

Junhyeok Jeong

Bill Pfeil

CS 372-X001

12/6/2019

Lab 5

1. What is the 48-bit Ethernet address of your computer?

The screenshot shows a Wireshark packet capture of an HTTP GET request. The packet list pane shows a list of packets, with packet 106 selected. The packet details pane shows the structure of the selected packet, including the Ethernet II frame, Internet Protocol Version 4, Transmission Control Protocol, and Hypertext Transfer Protocol. The packet bytes pane shows the raw data of the frame.

No.	Time	Source	Destination	Protocol	Length	Info
103	15:30:12.448842	192.168.0.24	192.168.0.21	TCP	164	8009 → 49847 [PSH, ACK] Seq=111 Ack=221 W
104	15:30:12.453687	128.119.245.12	192.168.0.21	TCP	66	80 → 56806 [SYN, ACK] Seq=0 Ack=1 Win=292
105	15:30:12.453739	192.168.0.21	128.119.245.12	TCP	54	56806 → 80 [ACK] Seq=1 Ack=1 Win=262144 L
106	15:30:12.453866	192.168.0.21	128.119.245.12	HTTP	484	GET /wireshark-labs/HTTP-ethereal-lab-fil
107	15:30:12.458357	128.119.245.12	192.168.0.21	TCP	66	80 → 56807 [SYN, ACK] Seq=0 Ack=1 Win=292
108	15:30:12.458386	192.168.0.21	128.119.245.12	TCP	54	56807 → 80 [ACK] Seq=1 Ack=1 Win=262144 L
109	15:30:12.489288	192.168.0.21	192.168.0.24	TCP	54	49847 → 8009 [ACK] Seq=221 Ack=221 Win=10
110	15:30:12.496273	40.117.135.216	192.168.0.21	TLSv1...	105	Change Cipher Spec, Encrypted Handshake M
111	15:30:12.496301	192.168.0.21	40.117.135.216	TCP	54	56805 → 443 [ACK] Seq=334 Ack=7085 Win=26
112	15:30:12.496741	192.168.0.21	40.117.135.216	TLSv1...	543	Application Data

> Frame 106: 484 bytes on wire (3872 bits), 484 bytes captured (3872 bits) on interface 0

▼ Ethernet II, Src: AsustekC_91:eb:e8 (04:d4:c4:91:eb:e8), Dst: Motorola_f5:79:b1 (88:b4:a6:f5:79:b1)

> Destination: Motorola_f5:79:b1 (88:b4:a6:f5:79:b1)

▼ Source: AsustekC_91:eb:e8 (04:d4:c4:91:eb:e8)

Address: AsustekC_91:eb:e8 (04:d4:c4:91:eb:e8)

.... 00. = LG bit: Globally unique address (factory default)

.... 00. = IG bit: Individual address (unicast)

Type: IPv4 (0x0800)

> Internet Protocol Version 4, Src: 192.168.0.21, Dst: 128.119.245.12

> Transmission Control Protocol, Src Port: 56806, Dst Port: 80, Seq: 1, Ack: 1, Len: 430

> Hypertext Transfer Protocol

0000 88 b4 a6 f5 79 b1 04 d4 c4 91 eb e8 08 00 45 00 ...y... ..E-

0010 01 d6 6e 0f 40 00 80 06 00 00 c0 a8 00 15 80 77 ...n.@... ..w

0020 f5 0c dd e6 00 50 67 94 6d 75 6b 56 18 f6 50 18 ...Pg. mukV..P.

0030 04 00 38 0a 00 00 47 45 54 20 2f 77 69 72 65 73 ...8...GE T /wires

0040 68 61 72 6b 2d 6c 61 62 73 2f 48 54 54 50 2d 65 hark-lab s/HTTP-e

Source Hardware Address (eth.src), 6 bytes

Packets: 137 · Displayed: 137 (100.0%) · Dropped: 0 (0.0%)

Profile: Default

*Ethernet

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/> Expression...

No.	Time	Source	Destination	Protocol	Length	Info
100	15:30:12.415125	AsustekC_91:eb:e8	Motorola_f5:79:b1	0x0800	54	IPv4
101	15:30:12.416501	AsustekC_91:eb:e8	Motorola_f5:79:b1	0x0800	147	IPv4
102	15:30:12.447421	AsustekC_91:eb:e8	Google_49:ea:f1	0x0800	164	IPv4
103	15:30:12.448842	Google_49:ea:f1	AsustekC_91:eb:e8	0x0800	164	IPv4
104	15:30:12.453687	Motorola_f5:79:b1	AsustekC_91:eb:e8	0x0800	66	IPv4
105	15:30:12.453739	AsustekC_91:eb:e8	Motorola_f5:79:b1	0x0800	54	IPv4
106	15:30:12.453866	AsustekC_91:eb:e8	Motorola_f5:79:b1	0x0800	484	IPv4
107	15:30:12.458357	Motorola_f5:79:b1	AsustekC_91:eb:e8	0x0800	66	IPv4
108	15:30:12.458386	AsustekC_91:eb:e8	Motorola_f5:79:b1	0x0800	54	IPv4
109	15:30:12.489288	AsustekC_91:eb:e8	Google_49:ea:f1	0x0800	54	IPv4

> Frame 106: 484 bytes on wire (3872 bits), 484 bytes captured (3872 bits) on interface 0

▼ Ethernet II, Src: AsustekC_91:eb:e8 (04:d4:c4:91:eb:e8), Dst: Motorola_f5:79:b1 (88:b4:a6:f5:79:b1)

- ▼ Destination: Motorola_f5:79:b1 (88:b4:a6:f5:79:b1)
Address: Motorola_f5:79:b1 (88:b4:a6:f5:79:b1)
.... ..0. = LG bit: Globally unique address (factory default)
.... ..0 = IG bit: Individual address (unicast)
- ▼ Source: AsustekC_91:eb:e8 (04:d4:c4:91:eb:e8)
Address: AsustekC_91:eb:e8 (04:d4:c4:91:eb:e8)
.... ..0. = LG bit: Globally unique address (factory default)
.... ..0 = IG bit: Individual address (unicast)
Type: IPv4 (0x0800)
- ▼ Data (470 bytes)
Data: 450001d66e0f400080060000c0a800158077f50cdde60050...
[Length: 470]

0000	88 b4 a6 f5 79 b1 04 d4 c4 91 eb e8 08 00 45 00y... ..E-
0010	01 d6 6e 0f 40 00 80 06 00 00 c0 a8 00 15 80 77	..n-@... ..w
0020	f5 0c dd e6 00 50 67 94 6d 75 6b 56 18 f6 50 18Pg. mukV..P-
0030	04 00 38 0a 00 00 47 45 54 20 2f 77 69 72 65 73	..8...GE T /wires
0040	68 61 72 6b 2d 6c 61 62 73 2f 48 54 54 50 2d 65	hark-lab s/HTTP-e

Source Hardware Address (eth.src), 6 bytes

Packets: 137 · Displayed: 137 (100.0%) · Dropped: 0 (0.0%) | Profile: Default

- My computer's 48-bit Ethernet address is 04:d4:c4:91:eb:e8

2. What is the 48-bit destination address in the Ethernet frame? Is this the Ethernet address of gaia.cs.umass.edu? (Hint: the answer is no). What device has this as its Ethernet address? [Note: this is an important question, and one that students sometimes get wrong. Re-read pages 468-469 in the text and make sure you understand the answer here.]

Wireshark packet capture analysis of an Ethernet frame. The packet list shows frame 106 selected, which is an IPv4 packet from AsustekC_91:eb:e8 to Motorola_f5:79:b1. The packet details pane shows the Ethernet II header, source and destination MAC addresses, and the IPv4 data. The packet bytes pane shows the raw hex and ASCII data, with the ASCII part showing '...y... ..E...'.

- The 48-bit destination address is 88:b4:a6:f5:79:b1. This is not the Ethernet address of gaia.cs.umass.edu. This 48-bit Ethernet address is for my router (Motorola router) to escape the subnet.

3. Give the hexadecimal value for the two-byte Frame type field. What upper layer protocol does this correspond to?

- The hexadecimal value for the two-byte Frame type field is 0x0800. This value corresponds to the IP protocol because the two-byte Frame type field will be passed the IP protocol layer as payload.

4. How many bytes from the very start of the Ethernet frame does the ASCII “G” in “GET” appear in the Ethernet frame?

The image shows a Wireshark capture of an HTTP GET request. The packet list at the top shows a GET request to /wireshark-labs/HTTP-ethereal-lab-file3.html. The packet details pane shows the request method and URI. The packet bytes pane shows the raw data, with the ASCII 'G' in 'GET' highlighted at offset 54.

No.	Time	Source	Destination	Protocol	Length	Info
97	15:30:12.415113	40.117.135.216	192.168.0.21	TCP	1514	443 → 56805 [ACK] Seq=2921 Ack=241 Win=26
98	15:30:12.415113	40.117.135.216	192.168.0.21	TCP	1514	443 → 56805 [ACK] Seq=4381 Ack=241 Win=26
99	15:30:12.415114	40.117.135.216	192.168.0.21	TLSv1...	1247	Server Hello, Certificate, Certificate St
100	15:30:12.415125	192.168.0.21	40.117.135.216	TCP	54	56805 → 443 [ACK] Seq=241 Ack=7034 Win=26
101	15:30:12.416501	192.168.0.21	40.117.135.216	TLSv1...	147	Client Key Exchange, Change Cipher Spec,
102	15:30:12.447421	192.168.0.21	192.168.0.24	TCP	164	49847 → 8009 [PSH, ACK] Seq=111 Ack=111 W
103	15:30:12.448842	192.168.0.24	192.168.0.21	TCP	164	8009 → 49847 [PSH, ACK] Seq=111 Ack=221 W
104	15:30:12.453687	128.119.245.12	192.168.0.21	TCP	66	80 → 56806 [SYN, ACK] Seq=0 Ack=1 Win=292
105	15:30:12.453739	192.168.0.21	128.119.245.12	TCP	54	56806 → 80 [ACK] Seq=1 Ack=1 Win=262144 L
106	15:30:12.453866	192.168.0.21	128.119.245.12	HTTP	484	GET /wireshark-labs/HTTP-ethereal-lab-fil

Destination: Motorola_f5:79:b1 (88:b4:a6:f5:79:b1)
Address: Motorola_f5:79:b1 (88:b4:a6:f5:79:b1)
.... ..0. = LG bit: Globally unique address (factory default)
.... ..0. = IG bit: Individual address (unicast)
Source: AsustekC_91:eb:e8 (04:d4:c4:91:eb:e8)
Address: AsustekC_91:eb:e8 (04:d4:c4:91:eb:e8)
.... ..0. = LG bit: Globally unique address (factory default)
.... ..0. = IG bit: Individual address (unicast)
Type: IPv4 (0x0800)
> Internet Protocol Version 4, Src: 192.168.0.21, Dst: 128.119.245.12
> Transmission Control Protocol, Src Port: 56806, Dst Port: 80, Seq: 1, Ack: 1, Len: 430
> Hypertext Transfer Protocol
> GET /wireshark-labs/HTTP-ethereal-lab-file3.html HTTP/1.1\r\n
> [Expert Info (Chat/Sequence): GET /wireshark-labs/HTTP-ethereal-lab-file3.html HTTP/1.1\r\n]
Request Method: GET
Request URI: /wireshark-labs/HTTP-ethereal-lab-file3.html

Offset	Bytes	Display Name
0030	04 00 38 0a 00 00 47 45 54 20 2f 77 69 72 65 73	..8..GET /wires
0040	68 61 72 6b 2d 6c 61 62 73 2f 48 54 54 50 2d 65	hark-lab s/HTTP-e
0050	74 68 65 72 65 61 6c 2d 6c 61 62 2d 66 69 6c 65	thereal- lab-file
0060	33 2e 68 74 6d 6c 20 48 54 54 50 2f 31 2e 31 0d	3.html H TTP/1.1
0070	0a 41 63 63 65 70 74 3a 20 74 65 78 74 2f 68 74	Accept: text/ht

Bytes 54-56: Request Method (http.request.method) | Packets: 137 · Displayed: 137 (100.0%) · Dropped: 0 (0.0%) | Profile: Default

- After 54 bytes the ASCII “G” in “GET” appear in the Ethernet frame.

Next, answer the following questions, based on the contents of the Ethernet frame containing the first byte of the HTTP response message.

5. What is the value of the Ethernet source address? Is this the address of your computer, or of gaia.cs.umass.edu (Hint: the answer is no). What device has this as its Ethernet address?

*Ethernet

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

http

No.	Time	Source	Destination	Protocol	Length	Info
106	15:30:12.453866	192.168.0.21	128.119.245.12	HTTP	484	GET /wireshark-labs/HTTP-ethereal-lab-file3.htm
122	15:30:12.570605	128.119.245.12	192.168.0.21	HTTP	535	HTTP/1.1 200 OK (text/html)

< >

> Frame 122: 535 bytes on wire (4280 bits), 535 bytes captured (4280 bits) on interface 0

▼ Ethernet II, Src: Motorola_f5:79:b1 (88:b4:a6:f5:79:b1), Dst: AsustekC_91:eb:e8 (04:d4:c4:91:eb:e8)

▼ Destination: AsustekC_91:eb:e8 (04:d4:c4:91:eb:e8)

Address: AsustekC_91:eb:e8 (04:d4:c4:91:eb:e8)

.... .. = LG bit: Globally unique address (factory default)

.... .. = IG bit: Individual address (unicast)

▼ Source: Motorola_f5:79:b1 (88:b4:a6:f5:79:b1)

Address: Motorola_f5:79:b1 (88:b4:a6:f5:79:b1)

.... .. = LG bit: Globally unique address (factory default)

.... .. = IG bit: Individual address (unicast)

Type: IPv4 (0x0800)

> Internet Protocol Version 4, Src: 128.119.245.12, Dst: 192.168.0.21

> Transmission Control Protocol, Src Port: 80, Dst Port: 56806, Seq: 4381, Ack: 431, Len: 481

> [4 Reassembled TCP Segments (4861 bytes): #116(1460), #118(1460), #120(1460), #122(481)]

▼ Hypertext Transfer Protocol

▼ HTTP/1.1 200 OK\r\n

> [Expert Info (Chat/Sequence): HTTP/1.1 200 OK\r\n]

0000	04 d4 c4 91 eb e8 88 b4 a6 f5 79 b1 08 00 45 20y...E
0010	02 09 d9 3e 40 00 2f 06 3a 4f 80 77 f5 0c c0 a8	...>@./ :0-w...
0020	00 15 00 50 dd e6 6b 56 2a 12 67 94 6f 23 50 18	...P...kV *.g.o#P.
0030	00 ed 11 85 00 00 68 6d 65 6e 74 73 20 69 6e 66hm ents inf

Frame (535 bytes) Reassembled TCP (4861 bytes)

Source or Destination Hardware Address (eth.addr), 6 bytes

Packets: 137 · Displayed: 2 (1.5%) · Dropped: 0 (0.0%) Profile: Default

*Ethernet

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/> Expression...

No.	Time	Source	Destination	Protocol	Length	Info
118	15:30:12.570354	Motorola_f5:79:b1	AsustekC_91:eb:e8	0x0800	1514	IPv4
119	15:30:12.570362	AsustekC_91:eb:e8	Motorola_f5:79:b1	0x0800	54	IPv4
120	15:30:12.570436	Motorola_f5:79:b1	AsustekC_91:eb:e8	0x0800	1514	IPv4
121	15:30:12.570444	AsustekC_91:eb:e8	Motorola_f5:79:b1	0x0800	54	IPv4
122	15:30:12.570605	Motorola_f5:79:b1	AsustekC_91:eb:e8	0x0800	535	IPv4
123	15:30:12.570610	AsustekC_91:eb:e8	Motorola_f5:79:b1	0x0800	54	IPv4
124	15:30:12.578519	Motorola_f5:79:b1	AsustekC_91:eb:e8	0x0800	60	IPv4
125	15:30:12.596066	Motorola_f5:79:b1	AsustekC_91:eb:e8	0x0800	763	IPv4
126	15:30:12.596111	AsustekC_91:eb:e8	Motorola_f5:79:b1	0x0800	54	IPv4
127	15:30:12.596174	AsustekC_91:eb:e8	Motorola_f5:79:b1	0x0800	54	IPv4

> Frame 122: 535 bytes on wire (4280 bits), 535 bytes captured (4280 bits) on interface 0

▼ Ethernet II, Src: Motorola_f5:79:b1 (88:b4:a6:f5:79:b1), Dst: AsustekC_91:eb:e8 (04:d4:c4:91:eb:e8)

- ▼ Destination: AsustekC_91:eb:e8 (04:d4:c4:91:eb:e8)
Address: AsustekC_91:eb:e8 (04:d4:c4:91:eb:e8)
.... ..0. = LG bit: Globally unique address (factory default)
.... ..0 = IG bit: Individual address (unicast)
- ▼ Source: Motorola_f5:79:b1 (88:b4:a6:f5:79:b1)
Address: Motorola_f5:79:b1 (88:b4:a6:f5:79:b1)
.... ..0. = LG bit: Globally unique address (factory default)
.... ..0 = IG bit: Individual address (unicast)
Type: IPv4 (0x0800)
- ▼ Data (521 bytes)
Data: 45200209d93e40002f063a4f8077f50cc0a800150050dde6...
[Length: 521]

0000	04 d4 c4 91 eb e8 88 b4 a6 f5 79 b1 08 00 45 20y...E
0010	02 09 d9 3e 40 00 2f 06 3a 4f 80 77 f5 0c c0 a8	...>@./.:0.w....
0020	00 15 00 50 dd e6 6b 56 2a 12 67 94 6f 23 50 18	...P...kV *.g.o#P.
0030	00 ed 11 85 00 00 68 6d 65 6e 74 73 20 69 6e 66hm ents inf
0040	6c 69 63 74 65 64 2e 0a 0a 3c 2f 70 3e 3c 70 3e	licted.. </p><p>

Source or Destination Hardware Address (eth.addr), 6 bytes | Packets: 137 · Displayed: 137 (100.0%) · Dropped: 0 (0.0%) | Profile: Default

- The value of Ethernet source address is 88:b4:a6:f5:79:b1. This value is not the address of my computer nor of gaia.cs.umass.edu because it is the address of my Motorola router for getting on the subnet.

6. What is the destination address in the Ethernet frame? Is this the Ethernet address of your computer?

*Ethernet

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/> Expression...

No.	Time	Source	Destination	Protocol	Length	Info
118	15:30:12.570354	Motorola_f5:79:b1	AsustekC_91:eb:e8	0x0800	1514	IPv4
119	15:30:12.570362	AsustekC_91:eb:e8	Motorola_f5:79:b1	0x0800	54	IPv4
120	15:30:12.570436	Motorola_f5:79:b1	AsustekC_91:eb:e8	0x0800	1514	IPv4
121	15:30:12.570444	AsustekC_91:eb:e8	Motorola_f5:79:b1	0x0800	54	IPv4
122	15:30:12.570605	Motorola_f5:79:b1	AsustekC_91:eb:e8	0x0800	535	IPv4
123	15:30:12.570610	AsustekC_91:eb:e8	Motorola_f5:79:b1	0x0800	54	IPv4
124	15:30:12.578519	Motorola_f5:79:b1	AsustekC_91:eb:e8	0x0800	60	IPv4
125	15:30:12.596066	Motorola_f5:79:b1	AsustekC_91:eb:e8	0x0800	763	IPv4
126	15:30:12.596111	AsustekC_91:eb:e8	Motorola_f5:79:b1	0x0800	54	IPv4
127	15:30:12.596174	AsustekC_91:eb:e8	Motorola_f5:79:b1	0x0800	54	IPv4

> Frame 122: 535 bytes on wire (4280 bits), 535 bytes captured (4280 bits) on interface 0

▼ Ethernet II, Src: Motorola_f5:79:b1 (88:b4:a6:f5:79:b1), Dst: AsustekC_91:eb:e8 (04:d4:c4:91:eb:e8)

▼ Destination: AsustekC_91:eb:e8 (04:d4:c4:91:eb:e8)

Address: AsustekC_91:eb:e8 (04:d4:c4:91:eb:e8)

.... .. = LG bit: Globally unique address (factory default)

.... .. = IG bit: Individual address (unicast)

▼ Source: Motorola_f5:79:b1 (88:b4:a6:f5:79:b1)

Address: Motorola_f5:79:b1 (88:b4:a6:f5:79:b1)

.... .. = LG bit: Globally unique address (factory default)

.... .. = IG bit: Individual address (unicast)

Type: IPv4 (0x0800)

▼ Data (521 bytes)

Data: 45200209d93e40002f063a4f8077f50cc0a800150050dde6...

[Length: 521]

0000	04 d4 c4 91 eb e8 88 b4 a6 f5 79 b1 08 00 45 20 -y...E
0010	02 09 d9 3e 40 00 2f 06 3a 4f 80 77 f5 0c c0 a8	...>@./.:0-w...
0020	00 15 00 50 dd e6 6b 56 2a 12 67 94 6f 23 50 18	...P..kV *.g.o#P.
0030	00 ed 11 85 00 00 68 6d 65 6e 74 73 20 69 6e 66hm ents inf
0040	6c 69 63 74 65 64 2e 0a 0a 3c 2f 70 3e 3c 70 3e	licted.. </p><p>

Source or Destination Hardware Address (eth.addr), 6 bytes

Packets: 137 • Displayed: 137 (100.0%) • Dropped: 0 (0.0%)

Profile: Default

- The destination address is 04:d4:c4:91:eb:e8. This is the address of my computer which manufactured by AsustekC

7. Give the hexadecimal value for the two-byte Frame type field. What upper layer protocol does this correspond to?

*Ethernet

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/> Expression...

No.	Time	Source	Destination	Protocol	Length	Info
118	15:30:12.570354	Motorola_f5:79:b1	AsustekC_91:eb:e8	0x0800	1514	IPv4
119	15:30:12.570362	AsustekC_91:eb:e8	Motorola_f5:79:b1	0x0800	54	IPv4
120	15:30:12.570436	Motorola_f5:79:b1	AsustekC_91:eb:e8	0x0800	1514	IPv4
121	15:30:12.570444	AsustekC_91:eb:e8	Motorola_f5:79:b1	0x0800	54	IPv4
122	15:30:12.570605	Motorola_f5:79:b1	AsustekC_91:eb:e8	0x0800	535	IPv4
123	15:30:12.570610	AsustekC_91:eb:e8	Motorola_f5:79:b1	0x0800	54	IPv4
124	15:30:12.578519	Motorola_f5:79:b1	AsustekC_91:eb:e8	0x0800	60	IPv4
125	15:30:12.596066	Motorola_f5:79:b1	AsustekC_91:eb:e8	0x0800	763	IPv4
126	15:30:12.596111	AsustekC_91:eb:e8	Motorola_f5:79:b1	0x0800	54	IPv4
127	15:30:12.596174	AsustekC_91:eb:e8	Motorola_f5:79:b1	0x0800	54	IPv4

> Frame 122: 535 bytes on wire (4280 bits), 535 bytes captured (4280 bits) on interface 0

▼ Ethernet II, Src: Motorola_f5:79:b1 (88:b4:a6:f5:79:b1), Dst: AsustekC_91:eb:e8 (04:d4:c4:91:eb:e8)

- ▼ Destination: AsustekC_91:eb:e8 (04:d4:c4:91:eb:e8)
Address: AsustekC_91:eb:e8 (04:d4:c4:91:eb:e8)
.... ..0. = LG bit: Globally unique address (factory default)
.... ..0 = IG bit: Individual address (unicast)
- ▼ Source: Motorola_f5:79:b1 (88:b4:a6:f5:79:b1)
Address: Motorola_f5:79:b1 (88:b4:a6:f5:79:b1)
.... ..0. = LG bit: Globally unique address (factory default)
.... ..0 = IG bit: Individual address (unicast)
- Type: IPv4 (0x0800)
- ▼ Data (521 bytes)
Data: 45200209d93e40002f063a4f8077f50cc0a800150050dde6...
[Length: 521]

0000	04 d4 c4 91 eb e8 88 b4 a6 f5 79 b1 08 00 45 20y...E
0010	02 09 d9 3e 40 00 2f 06 3a 4f 80 77 f5 0c c0 a8	...>@./.:0:w...
0020	00 15 00 50 dd e6 6b 56 2a 12 67 94 6f 23 50 18	...P..kV *.g.o#P
0030	00 ed 11 85 00 00 68 6d 65 6e 74 73 20 69 6e 66hm ents inf
0040	6c 69 63 74 65 64 2e 0a 0a 3c 2f 70 3e 3c 70 3e	llicted.. </p><p>

Type (eth.type), 2 bytes

Packets: 137 · Displayed: 137 (100.0%) · Dropped: 0 (0.0%) | Profile: Default

- The hexadecimal value for the two-byte Frame type field is 0x0800. This value corresponds to the IP protocol like http request packet (Question #3).

8. How many bytes from the very start of the Ethernet frame does the ASCII “O” in “OK” (i.e., the HTTP response code) appear in the Ethernet frame?

Wireshark packet capture showing an HTTP 200 OK response. The packet list shows packet 122 as the final segment of a reassembled TCP connection. The packet details pane shows the HTTP response structure, including status code 200 and response phrase 'OK'. The packet bytes pane shows the raw data, with the ASCII 'O' in 'OK' highlighted at offset 13.

No.	Time	Source	Destination	Protocol	Length	Info
113	15:30:12.496829	192.168.0.21	40.117.135.216	TLSv1...	1872	Application Data
114	15:30:12.560542	128.119.245.12	192.168.0.21	TCP	60	80 → 56806 [ACK] Seq=1 Ack=431 Win=30336
115	15:30:12.560952	128.119.245.12	192.168.0.21	TCP	60	[TCP Dup ACK 114#1] 80 → 56806 [ACK] Seq=
116	15:30:12.570274	128.119.245.12	192.168.0.21	TCP	1514	80 → 56806 [ACK] Seq=1 Ack=431 Win=30336
117	15:30:12.570295	192.168.0.21	128.119.245.12	TCP	54	56806 → 80 [ACK] Seq=431 Ack=1461 Win=262
118	15:30:12.570354	128.119.245.12	192.168.0.21	TCP	1514	80 → 56806 [ACK] Seq=1461 Ack=431 Win=303
119	15:30:12.570362	192.168.0.21	128.119.245.12	TCP	54	56806 → 80 [ACK] Seq=431 Ack=2921 Win=262
120	15:30:12.570436	128.119.245.12	192.168.0.21	TCP	1514	80 → 56806 [ACK] Seq=2921 Ack=431 Win=303
121	15:30:12.570444	192.168.0.21	128.119.245.12	TCP	54	56806 → 80 [ACK] Seq=431 Ack=4381 Win=262
122	15:30:12.570605	128.119.245.12	192.168.0.21	HTTP	535	HTTP/1.1 200 OK (text/html)

Transmission Control Protocol, Src Port: 80, Dst Port: 56806, Seq: 4381, Ack: 431, Len: 481
 [4 Reassembled TCP Segments (4861 bytes): #116(1460), #118(1460), #120(1460), #122(481)]
 Hypertext Transfer Protocol
 HTTP/1.1 200 OK\r\n
 [Expert Info (Chat/Sequence): HTTP/1.1 200 OK\r\n]
 Response Version: HTTP/1.1
 Status Code: 200
 [Status Code Description: OK]
 Response Phrase: OK
 Date: Sat, 30 Nov 2019 23:30:12 GMT\r\n
 Server: Apache/2.4.6 (CentOS) OpenSSL/1.0.2k-fips PHP/5.4.16 mod_perl/2.0.11 Perl/v5.16.3\r\n
 Last-Modified: Sat, 30 Nov 2019 06:59:02 GMT\r\n
 ETag: "1194-5988ae1acad69"\r\n
 Accept-Ranges: bytes\r\n
 Content-Length: 4500\r\n
 Keep-Alive: timeout=5, max=100\r\n
 Connection: Keep-Alive\r\n

0000 48 54 54 50 2f 31 2e 31 20 32 30 30 20 4f 4b 0d HTTP/1.1 200 OK
 0010 0a 44 61 74 65 3a 20 53 61 74 2c 20 33 30 20 4e Date: S at, 30 N
 0020 6f 76 20 32 30 31 39 20 32 33 3a 33 30 3a 31 32 ov 2019 23:30:12
 0030 20 47 4d 54 0d 0a 53 65 72 76 65 72 3a 20 41 70 GMT Se rver: Ap

Frame (535 bytes) Reassembled TCP (4861 bytes)
 Bytes 13-14: Response Phrase (http.response.phrase) | Packets: 137 · Displayed: 137 (100.0%) · Dropped: 0 (0.0%) | Profile: Default

- After 13 bytes, the ASCII “O” in “OK” appears in the Ethernet frame of HTTP response packet.

The Windows arp command with no arguments will display the contents of the ARP cache on your computer. Run the arp command.

9. Write down the contents of your computer’s ARP cache. What is the meaning of each column value?

```
Administrator: Command Prompt
C:\WINDOWS\system32>arp -a

Interface: 192.168.0.21 --- 0x9
Internet Address      Physical Address      Type
192.168.0.1           88-b4-a6-f5-79-b1    dynamic
192.168.0.13          c4-1c-ff-b3-b0-3b    dynamic
192.168.0.17          b8-31-b5-f7-88-8a    dynamic
192.168.0.24          d4-f5-47-49-ea-f1    dynamic
192.168.0.255         ff-ff-ff-ff-ff-ff    static
224.0.0.2             01-00-5e-00-00-02    static
224.0.0.22            01-00-5e-00-00-16    static
224.0.0.251           01-00-5e-00-00-fb    static
224.0.0.252           01-00-5e-00-00-fc    static
230.0.0.1             01-00-5e-00-00-01    static
239.255.255.250       01-00-5e-7f-ff-fa    static
255.255.255.255       ff-ff-ff-ff-ff-ff    static

Interface: 169.254.150.244 --- 0x11
Internet Address      Physical Address      Type
169.254.255.255       ff-ff-ff-ff-ff-ff    static
224.0.0.2             01-00-5e-00-00-02    static
224.0.0.22            01-00-5e-00-00-16    static
224.0.0.251           01-00-5e-00-00-fb    static
224.0.0.252           01-00-5e-00-00-fc    static
230.0.0.1             01-00-5e-00-00-01    static
239.255.255.250       01-00-5e-7f-ff-fa    static
255.255.255.255       ff-ff-ff-ff-ff-ff    static

C:\WINDOWS\system32>
```

- At first, my desktop computer interface is 192.168.0.21. The Internet Address column means the IP address of each devices. The Physical Address column means the MAC address of each devices. The Type column shows that the protocol type of each devices. For example, my router's IP is 192.168.0.1, MAC address is 88-b4-a6-f5-79-b1, and the protocol type is dynamic.

10. What are the hexadecimal values for the source and destination addresses in the Ethernet frame containing the ARP request message?

*Ethernet

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

arp

No.	Time	Source	Destination	Protocol	Length	Info
833	19:39:50.525329	SeikoEps_73:be:53	Broadcast	ARP	60	Gratuitous ARP for 192.168.0.16 (Request)
1010	19:40:06.159781	AsustekC_91:eb:e8	Motorola_f5:79:b1	ARP	42	Who has 192.168.0.1? Tell 192.168.0.21
1011	19:40:06.160045	Motorola_f5:79:b1	AsustekC_91:eb:e8	ARP	60	192.168.0.1 is at 88:b4:a6:f5:79:b1

< >

> Frame 833: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0

▼ Ethernet II, Src: SeikoEps_73:be:53 (38:9d:92:73:be:53), Dst: Broadcast (ff:ff:ff:ff:ff:ff)

▼ Destination: Broadcast (ff:ff:ff:ff:ff:ff)

Address: Broadcast (ff:ff:ff:ff:ff:ff)

....1. = LG bit: Locally administered address (this is NOT the factory default)

....1. = IG bit: Group address (multicast/broadcast)

▼ Source: SeikoEps_73:be:53 (38:9d:92:73:be:53)

Address: SeikoEps_73:be:53 (38:9d:92:73:be:53)

....0. = LG bit: Globally unique address (factory default)

....0. = IG bit: Individual address (unicast)

Type: ARP (0x0806)

Padding: 00000100000000000000000508c6002508c6002

▼ Address Resolution Protocol (request/gratuitous ARP)

Hardware type: Ethernet (1)

Protocol type: IPv4 (0x0800)

Hardware size: 6

Protocol size: 4

```

0000  ff ff ff ff ff ff 38 9d 92 73 be 53 08 06 00 01  ....8. .s.S....
0010  08 00 06 04 00 01 38 9d 92 73 be 53 c0 a8 00 10  ....8. .s.S....
0020  00 00 00 00 00 00 c0 a8 00 10 00 00 01 00 00 00  .........
0030  00 00 00 00 50 8c 60 02 50 8c 60 02  ....P..P..

```

Source or Destination Hardware Address (eth.addr), 6 bytes

Packets: 1118 · Displayed: 3 (0.3%) · Dropped: 0 (0.0%) | Profile: Default

- The source address is 38:9d:92:73:be:53 and the destination address is ff:ff:ff:ff:ff:ff in the Ethernet frame of ARP request message.

11. Give the hexadecimal value for the two-byte Ethernet Frame type field. What upper layer protocol does this correspond to?

***Ethernet**

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

arp

No.	Time	Source	Destination	Protocol	Length	Info
833	19:39:50.525329	SeikoEps_73:be:53	Broadcast	ARP	60	Gratuitous ARP for 192.168.0.16 (Request)
1010	19:40:06.159781	AsustekC_91:eb:e8	Motorola_f5:79:b1	ARP	42	Who has 192.168.0.1? Tell 192.168.0.21
1011	19:40:06.160045	Motorola_f5:79:b1	AsustekC_91:eb:e8	ARP	60	192.168.0.1 is at 88:b4:a6:f5:79:b1

< >

> Frame 833: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0

▼ Ethernet II, Src: SeikoEps_73:be:53 (38:9d:92:73:be:53), Dst: Broadcast (ff:ff:ff:ff:ff:ff)

- ▼ Destination: Broadcast (ff:ff:ff:ff:ff:ff)
 - Address: Broadcast (ff:ff:ff:ff:ff:ff)
 -1. = LG bit: Locally administered address (this is NOT the factory default)
 -1. = IG bit: Group address (multicast/broadcast)
- ▼ Source: SeikoEps_73:be:53 (38:9d:92:73:be:53)
 - Address: SeikoEps_73:be:53 (38:9d:92:73:be:53)
 -0. = LG bit: Globally unique address (factory default)
 -0. = IG bit: Individual address (unicast)
 - Type: ARP (0x0806)
 - Padding: 000001000000000000000508c6002508c6002
- ▼ Address Resolution Protocol (request/gratuitous ARP)
 - Hardware type: Ethernet (1)
 - Protocol type: IPv4 (0x0800)
 - Hardware size: 6
 - Protocol size: 4

```

0000  ff ff ff ff ff ff 38 9d 92 73 be 53 08 06 00 01  ....8. .s.S....
0010  08 00 06 04 00 01 38 9d 92 73 be 53 c0 a8 00 10  ....8. .s.S....
0020  00 00 00 00 00 00 c0 a8 00 10 00 00 01 00 00 00  ....P...P...
0030  00 00 00 00 50 8c 60 02 50 8c 60 02  ....P...P...
  
```

Type (eth.type), 2 bytes | Packets: 1118 · Displayed: 3 (0.3%) · Dropped: 0 (0.0%) | Profile: Default

- The hexadecimal value for two-byte Ethernet Frame type field is 0x0806. This value corresponds to the ARP protocol.

12. Download the ARP specification from <ftp://ftp.rfc-editor.org/in-notes/std/std37.txt>. A readable, detailed discussion of ARP is also at <http://www.erg.abdn.ac.uk/users/gorry/course/inet-pages/arp.html>.

a) How many bytes from the very beginning of the Ethernet frame does the ARP opcode field begin?

*Ethernet

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

arp

No.	Time	Source	Destination	Protocol	Length	Info
833	19:39:50.525329	SeikoEps_73:be:53	Broadcast	ARP	60	Gratuitous ARP for 192.168.0.16 (Request)
1010	19:40:06.159781	AsustekC_91:eb:e8	Motorola_f5:79:b1	ARP	42	Who has 192.168.0.1? Tell 192.168.0.21
1011	19:40:06.160045	Motorola_f5:79:b1	AsustekC_91:eb:e8	ARP	60	192.168.0.1 is at 88:b4:a6:f5:79:b1

Address: SeikoEps_73:be:53 (38:9d:92:73:be:53)
0. = LG bit: Globally unique address (factory default)
0. = IG bit: Individual address (unicast)
 Type: ARP (0x0806)
 Padding: 0000010000000000000000508c6002508c6002
 Address Resolution Protocol (request/gratuitous ARP)
 Hardware type: Ethernet (1)
 Protocol type: IPv4 (0x0800)
 Hardware size: 6
 Protocol size: 4
 Opcode: request (1)
 [Is gratuitous: True]
 Sender MAC address: SeikoEps_73:be:53 (38:9d:92:73:be:53)
 Sender IP address: 192.168.0.16
 Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
 Target IP address: 192.168.0.16

```

0000  ff ff ff ff ff ff 38 9d 92 73 be 53 08 06 00 01  ....8. .s.S...
0010  08 00 06 04 00 01 38 9d 92 73 be 53 c0 a8 00 10  ....8. .s.S...
0020  00 00 00 00 00 00 c0 a8 00 10 00 00 01 00 00 00  ....P.. P..
0030  00 00 00 00 50 8c 60 02 50 8c 60 02  ....P.. P..
  
```

Bytes 20-21: Opcode (arp.opcode) | Packets: 1118 · Displayed: 3 (0.3%) · Dropped: 0 (0.0%) | Profile: Default

- The ARP Opcode field begins after 20 bytes from the very beginning of the Ethernet frame.

b) What is the value of the opcode field within the ARP-payload part of the Ethernet frame in which an ARP request is made?

- The value of the opcode field within the ARP-payload part of the Ethernet frame is 0x0001 for ARP request.

c) Does the ARP message contain the IP address of the sender?

- Yes, the IP address of the sender is 192.168.0.16

d) Where in the ARP request does the “question” appear – the Ethernet address of the machine whose corresponding IP address is being queried?

*Ethernet

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

arp

No.	Time	Source	Destination	Protocol	Length	Info
833	19:39:50.525329	SeikoEps_73:be:53	Broadcast	ARP	60	Gratuitous ARP for 192.168.0.16 (Request)
1010	19:40:06.159781	AsustekC_91:eb:e8	Motorola_f5:79:b1	ARP	42	Who has 192.168.0.1? Tell 192.168.0.21
1011	19:40:06.160045	Motorola_f5:79:b1	AsustekC_91:eb:e8	ARP	60	192.168.0.1 is at 88:b4:a6:f5:79:b1

.... ..0 = IG bit: Individual address (unicast)
 Source: AsustekC_91:eb:e8 (04:d4:c4:91:eb:e8)
 Address: AsustekC_91:eb:e8 (04:d4:c4:91:eb:e8)
0 = LG bit: Globally unique address (factory default)
0 = IG bit: Individual address (unicast)
 Type: ARP (0x0806)
 Address Resolution Protocol (request)
 Hardware type: Ethernet (1)
 Protocol type: IPv4 (0x0800)
 Hardware size: 6
 Protocol size: 4
 Opcode: request (1)
 Sender MAC address: AsustekC_91:eb:e8 (04:d4:c4:91:eb:e8)
 Sender IP address: 192.168.0.21
 Target MAC address: Motorola_f5:79:b1 (88:b4:a6:f5:79:b1)
 Target IP address: 192.168.0.1

0000 88 b4 a6 f5 79 b1 04 d4 c4 91 eb e8 08 06 00 01y:.. ..
 0010 08 00 06 04 00 01 04 d4 c4 91 eb e8 c0 a8 00 15
 0020 88 b4 a6 f5 79 b1 c0 a8 00 01y... ..

Target IP address (arp.dst.proto_ipv4), 4 bytes

Packets: 1118 · Displayed: 3 (0.3%) · Dropped: 0 (0.0%) Profile: Default

- The question “Who has 192.168.0.1?” appears in the target MAC address 88:b4:a6:f5:79:b1 which is the destination as my router.

13. Now find the ARP reply that was sent in response to the ARP request.

a) How many bytes from the very beginning of the Ethernet frame does the ARP opcode field begin?

14. What are the hexadecimal values for the source and destination addresses in the Ethernet frame containing the ARP reply message?

The screenshot shows a Wireshark capture of ARP traffic. The packet list at the top shows three packets:

No.	Time	Source	Destination	Protocol	Length	Info
833	19:39:50.525329	SeikoEps_73:be:53	Broadcast	ARP	60	Gratuitous ARP for 192.168.0.16 (Request)
1010	19:40:06.159781	AsustekC_91:eb:e8	Motorola_f5:79:b1	ARP	42	Who has 192.168.0.1? Tell 192.168.0.21
1011	19:40:06.160045	Motorola_f5:79:b1	AsustekC_91:eb:e8	ARP	60	192.168.0.1 is at 88:b4:a6:f5:79:b1

The packet details for packet 1011 (ARP reply) are expanded, showing the Ethernet II header with source address 88:b4:a6:f5:79:b1 and destination address 04:d4:c4:91:eb:e8. The ARP payload shows it is a reply for 192.168.0.1.

- The hexadecimal value for source address is 88:b4:a6:f5:79:b1 and for destination address is 04:d4:c4:91:eb:e8

15. Open the ethernet-ethereal-trace-1 trace file in <http://gaia.cs.umass.edu/wireshark-labs/wireshark-traces.zip>. The first and second ARP packets in this trace correspond to an ARP request sent by the computer running Wireshark, and the ARP reply sent to the computer running Wireshark by the computer with the ARP-requested Ethernet address. But there is yet another computer on this network, as indicated by packet 6 – another ARP request. Why is there no ARP reply (sent in response to the ARP request in packet 6) in the packet trace?

- Since we are not at the machine that sent the packet 6's original ARP request (different source between packet 1 and 6), we can't see the reply. We can only see the ARP reply from our machine's ARP request because it sent to our machine directly.

Extra Credit

EX-1. The arp command:

`arp -s InetAddr EtherAddr`

allows you to manually add an entry to the ARP cache that resolves the IP address InetAddr to the physical address EtherAddr. What would happen if, when you manually added an entry, you entered the correct IP address, but the wrong Ethernet address for that remote interface?

- At first, ARP cache two kind of entries which are static and dynamic. The ARP command will assign the IP address dynamically which is static entry if I manually add an entry to the ARP cache with correct IP address. However, since I entered the wrong Ethernet address, full packet loss will be happened during data transmission.

EX-2. What is the default amount of time that an entry remains in your ARP cache before being removed. You can determine this empirically (by monitoring the cache contents) or by looking this up in your operation system documentation. Indicate how/where you determined this value.

```
Administrator: Command Prompt

169.254.255.255      ff-ff-ff-ff-ff-ff      static
224.0.0.2           01-00-5e-00-00-02      static
224.0.0.22          01-00-5e-00-00-16      static
224.0.0.251         01-00-5e-00-00-fb      static
224.0.0.252         01-00-5e-00-00-fc      static
230.0.0.1           01-00-5e-00-00-01      static
239.255.255.250     01-00-5e-7f-ff-fa      static
255.255.255.255     ff-ff-ff-ff-ff-ff      static

C:\WINDOWS\system32>interface ipv4 show
'interface' is not recognized as an internal or external command,
operable program or batch file.

C:\WINDOWS\system32>netsh interface ipv4 show interfaces

Idx  Met      MTU      State      Name
-----
1     75  4294967295 connected  Loopback Pseudo-Interface 1
8     65    1500 disconnected Bluetooth Network Connection
16    25    1500 disconnected Wi-Fi
9     25    1500 connected  Ethernet
4     25    1500 disconnected Local Area Connection* 9
5     25    1500 disconnected Local Area Connection* 10
17    25    1500 connected  Npcap Loopback Adapter

C:\WINDOWS\system32>netsh interface ipv4 show interface 9

Interface Ethernet Parameters
-----
```

```
Administrator: Command Prompt

Interface Ethernet Parameters
-----
IfLuid                : ethernet_32769
IfIndex               : 9
State                 : connected
Metric                : 25
Link MTU              : 1500 bytes
Reachable Time        : 27000 ms
Base Reachable Time   : 30000 ms
Retransmission Interval : 1000 ms
DAD Transmits         : 3
Site Prefix Length    : 64
Site Id               : 1
Forwarding            : disabled
Advertising           : disabled
Neighbor Discovery     : enabled
Neighbor Unreachability Detection : enabled
Router Discovery      : dhcp
Managed Address Configuration : enabled
Other Stateful Configuration : enabled
Weak Host Sends       : disabled
Weak Host Receives    : disabled
Use Automatic Metric  : enabled
Ignore Default Routes : disabled
Advertised Router Lifetime : 1800 seconds
Advertise Default Route : disabled
Current Hop Limit     : 0
Force ARPND Wake up patterns : disabled
Directed MAC Wake up patterns : disabled
```

- Since my computer Ethernet interface shows that the default amount of time that an entry remains in my ARP cache before being removed is 27 seconds (27000 ms).

extra credit #2 – Socket programming

Programming language: C

Explanation: I used MobaXterm terminal by creating another process for listening on local host.

1. server.c

```

C server.c  X
C: > Users > 15419 > AppData > Local > Temp > Mxt121 > RemoteFiles > 132778_7_6 > C server.c
1  //written by Junhyeok Jeong
2
3  // server code for UDP socket programming
4  #include <arpa/inet.h>
5  #include <netinet/in.h>
6  #include <stdio.h>
7  #include <stdlib.h>
8  #include <string.h>
9  #include <sys/socket.h>
10 #include <sys/types.h>
11 #include <unistd.h>
12
13 #define IP_PROTOCOL 0
14 #define PORT_NO 5000
15 #define NET_BUF_SIZE 32
16 #define cipherKey 'S'
17 #define sendrecvflag 0
18 #define nofile "File Not Found!"
19
20 // function to clear buffer
21 void clearBuf(char* b)
22 {
23     int i;
24     for (i = 0; i < NET_BUF_SIZE; i++)
25         b[i] = '\0';
26 }
27
28 // function to encrypt
29 char Cipher(char ch)
30 {
31     return ch ^ cipherKey;
32 }
33
34 // function sending file
35 int sendFile(FILE* fp, char* buf, int s)
36 {
37     int i, len;
38     if (fp == NULL) {
39         strcpy(buf, nofile);
40         len = strlen(nofile);
41         buf[len] = EOF;
42         for (i = 0; i <= len; i++)
43             buf[i] = Cipher(buf[i]);
44         return 1;
45     }
46
47     char ch, ch2;
48     for (i = 0; i < s; i++) {
49         ch = fgetc(fp);

```

```

47     char ch, ch2;
48     for (i = 0; i < s; i++) {
49         ch = fgetc(fp);
50         ch2 = Cipher(ch);
51         buf[i] = ch2;
52         if (ch == EOF)
53             return 1;
54     }
55     return 0;
56 }
57
58 // driver code
59 int main()
60 {
61     int sockfd, nBytes;
62     struct sockaddr_in addr_con;
63     int addrlen = sizeof(addr_con);
64     addr_con.sin_family = AF_INET;
65     addr_con.sin_port = htons(PORT_NO);
66     addr_con.sin_addr.s_addr = INADDR_ANY;
67     char net_buf[NET_BUF_SIZE];
68     FILE* fp;
69
70     // socket()
71     sockfd = socket(AF_INET, SOCK_DGRAM, IP_PROTOCOL);
72
73     if (sockfd < 0)
74         printf("\nError: Check your socket!!\n");
75     else
76         printf("\nfile name will be received on sockfd %d \n", sockfd);
77
78     // bind()
79     if (bind(sockfd, (struct sockaddr*)&addr_con, sizeof(addr_con)) == 0)
80         printf("\nSuccessfully binded!\n");
81     else
82         printf("\nBinding Failed!\n");
83
84     while (1) {
85         printf("\nlistening client connection and Waiting for file name...\n");
86
87         // receive file name
88         clearBuf(net_buf);
89
90         nBytes = recvfrom(sockfd, net_buf,
91                         NET_BUF_SIZE, 0,
92                         (struct sockaddr*)&addr_con, &addrlen);
93     }

```

```

// receive file name
clearBuf(net_buf);

nBytes = recvfrom(sockfd, net_buf,
                  NET_BUF_SIZE, sendrecvflag,
                  (struct sockaddr*)&addr_con, &addrlen);

fp = fopen(net_buf, "r");
printf("\nFile Name Received: %s\n", net_buf);
if (fp == NULL)
    printf("\nFile open failed!\n");
else
    printf("\nFile Successfully opened!\n");

while (1) {
    // process
    if (sendFile(fp, net_buf, NET_BUF_SIZE)) {
        sendto(sockfd, net_buf, NET_BUF_SIZE,
                sendrecvflag,
                (struct sockaddr*)&addr_con, addrlen);
        break;
    }

    // send
    sendto(sockfd, net_buf, NET_BUF_SIZE,
            sendrecvflag,
            (struct sockaddr*)&addr_con, addrlen);
    clearBuf(net_buf);
}
if (fp != NULL)
    fclose(fp);
}
return 0;
}

```

2. client.c

C server.c C client.c X
C: > Users > 15419 > AppData > Local > Temp > Mxt121 > RemoteFiles > 132778_7_7 > C client.c > main()

```
1 //written by Junhyeok Jeong
2
3 // client code for UDP socket programming
4 #include <arpa/inet.h>
5 #include <netinet/in.h>
6 #include <stdio.h>
7 #include <stdlib.h>
8 #include <string.h>
9 #include <sys/socket.h>
10 #include <sys/types.h>
11 #include <unistd.h>
12
13 #define IP_PROTOCOL 0
14 #define IP_ADDRESS "127.0.0.1" // localhost
15 #define PORT_NO 5000
16 #define NET_BUF_SIZE 32
17 #define cipherKey 'S'
18 #define sendrecvflag 0
19
20 // function to clear buffer
21 void clearBuf(char* b)
22 {
23     int i;
24     for (i = 0; i < NET_BUF_SIZE; i++)
25         b[i] = '\0';
26 }
27
28 // function for decryption
29 char Cipher(char ch)
30 {
31     return ch ^ cipherKey;
32 }
33
34 // function to receive file
35 int rcvFile(char* buf, int s, FILE *write_file)
36 {
37     int i;
38     char ch;
39
40     for (i = 0; i < s; i++) {
41         ch = buf[i];
42         ch = Cipher(ch);
43         if (ch == EOF)
44             return 1;
45         else
46             fprintf(write_file, "%c", ch);
47         printf("%c", ch);
```



```

52 // driver code
53 int main()
54 {
55     int sockfd, nBytes;
56     struct sockaddr_in addr_con;
57     int addrlen = sizeof(addr_con);
58     addr_con.sin_family = AF_INET;
59     addr_con.sin_port = htons(PORT_NO);
60     addr_con.sin_addr.s_addr = inet_addr(IP_ADDRESS);
61     char net_buf[NET_BUF_SIZE];
62     FILE *write_file;
63
64     // socket()
65     sockfd = socket(AF_INET, SOCK_DGRAM,
66                     IP_PROTOCOL);
67
68     if (sockfd < 0)
69         printf("\nError: check your socket!!\n");
70     else
71         printf("\nfile will be received on sockfd %d\n", sockfd);
72
73     while (1) {
74         printf("\nPlease enter file name to receive:\n");
75         scanf("%s", net_buf);
76         write_file = fopen(net_buf, "w");
77
78         sendto(sockfd, net_buf, NET_BUF_SIZE,
79                sendrecvflag, (struct sockaddr*)&addr_con,
80                addrlen);
81
82         printf("\n-----Data Received-----\n");
83
84         while (1) {
85             // receive
86             clearBuf(net_buf);
87             nBytes = recvfrom(sockfd, net_buf, NET_BUF_SIZE,
88                               sendrecvflag, (struct sockaddr*)&addr_con,
89                               &addrlen);
90
91             // process
92             if (recvFile(net_buf, NET_BUF_SIZE, write_file)) {
93                 break;
94             }
95         }
96         printf("\n-----\n");

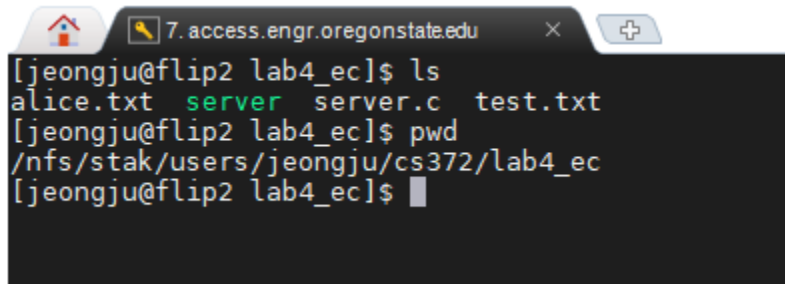
```

```

98         printf("\nthe requested file is received and stored successfully! And socket connection is end\n");
99         break;
100     }
101
102     return 0;
103 }
104
105

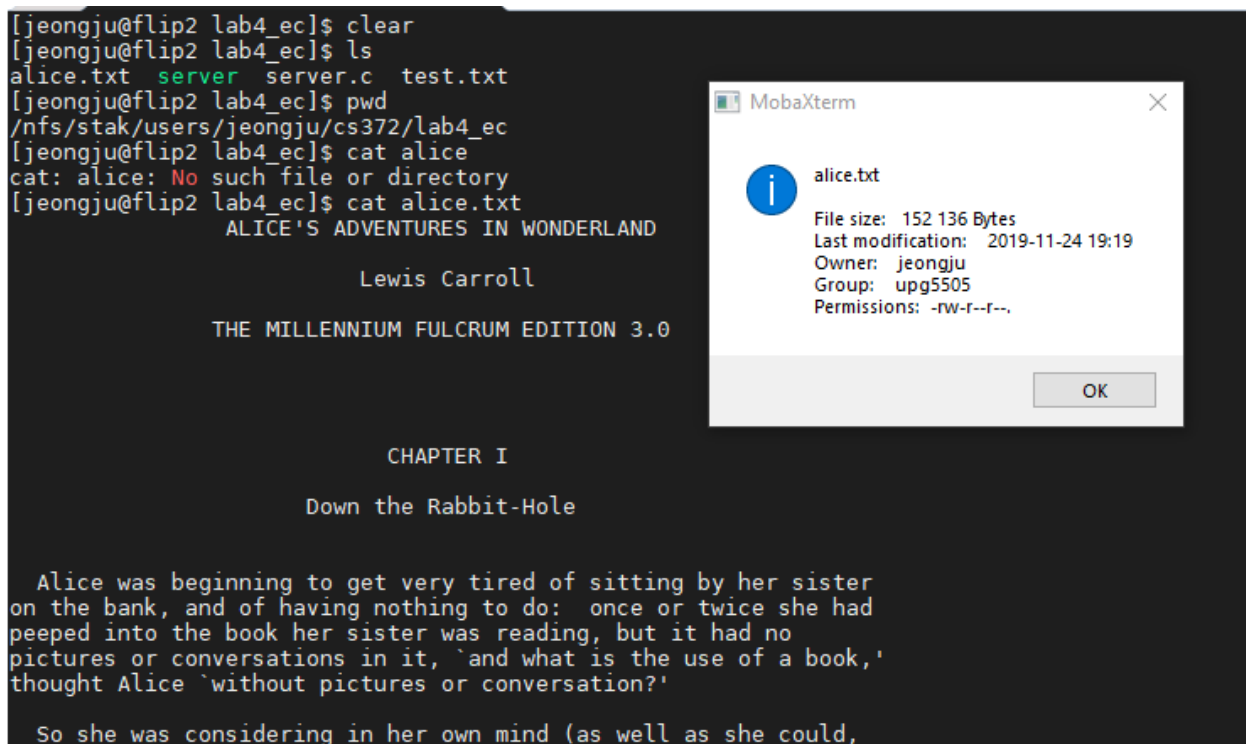
```

3. Server.c is in different directory



```
[jeongju@flip2 lab4_ec]$ ls
alice.txt  server  server.c  test.txt
[jeongju@flip2 lab4_ec]$ pwd
/nfs/stak/users/jeongju/cs372/lab4_ec
[jeongju@flip2 lab4_ec]$
```

4. Test file: alice.txt (from lab3)



```
[jeongju@flip2 lab4_ec]$ clear
[jeongju@flip2 lab4_ec]$ ls
alice.txt  server  server.c  test.txt
[jeongju@flip2 lab4_ec]$ pwd
/nfs/stak/users/jeongju/cs372/lab4_ec
[jeongju@flip2 lab4_ec]$ cat alice
cat: alice: No such file or directory
[jeongju@flip2 lab4_ec]$ cat alice.txt
ALICE'S ADVENTURES IN WONDERLAND

Lewis Carroll

THE MILLENNIUM FULCRUM EDITION 3.0

CHAPTER I

Down the Rabbit-Hole

Alice was beginning to get very tired of sitting by her sister
on the bank, and of having nothing to do: once or twice she had
peeped into the book her sister was reading, but it had no
pictures or conversations in it, 'and what is the use of a book,'
thought Alice 'without pictures or conversation?'

So she was considering in her own mind (as well as she could,
```

MobaXterm

i alice.txt

File size: 152 136 Bytes
Last modification: 2019-11-24 19:19
Owner: jeongju
Group: upg5505
Permissions: -rw-r--r--

OK

5. Compile server.c on child process for listening

```

[jeongju@flip2 lab4_ec]$ ps aux | grep jeongju
root      23445  0.0  0.0 183720  6136 ?        Ss   19:23   0:00 sshd: jeongju [priv]
jeongju   29186  0.0  0.0 165628  3072 pts/111  R+   19:53   0:00 ps aux
jeongju   29187  0.0  0.0 112716   964 pts/111  S+   19:53   0:00 grep --color=auto jeongju
jeongju   32672  0.0  0.0 184036  3096 ?        S    19:23   0:00 sshd: jeongju@pts/111
jeongju   32723  0.0  0.0 136644  2596 pts/111  Ss   19:23   0:00 -tcsh
[jeongju@flip2 lab4_ec]$ ./server &
[1] 487
[jeongju@flip2 lab4_ec]$ file name is received on sockfd 3

Successfully binded!

Waiting for file name...
jobs
[1]  + Running                  ./server
[jeongju@flip2 lab4_ec]$ █

```

6. Compile client.c on other directory

```

[jeongju@flip2 lab4_ec]$ ps aux | grep jeongju
root      23445  0.0  0.0 183720  6136 ?        Ss   19:23   0:00 sshd: jeongju [priv]
jeongju   29186  0.0  0.0 165628  3072 pts/111  R+   19:53   0:00 ps aux
jeongju   29187  0.0  0.0 112716   964 pts/111  S+   19:53   0:00 grep --color=auto jeongju
jeongju   32672  0.0  0.0 184036  3096 ?        S    19:23   0:00 sshd: jeongju@pts/111
jeongju   32723  0.0  0.0 136644  2596 pts/111  Ss   19:23   0:00 -tcsh
[jeongju@flip2 lab4_ec]$ ./server &
[1] 487
[jeongju@flip2 lab4_ec]$ file name is received on sockfd 3

Successfully binded!

Waiting for file name...
jobs
[1]  + Running                  ./server
[jeongju@flip2 lab4_ec]$ ls
alice.txt  server  server.c  test.txt
[jeongju@flip2 lab4_ec]$ cd ..
[jeongju@flip2 ~/cs372]$ pwd
/nfs/stak/users/jeongju/cs372
[jeongju@flip2 ~/cs372]$ ls
client  client.c  ec_lab1.py  ec_lab2.c  lab2  lab4_ec
[jeongju@flip2 ~/cs372]$ ./client
file is received

```

Please enter file name to receive:

█

7. Error handling for wrong file name

```
Please enter file name to receive:
idontknow

-----Data Received-----

File Name Received: idontknow

File open failed!

Waiting for file name...
File Not Found!
-----

Please enter file name to receive:
█
```

8. Test through alice.txt file and check the directory

```

[jeongju@flip2 ~/cs372]$ jobs
[1]  + Running                  ./server
[jeongju@flip2 ~/cs372]$ ls
client  client.c  ec_lab1.py  ec_lab2.c  lab2  lab4_ec
[jeongju@flip2 ~/cs372]$ clear
[jeongju@flip2 ~/cs372]$ gcc -o client client.c
[jeongju@flip2 ~/cs372]$ ./client

file will be received on sockfd 3

Please enter file name to receive:
alice.txt

-----Data Received-----

File Name Received: alice.txt

File Successfully opened!
      ALICE'S ADVENTURES IN WONDERLAND

      Lewis Carroll

      THE MILLENNIUM FULCRUM EDITION 3.0

      CHAPTER I

      Down the Rabbit-Hole

      Alice was beginning to get very tired of sitting by her sister
on the bank, and of having nothing to do: once or twice she had
peeped into the book her sister was reading, but it had no
pictures or conversations in it, 'and what is the use of a book,'
thought Alice 'without pictures or conversation?'

      So she was considering in her own mind (as well as she could,
for the hot day made her feel very sleepy and stupid), whether
the pleasure of making a daisy-chain would be worth the trouble
of getting up and picking the daisies, when suddenly a White
Rabbit with pink eyes ran close by her.

      There was nothing so VERY remarkable in that; nor did Alice
think it so VERY much out of the way to hear the Rabbit say to
itself, 'Oh dear! Oh dear! I shall be late!' (when she thought
it over afterwards, it occurred to her that she ought to have
wondered at this, but at the time it all seemed quite natural);
but when the Rabbit actually TOOK A WATCH OUT OF ITS WAISTCOAT-
POCKET, and looked at it, and then hurried on, Alice started to
her feet, for it flashed across her mind that she had never
before seen a rabbit with either a waistcoat-pocket, or a watch to
take out of it, and burning with curiosity, she ran across the
field after it, and fortunately was just in time to see it pop
down a large rabbit-hole under the hedge.

      In another moment down went Alice after it, never once
considering how in the world she was to get out again.

```

...

So she sat on, with **closed** eyes, and half believed herself in Wonderland, though she knew she had but to open them again, and all would change to dull reality--the grass would be only rustling in the wind, and the pool rippling to the waving of the reeds--the rattling teacups would change to tinkling sheep-bells, and the Queen's shrill cries to the voice of the shepherd boy--and the sneeze of the baby, the shriek of the Gryphon, and all the other queer noises, would change (she knew) to the confused clamour of the busy farm-yard--while the lowing of the cattle in the distance would take the place of the Mock Turtle's heavy sobs.

Lastly, she pictured to herself how this same little sister of hers would, in the after-time, be herself a grown woman; and how she would keep, through all her riper years, the simple and loving heart of her childhood: and how she would gather about her other little children, and make **THEIR** eyes bright and eager with many a strange tale, perhaps even with the dream of Wonderland of long ago: and how she would feel with all their simple sorrows, and find a pleasure in all their simple joys, remembering her own child-life, and the happy summer days.

THE END

Waiting for file name...

the requested file is received and stored **successfully**! And socket connection is end

[jeongju@flip2 ~/cs372]\$ ls

alice.txt **client** client.c **ec_lab1.py** ec_lab2.c **lab2** **lab4_ec**

[jeongju@flip2 ~/cs372]\$ █

```
[jeongju@flip2 ~/cs372]$ ls
alice.txt  client  client.c  ec_lab1.py  ec_lab2.c  lab2  lab4_ec
[jeongju@flip2 ~/cs372]$ vim alice.txt
[jeongju@flip2 ~/cs372]$ cat alice.txt
    ALICE'S ADVENTURES IN WONDERLAND
```

Lewis Carroll

THE MILLENNIUM FULCRUM EDITION 3.0

CHAPTER I

Down the Rabbit-Hole

Alice was beginning to get very tired of sitting by her sister on the bank, and of having nothing to do: once or twice she had peeped into the book her sister was reading, but it had no pictures or conversations in it, 'and what is the use of a book,' thought Alice 'without pictures or conversation?'

So she was considering in her own mind (as well as she could, for the hot day made her feel very sleepy and stupid), whether the pleasure of making a daisy-chain would be worth the trouble of getting up and picking the daisies, when suddenly a White Rabbit with pink eyes ran close by her.

There was nothing so VERY remarkable in that; nor did Alice think it so VERY much out of the way to hear the Rabbit say to itself, 'Oh dear! Oh dear! I shall be late!' (when she thought it over afterwards, it occurred to her that she ought to have wondered at this, but at the time it all seemed quite natural); but when the Rabbit actually TOOK A WATCH OUT OF ITS WAISTCOAT-POCKET, and looked at it, and then hurried on, Alice started to her feet, for it flashed across her mind that she had never before seen a rabbit with either a waistcoat-pocket, or a watch to take out of it, and burning with curiosity, she ran across the field after it, and fortunately was just in time to see it pop down a large rabbit-hole under the hedge.

In another moment down went Alice after it, never once considering how in the world she was to get out again.

The rabbit-hole went straight on like a tunnel for some way, and then dipped suddenly down, so suddenly that Alice had not a moment to think about stopping herself before she found herself falling down a very deep well.

Either the well was very deep, or she fell very slowly, for she had plenty of time as she went down to look about her and to wonder what was going to happen next. First, she tried to look down and make out what she was coming to, but it was too dark to see anything; then she looked at the sides of the well, and noticed that they were filled with cupboards and book-shelves;