295 (41621)  
Flags: 0x0000  
0... .. = Reserved bit: Not set  
.0.. .. = Don't fragment: Not set  
..0. .... = More fragments: Not set  
...0 0000 0000 0000 = Fragment offset: 0  
Time to live: 64  
Protocol: ICMP (1)  
Header checksum: 0xc561 [validation disabled]  
[Header checksum status: Unverified]  
Source: 192.168.2.104 (192.168.2.104)  
Destination: www.sustech.edu.cn.w.cdngslb.com (183.232.151.209)  
Internet Control Message Protocol  
Type: 8 (Echo (ping) request)  
Code: 0  
Checksum: 0x4c5e [correct]  
[Checksum Status: Good]  
Identifier (BE): 1 (0x0001)  
Identifier (LE): 256 (0x0100)  
Sequence number (BE): 253 (0x00fd)  
Sequence number (LE): 64768 (0xfd00)  
[Response frame: 8016]  
Data (32 bytes)  
Data: 6162636465666768696a6b6c6d6e6f707172737475767761...

icmp over ip

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

# ICMP Echo Reply

ip.proto==1

No.	Time	Source	Destination
8016	68.554768	www.sustech.edu.cn.w.cdnslb.com	192.168.2.104

> Ethernet II, Src: Skyworth\_de:ad:05 (00:1a:9a:de:ad:05), Dst: IntelCor\_5c:69:58 (90:61:ae:5c:69:58)

✓ Internet Protocol Version 4, Src: www.sustech.edu.cn.w.cdnslb.com (183.232.151.209), Dst: 192.168.2.104

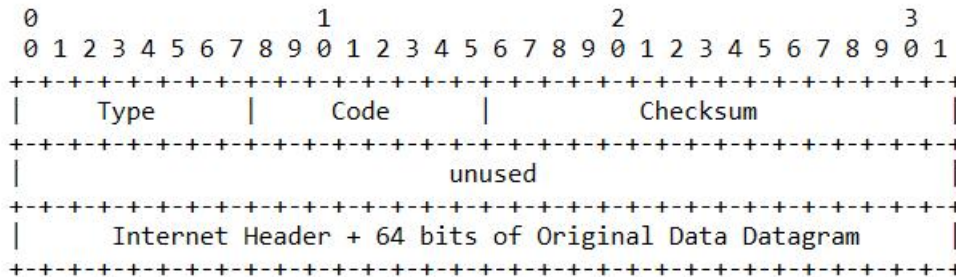
- 0100 .... = Version: 4
- ... 0101 = Header Length: 20 bytes (5)
- > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
- Total Length: 60
- Identification: 0xa295 (41621)
- ✓ Flags: 0x0000
  - 0... .. = Reserved bit: Not set
  - .0... .. = Don't fragment: Not set
  - ..0... .. = More fragments: Not set
  - ...0 0000 0000 0000 = Fragment offset: 0
- Time to live: 24
- Protocol: ICMP (1)
- Header checksum: 0xed61 [validation disabled]
- [Header checksum status: Unverified]
- Source: www.sustech.edu.cn.w.cdnslb.com (183.232.151.209)
- Destination: 192.168.2.104 (192.168.2.104)
- ✓ Internet Control Message Protocol
  - Type: 0 (Echo (ping) reply)
  - Code: 0
  - Checksum: 0x545e [correct]
  - [Checksum Status: Good]
  - Identifier (BE): 1 (0x0001)
  - Identifier (LE): 256 (0x0100)
  - Sequence number (BE): 253 (0x00fd)
  - Sequence number (LE): 64768 (0xfd00)
  - [Request frame: 8015]
  - [Response time: 23.759 ms]
  - ✓ Data (32 bytes)
    - Data: 6162636465666768696a6b6c6d6e6f707172737475767761...
    - [Length: 32]

icmp reply over ip

# 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`

```

Internet Protocol Version 4 Src: 192.168.2.1 (192.168.2.1), Dst: 192.168.2.104 (192.168.2.104)
  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: 56
  Identification: 0x07cf (1999)
  > Flags: 0x0000
  Time to live: 64
  Protocol: ICMP (1)
  Header checksum: 0xed3c [validation disabled]
  [Header checksum status: Unverified]
  Source: 192.168.2.1 (192.168.2.1)
  Destination: 192.168.2.104 (192.168.2.104)

Internet Control Message Protocol
  Type: 11 (Time-to-live exceeded)
  Code: 0 (Time to live exceeded in transit)
  Checksum: 0x101b [correct]
  [Checksum Status: Good]

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)
  Total Length: 1500
  Identification: 0x0a9c (2716)
  > Flags: 0x2000, More fragments
  > Time to live: 1
  Protocol: ICMP (1)
  Header checksum: 0x686a [validation disabled]
  [Header checksum status: Unverified]
  Source: 192.168.2.104 (192.168.2.104)
  Destination: 116.7.234.3 (116.7.234.3)

Internet Control Message Protocol
```

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](http://www.sustech.edu.cn), and capture the packets while tracing.

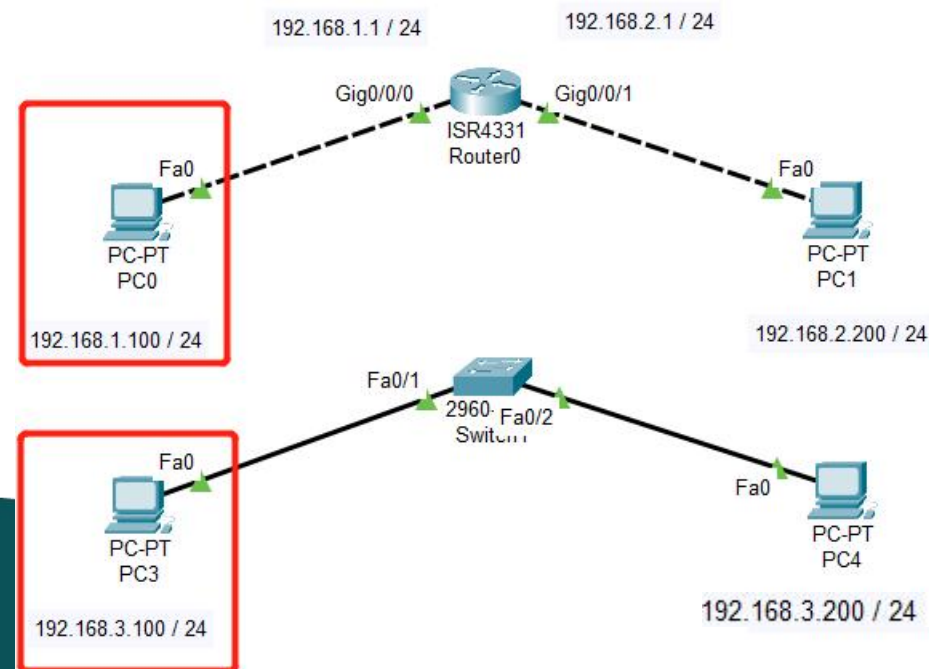
- Is there any 'Time-to-live exceeded' ICMP packets?
- What's the difference between these ICMP packets which are invoked by 'tracert' and ICMP echo request/replay packets which are invoked by 'ping'? List at least 3 aspects.

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.

PC0到PC3显示目标主机不可达，PC3到PC0显示的却是超时  
PC0-PC3：路由器因为找不到转发表和正确的出链路  
PC3-PC0：交换机没有这个功能因此还在找，所以一直在等直到超时  
因此，路由器有生命，交换机无生命

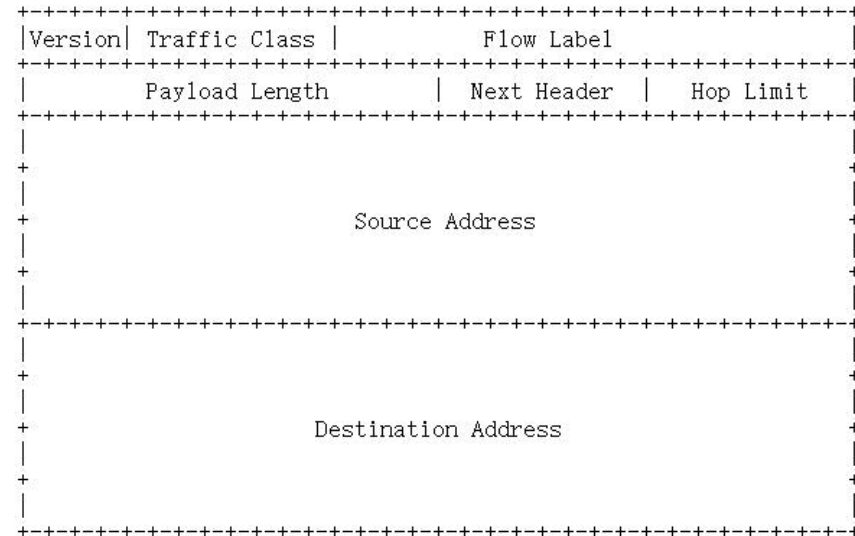


# Part C. IPv6(1)

- **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.

# IPv6(2)

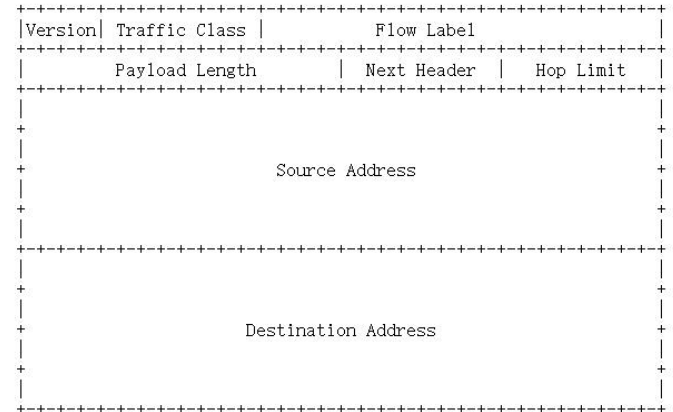
- **Version**  
**4-bit** Internet Protocol version number = 6.
- **Traffic Class**  
**8-bit** traffic class field.
- **Flow Label**  
**20-bit** flow label.
- **Payload Length**  
**16-bit unsigned integer. Length of the IPv6 payload,** i.e., the rest of the packet following this IPv6 header, in octets. (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.** Decrement 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)

icmpv6						
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	::1	::1	ICMPv6	84	Echo (ping) request id=0x0001, se
<						
> Null/Loopback						
v Internet Protocol Version 6, Src: ::1, Dst: ::1						
0110 .... = Version: 6 > .... 0000 0000 .... = Traffic Class: 0x00 (DSCP: CS0, ECN: Not-ECT) .... 0000 0000 0000 0000 = Flow Label: 0x000000 Payload Length: 40 Next Header: ICMPv6 (58) Hop Limit: 64 Source: ::1 Destination: ::1						
v Internet Control Message Protocol v6						
Type: Echo (ping) request (128)						
Code: 0						
Checksum: 0xd4a6 [correct]						
[Checksum Status: Good]						
Identifier: 0x0001						
Sequence: 80						
<a href="#">[Response In: 2]</a>						
v Data (32 bytes)						
Data: 6162636465666768696a6b6c6d6e6f707172737475767761...						
[Length: 32]						
0000 18 00 00 00 60 00 00 00 00 28 3a 40 00 00 00 00 .....(:@.... 0010 00 00 00 00 00 00 00 00 00 00 00 01 00 00 00 00 ..... 0020 00 00 00 00 00 00 00 00 00 00 00 01 80 00 d4 a6 .....						
Internet Protocol Version 6 (ipv6), 40 byte(s)					分组: 161	



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



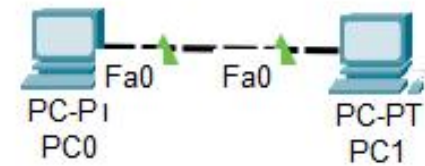
# IPv6 Address

- 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
```

10 bits	54 bits	64 bits
1111111010	0	interface ID

# Practise 11.3



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

- 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 datagram and IPv6 datagram? 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 configuration to make the HTTP Server is accessable by its Domain name.

11.3-1

Q1 : PC0 : FE80::201:C9FF:FEEB:6A28

PC1 : FE80::206:2AFF:FE44:E736

