CS542 Computer Networks I – Fundamentals
Assignment #2

Instructor name: Edward Chlebus

Student Name: Lohith Anantavarapu
CWID: A20536789
Hawk ID: alohith@hawk.iit.edu

Collaborator details:

Student Name: Vadlamudi Manogna
CWID: A20551908
Hawk ID: mvadlamudi@hawk.iit.edu

(Both have contributed equally)

1. Given an IP datagram with the fragmentation offset of 0000001011010_2, HLEN of 5_{16} and the total length of $007A_{16}$, find the numbers of the first byte and the last byte of data in this datagram.

Answer:

Given

Fragmentation offset = $0000001011010_2 = 90_{10}$

HLEN = $5_{16} = (5 \times 16^0) = 5$

Total length = $(007A)_{16} = (0 \times 16^3) + (0 \times 16^2) + (7 \times 16^1) + (10 \times 16^0)$

= 112 + 10

= $(122)_{10}$

→ First byte = The Fragmentation Offset specifies the position of the fragment in units of 8 bytes. Multiply the Fragmentation Offset by 8 to get the offset in bytes.

= 90 * 8 = 720

→ Data Length = Total Length−Header length =122 – (5*4) = 122 – 10 = 102

→ Last byte = First byte + Data Length – 1 = 720 + 102 – 1 = 821

The number of first byte and the last byte of the datagram are 720(first byte) and 821(last byte).

2. Consider fragmenting an original IP datagram whose total size is 6000_{10} bytes with a base header only. The offset of the second fragment is 98_{16}. Answer the following questions:

a. How many fragments are there? Give the data range for each of them. (Assume that all the fragments except the last one are equal.)

b. What is the total size of each fragment?

c. What is the fragmentation offset of the last fragment?

<u>Answer:</u>

Given:

Total size = 6000
Base header = 20
Data size = 6000 - 20 = 5980
Offset = $98_{16} = (9 \times 16^1) + (8 \times 16^0) = 144 + 8 = (152)_{10}$

a. Number of Fragments and Data Range:

value of first byte in second fragment = 152 * 8 = 1216
since total size = 6000
1st fragment => range => 000 – 1215 (size = 1216 bytes)
Offset => 000/8 = 0
2nd fragment => 1216 – 2431 (size = 1216 bytes)
Offset =>1216/8 = 152
3rd fragment => 2432 – 3647 (size = 1216 bytes)
Offset => 2432/8 = 304
4th fragment => 3648 – 4863 (size = 1216 bytes)
Offset => 3648/8 = 456
5th fragment => 4864 – 5979 (size = 1116 bytes)
Offset => 4864/8 = 608
There are total of 5 fragments.

b. Total size of each fragment

As, the first 4 fragments have equal size = 1216 (data size)
The total size of each of the first 4 fragments = 1216 + 20 (base header) = 1236 bytes
The total size of the last (5th) fragment = 1116 (data size) + 20 (base header) = 1136 bytes

c. Fragmentation offset of the last fragment = 608 (4864 / 8)

3. The first few hexadecimal digits of an IP datagram are as follows: 4E00 00B4 0034 408F. Find the total length, the header length, and the data size. Is there the next fragment? Can this datagram be fragmented?

<u>Answer:</u>

Given IP datagram.
4E00 00B4 0034 408F

- Total length = $(00B4)_{16} = (0 \times 16^3) + (0 \times 16^2) + (11 \times 16^1) + (4 \times 16^0)$
  $= 176 + 4 = (180)_{10} = 180$ bytes
- Header length = **E - 1110** in binary. The HLEN is 4 bits, which is **1110** or 14 in decimal. This means the header length is 56 bytes.
- Data size = Total length – header length = 180 – 56 bytes = 124 bytes

The total length, header length and data size are 180 bytes, 56 bytes and 124 bytes.

Is there the next fragment:

Fragmentation offset + Flags = $(408F)_{16} = (0100000010001111)_2$

- Flags(3-bits): 010

Reserved	Do not fragment	More fragments

- The “do not fragment” flag is set to 1.
- Fragmentation offset(13-bits): 0000010001111

The ‘D’ (do not fragment) bit is set to 1 we cannot fragment and hence there is no next fragment

4. An IP packet with HLEN = E_{16} carries 2540_{10} bytes of data. What is the size of the “Options”? What is the value (in the hexadecimal format) of the “Total length” field?

<u>Answer:</u>

Given HLEN = E = 14
2540 bytes of data
Header length = $(14 \times 4) = 56$ bytes

- As the base header is 20 bytes
- The remaining bytes are the options i.e., 56 – 20 = 36bytes
- Total length = 2540 + 56 = 2,596 bytes

The size of the “options” is 36 bytes.
The values of “total length” in decimal is 2,596 bytes , 1010 0010 0100 in binary and A24 in hexadecimal format.

5. An original IP datagram, that carries 3470_{10} bytes of data and has only the base header, was fragmented. The first fragment contains bytes from 0 to 399. All the fragments except the last one are equal. What is the total overhead (in bytes) needed to send all the data of the original datagram to the destination?

Answer:

Given

Data = 3470
HLEN = 20 bytes = $5 * 16^0 = 5 * 1 = 5_{16}$
First fragment bytes = 0 to 399
Size of first fragment = 400 bytes(all fragments except the last fragment have this size)
Number of fragments = 0 – 399 – 1st
400 – 799 – 2nd
800 – 1199 – 3rd
1200 – 1599 – 4th
1600 – 1999 – 5th
2000 – 2399 – 6th
2400 – 2799 – 7th
2800 – 3199 – 8th
3200 – 3469 – 9th
Size of the last fragment = 270 bytes
So, total overhead = 8*20 = 160 bytes

6. An IP packet has arrived with a "D" bit value of 0, an "M" bit value of 0, and fragmentation offset value to zero. Is this packet the first fragment, the last fragment, the middle fragment, or the only fragment?

Answer:

An IP packet with a "D" bit value of 0, which specifies 'do not fragment' bit set to 0 i.e., can be fragmented. "M" bit value is 0, which specifies that it can be the last or only fragment. Fragmentation offset value is 0, which specifies that it is the first fragment. The fragment that satisfy all the mentioned conditions should be the first and the only fragment.

7. A header of a UDP datagram is (in the hexadecimal format): 0315 C43B 00B3 001C (give your answers in the decimal format)

a. What is the source port number?

b. What is the destination port number?

c. What is the total length of this UDP datagram?

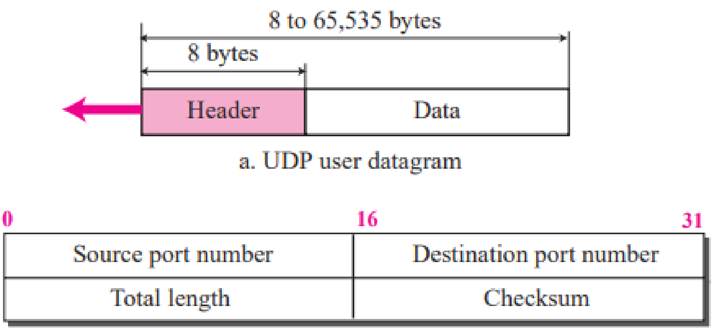
d. Is this UDP datagram sent from client to server or vice versa?

Answer:

Given UDP header dump:

0315 C43B 00B3 001C

Figure 14.2 *User datagram format*



From the figure the source & destination port number, total length can be derived.

a. Source port number:
$(0315)_{16} = (0 \times 16^3) + (3 \times 16^2) + (1 \times 16^1) + (5 \times 16^0)$
$= 768 + 16 + 5$
$= (789)_{10}$ [well-known port number]

b. Destination port number:
$(C43B)_{16} = (12 \times 16^3) + (4 \times 16^2) + (3 \times 16^1) + (11 \times 16^0)$
$= 49152 + 1024 + 48 + 11$
$= (50235)_{10}$ [ephemeral port number]

c. Total Length of UDP datagram: $(00B3)_{16} = (0 \times 16^3) + (0 \times 16^2) + (11 \times 16^1) + (3 \times 16^0)$
$= 176 + 3$
$= (179)_{10}$

d. Since the source port number is well-known and the destination port number is ephemeral port number. Hence, this UDP datagram is sent from **server to client**.

8. An IP packet has the base header and the total size of 1200_{10} bytes. A UDP datagram is encapsulated in this IP packet. How many bytes of data does this UDP datagram carry?

Answer:

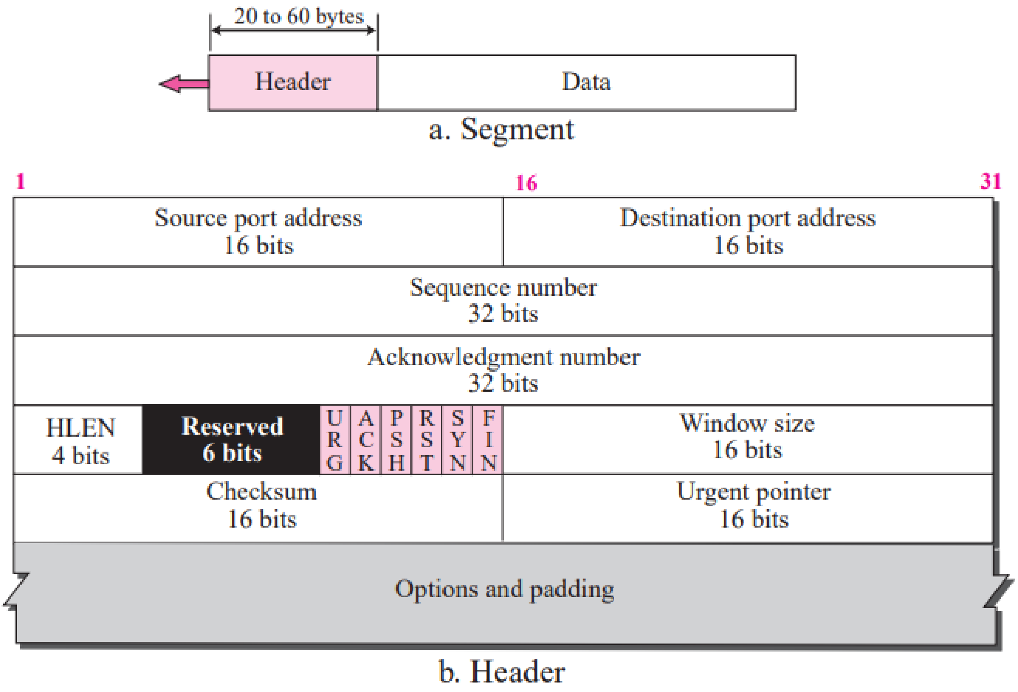
Base header length = 20bytes
Given Total size = 1200 bytes
UDP total length = Total size – IP Header length
UDP total length = 1200 – 20 = 1180 bytes
UDP datagram has a header of 8 bytes
UDP datagram carries a data of 1180 – 8 = 1172 bytes

9. The initial sequence number in the TCP client-server transmission was 4357. The highest ACK number that the server sent to the client was 10842. How many bytes of data were successfully transmitted from the client to the server?
Answer:

To determine the number of bytes successfully transmitted from the client to the server, we need to subtract the initial sequence number from the highest ACK number.
Number of bytes transmitted = Highest ACK number−Initial sequence number
Number of bytes transmitted = 10842−4357
Number of bytes transmitted = 6485
Therefore, 6485 bytes of data were successfully transmitted from the client to the server.

10. The following TCP header dump is given in the hexadecimal format: CB5A00D3 00B41234 00003021 50100EB4 00500000 (give your answers in the decimal format).
a. What are the source and destination port numbers?
b. What is the sequence number?
c. What is the acknowledgment number?
d. What is the header length?
e. Which flags are set?
f. What is the window size?
Answer:

Figure 15.5 *TCP segment format*



Given the Hexadecimal dump is:

CB5A00D3 00B41234 00003021 50100EB4 00500000

From the figure source & destination port, sequence &Acknowledgment number, header length, control field and window size can be derived.

a. Source and destination port numbers:

- **Source Port:** It is a 16-bit field that defines the port number of the application program in the host that is sending the segment.
- Source port: CB5A (Hex) = $(CB5A)_{16}$
  $= (12 \times 16^3) + (11 \times 16^2) + (5 \times 16^1) + (10 \times 16^0)$
  $= 49152 + 2816 + 80 + 10$
  $= (52058)_{10}$
- **Destination Port:** It is a 16-bit field that defines the port number of the application program in the host that is receiving the segment.
- Destination port: 00D3 (Hex)
  $=(00D3)_{16} = (0 \times 16^3) + (0 \times 16^2) + (13 \times 16^1) + (3 \times 16^0)$
  $= 208 + 3$
  $= (211)_{10}$

b. **Sequence number:** It is a 32-bit field defines the number assigned to the first byte of data contained in this segment.

- 00B41234 (hex)

$=(00B41234)_{16} = (0 \times 16^7) + (0 \times 16^6) + (11 \times 16^5) + (4 \times 16^4) + (1 \times 16^3) + (2 \times 16^2) + (3 \times 16^1) + (4 \times 16^0)$
$= 11534336 + 262144 + 4096 + 512 + 48 + 4$
$= (11801140)_{10}$

c. **Acknowledgment number:** It is a 32-bit field defines the byte number that the receiver of the segment is expecting to receive from the other party.

- 00003021 (hex)
  $=(00003021)_{16} =(0 \times 16^7) + (0 \times 16^6) + (0 \times 16^5) + (0 \times 16^4) + (3 \times 16^3) + (0 \times 16^2) + (2 \times 16^1) + (1 \times 16^0)$
  $= 12288 + 32 + 1$
  $= (12321)_{10}$

d. **Header length:** It is a 4-bit field indicates the number of 4-byte words in the TCP header. The length of the header can be between 20 and 60 bytes. The value of this field is 5 ($5 \times 4 = 20$). The header length is 20 bytes.

e. Reserved bits and Control field is 010(Hex), 12 bits n total.
010 (hex) = 000000010000 (binary). The flags are:
The last 6 bits are for control field or flags, i.e.,
URG: 0
ACK: 1
PSH: 0
RST: 0
SYN: 0
FIN: 0

Hence flags set **ACK**.

f. Window size: the relevant part is 0EB4 (Hex)
$(0EB4)_{16} = (0 \times 16^3) + (14 \times 16^2) + (11 \times 16^1) + (4 \times 16^0)$
$= 3584 + 176 + 4$
$= (3764)_{10}$
Window size = 3764

11. A TCP client-server connection was established with the initial sequence number of 1003_{10}. The client sent 500_{10} bytes of data in the first segment. What is the sequence number of this segment and the range of the transferred bytes? What is the sequence number of the second segment sent by this client?

<u>Answer:</u>

Given values:

1. **Initial sequence number:** 1003_{10}

2. **Bytes sent in the first segment:** 500_{10}

Sequence number of the first segment:
Sequence number=Initial sequence number=1003_{10}
Range of transferred bytes in the first segment:

- Starting byte position = Sequence number of the first segment=1003_{10}
- Ending byte position = Starting byte position + Bytes sent − 1 = 1003 + 500 − 1 = 1502_{10}.
- Range of bytes transferred is 1003 – 1502.

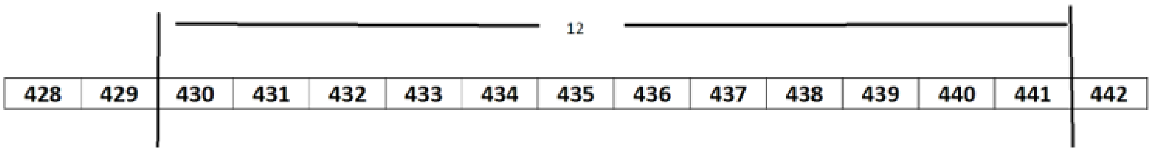
Sequence number of the second segment:

- Sequence number of the second segment = Ending byte position of the first segment + 1 = 1502 + 1 = 1503_{10}.

Segment 1 → Sequence Number: 1003_{10} Range: 1003_{10} to 1502_{10}
The value in the sequence number field of a segment defines the number assigned to the first data byte contained in that segment.

12. The current *cwnd*=12 and *rwnd*=26. The last acknowledgment received is 430. Draw the diagram showing this TCP window. A new TCP segment has just arrived with an acknowledgment number of 433 and *rwnd*=x. What is the minimum value of x to avoid shrinking the window?
<u>Answer:</u>

The current cwnd = 12 and rwnd = 26, last acknowledgment received is 430.
The window size = min (cwnd, rwnd)
Window size = 12



It is given that a new segment arrive with an acknowledgment number = 433 and rwnd = x
So, in order to avoid shrinking the minimum value of x can be calculated as,

new ackNo + new rwnd ≥ last ackNo + last rwnd

That is, 433 + x >= 430 + 26 minimum values of x so we can consider 433 + x = 430 + 26
x = 430 + 26 – 433 = 23
The minimum value of x to avoid shrinking is 23 i.e., rwnd = 23.

13. Is the size of the ARP packet fixed? Explain your answer.

<u>Answer:</u>

- No, not necessarily, since ARP is flexible enough to be used to discover or describe a connection between other kinds of addresses, which don't have to be the same lengths as IPv4 and Ethernet addresses.
- The ARP packet size must vary because it contains 2 Hardware/MAC addresses in it and 2 different protocol addresses in it. Depending on the datalink and network protocol used the size addresses vary.

14. A host with an IP address 171.65.22.101 and a physical address A9:27:BB:F3:29:56 has a packet to send to a host in another network. The destination IP and physical addresses are 119.254.100.1 and AC:45:9D:E2:DD:67, respectively (this physical address is unknown to the sender). The next hop for this destination found in the sender's routing table is router R2 with an IP address 118.254.100.1 and a physical address AC:45:9C:52:66:B9 (this physical address is unknown to the sender). Show the ARP request and reply packets. Fill all the necessary fields. Ethernet and IPv4 protocols are implemented at the data link layer and the network layer, respectively.

<u>Answer:</u>

The ARP request and reply packets for the given scenario:

ARP Request -

From source to router R1 -

0x0001		0x0800
0x06	0x04	0x0001

A9:27:BB:F3:29:56 0xAB411665 0x000000000000 0x76FE6401

Preamble and SFD	0xFFFFFFFF FFFF	A9:27:BB:F3 :29:56	0x0806	Data 28 bytes	CRC

ARP Reply -

From router R1 to source -

0x0001	0x0800

0x06	0x04	0x0002

AC:45:9C:52:66:B9
0x76FE6401
A9:27:BB:F3:29:56
0xAB411665

Preamble and SFD	A9:27:BB:F3 :29:56	AC:45:9C:52 :66:B9	0x0806	Data	CRC

ARP request -

From router R1 to destination -

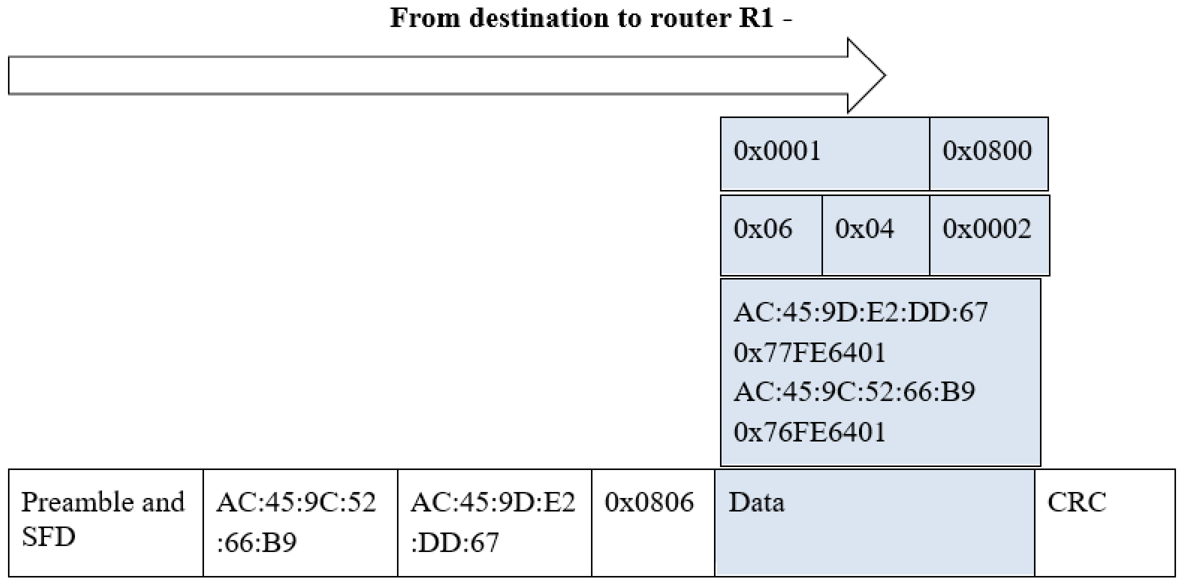
0x0001	0x0800

0x06	0x04	0x0001

AC:45:9C:52:66:B9
0x76FE6401
0x000000000000
0x77FE6401

Preamble and SFD	0xFFFFFFFF FFFF	AC:45:9C:52 :66:B9	0x0806	Data 28 bytes	CRC

ARP Reply -



15. What destination address is used in the Ethernet frame carrying an ARP request? Explain your answer.

<u>Answer:</u>

An ARP request is broadcast and ARP reply is unicast. Every host or router on the network receives and processes the ARP query packet, but only the intended recipient recognizes its IP address and sends back an ARP response packet. The destination address used in Ethernet frame carrying an ARP request is same as the Ethernet broadcast address i.e., the broadcast address of Ethernet is **FF:FF:FF:FF:FF:FF.**

16. Router R1 has received an ARP request. Can this ARP packet be used to update a cache table of R1? Explain your answer. Note that R1 has received an ARP request not an ARP reply.

<u>Answer:</u>

No, an ARP request cannot be used to update a cache table of R1.

- When a device receives an ARP request, it means that the requesting device is trying to find the MAC address of another device on the network.
- The ARP request contains the IP address of the target device, but it does not contain the MAC address.
- The ARP cache table of a router like R1 is updated when it receives ARP replies that provide the necessary mappings between IP addresses and MAC addresses.
- ARP requests are used to trigger the process by asking for the MAC address associated with a specific IP address, but it's the ARP replies that contain the information used to update the ARP cache.