CS 542 - Assignment 2

Sukanta Sharma (A20472623), Vidhi Kakani (A20473969)

CS 542 – Computer Networks – I | Spring 2021 | Illinois Institute of Technology

5/14/21

Instructions:

- Please submit soft copies on the blackboard.
- Team submissions are accepted. A team of 1 – 3 is accepted.
- For team submissions, one submission is sufficient. Anyone from the team can submit—no need for everyone in the group to submit.
- All should submit typewritten documents. The handwritten ones are not accepted. Zero points will be awarded if the submission is not a typewritten one.
- Please contact **Viswatej Kasapu (vkasapu@hawk.iit.edu)** if something is not clear. After submission, please do not say any excuses like "we understood differently." If you doubt any questions, please email me but do not expect me to give ideas or hints to the solution.
- **The due date is Friday, May 14, 2021, at 11:59 PM (midnight) Central Time.**
- Submissions after the due date are not accepted. This is the hard deadline. I must submit the grades to the University before the deadline. So, I cannot provide extensions to you. **Please submit before the due date without fail. Otherwise, zero points will be given.**

Instructions:

- Please submit soft copies on the blackboard.
- For every question which has a calculation, you should show steps clearly. No points will be given for direct answers. Explanation and justification are mandatory.
- In all questions, provide answers in the decimal system (not binary, hexadecimal, or 256 bases).

1. A host with IP address 130.23.43.20 and physical address B2:34:55:10:22:10 has a packet to send to another host on another network with IP address 141.23.56.21 and physical address A4:6E:F4:59:83:AB. The next-hop (router) for this destination in the sender's routing table is Router R1 with IP address 130.23.43.25 and physical address B2:53:45:01:33:10. Give the ARP request packet format from the sender and its corresponding reply packet format filled with all necessary fields. Consider the Ethernet as hardware type and IPv4 as protocol type. **(10 points)**

Ans:

The structure of the ARP Packet is given as below:

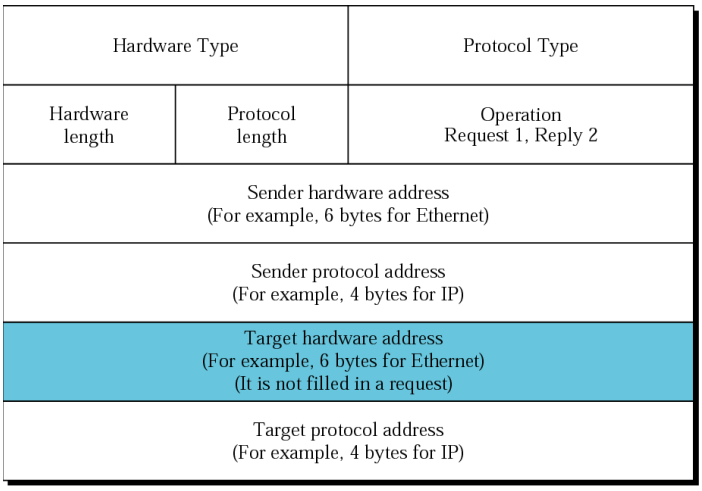


Figure 1: ARP Packet

- ARP Request Packet:

The ARP Request Packet is given as follows:

0x0001		0x0800
0x06	0x04	0x0001
0xB23455102210		
0x82172B14		130.23.43.20
0x000000000000		
0x82172B19		130.23.43.25

- ARP Reply Packet:

The ARP Request Packet is given as follows:

0x0001		0x0800
0x06	0x04	0x0002
0xB25345013310		
0x82172B19		130.23.43.25
0xB23455102210		
0x82172B14		130.23.43.20

2. Consider the updated ARP cache table at time t. The maximum number of attempts is 10, and the time-out value is 600 seconds. After 120 seconds, the input module receives two ARP packets, and the output module receives one IP packet from IP software. These are the only three packets host received in the last 120 seconds. Consider cache table is updated every 60 seconds. Give the updated cache table at times t+60 seconds and t+120 seconds, respectively. **(10 points)**

Packets Received:

- An ARP reply from the host with IP address 114.5.7.89 and physical address 457342ACAE32
- An ARP reply from the host with IP address 201.11.56.7 and physical address A46EF45983BC
- An IP packet that has to be forwarded to the next hop with IP address 188.11.8.71

State	Queue	Attempt	Time-out	Protocol Address	Hardware Address
R	5		500	180.3.6.1	ACAE32457342
P	2	2		129.34.4.8	
P	14	7		201.11.56.7	
R	8		60	114.5.7.89	457342ACAE32
F					

Ans:

- Cache Table at time t+60 seconds:

For every entry in the cache table

1. For the first entry, the state is RESOLVED, so it will decrease the value of the timeout by 60 seconds (i.e., 500 - 60 = 440 seconds).
2. For the second entry, the state is PENDING, so it will increase the value of the attempt by 1 (i.e., 2 + 1 = 3) and send an ARP request.
3. For the third entry, the state is PENDING, so it will increase the value of the attempt by 1 (i.e., 7 + 1 = 8) and send an ARP request.
4. For the fourth entry, the state is RESOLVED, so it will decrease the value of the timeout by 60 seconds (i.e., 60 – 60 = 0 seconds). The updated timeout value is equal to zero, so the state of the entry is changed to FREE, and the corresponding queue is destroyed.
5. For the fifth entry, the state is FREE, so it will continue.

The final updated cache table after **t + 60** seconds is given below:

State	Queue	Attempt	Time-out	Protocol Address	Hardware Address
R	5		440	180.3.6.1	ACAE32457342
P	2	3		129.34.4.8	
P	14	8		201.11.56.7	
F					

State	Queue	Attempt	Time-out	Protocol Address	Hardware Address
F					

- Cache Table at time t+120 seconds:
  - **Cache-Control Module:**

    For every entry in the cache table:

    1. For the first entry, the state is RESOLVED, so it will decrease the value of the timeout by 60 seconds (i.e., 440 – 60 = 380 seconds).
    2. For the second entry, the state is PENDING, so it will increase the value of the attempt by 1 (i.e., 3 + 1 = 4) and send an ARP request.
    3. For the third entry, the state is PENDING, so it will increase the value of the attempt by 1 (i.e., 8 + 1 = 9) and send an ARP request.
    4. For the fourth entry, the state is FREE, so it will continue.
    5. For the fifth entry, the state is FREE, so it will continue.

    After the operation of the Cache-control module the Cache-table will be given as below:

State	Queue	Attempt	Time-out	Protocol Address	Hardware Address
R	5		380	180.3.6.1	ACAE32457342
P	2	4		129.34.4.8	
P	14	9		201.11.56.7	
F					
F					

  - **Input Module:**

1. The input module receives an ARP packet with target protocol (IP) address 114.5.7.89 and physical address 457342ACAE32. The module checks the table and finds this address. The module checks the table and does not find this address. It will create a cache entry with the state set to RESOLVED and timeout value set to 600 seconds and add the entry to the table with the received protocol and hardware address. It will create a new queue number 20 for this destination address. The updated cache-table will be given as:

State	Queue	Attempt	Time-out	Protocol Address	Hardware Address
R	5		380	180.3.6.1	ACAE32457342
P	2	4		129.34.4.8	
P	14	9		201.11.56.7	

State	Queue	Attempt	Time-out	Protocol Address	Hardware Address
R	20		600	114.5.7.89	457342ACAE32
F					

2. The input module receives an ARP packet with target protocol (IP) address 201.11.56.7 and physical address A46EF45983BC. The module checks the table and finds this address. It changes the state of the corresponding entry to RESOLVED, the attempt value is deleted, and sets the timeout value to 600 seconds. The module then adds the target hardware address (i.e., A46EF45983BC) to the entry. Now it accesses queue 14 and sends all the packets in this queue, one by one, to the data link layer. The updated cache table will be given as below:

State	Queue	Attempt	Time-out	Protocol Address	Hardware Address
R	5		380	180.3.6.1	ACAE32457342
P	2	4		129.34.4.8	
R	14		600	201.11.56.7	A46EF45983BC
R	20		600	114.5.7.89	457342ACAE32
F					

- **Output Module:**

The output module receives an IP packet with the next-hop address 188.11.8.71. It checks the cache table and does not find this address in the table. The module adds an entry to the table with the state PENDING and the attempt value 1. It creates a new queue number 15 for this destination and enqueues the packet. It then sends an ARP request to the data link layer for this destination.

The final updated cache table after **t + 120** seconds is given below:

State	Queue	Attempt	Time-out	Protocol Address	Hardware Address
R	5		380	180.3.6.1	ACAE32457342
P	2	4		129.34.4.8	
R	14		600	201.11.56.7	A46EF45983BC
R	20		600	114.5.7.89	457342ACAE32
P	15	1		188.11.8.71	

3. For each one, mention whether it is a valid or invalid value for the HLEN field in the IP datagram header. Give your supporting reasons. **(4 points)**

a. 1011

Ans:

The header length should be between 20 and 60 bytes. HLEN field contains the value in 4-byte word format in binary. So, the header length is $1011_2 * 4_{10} = 11_{10} * 4_{10} = 44_{10} bytes$. So, the given value for the HLEN field is valid.

b. 1201

Ans:

The header length should be between 20 and 60 bytes. HLEN field contains the value in 4-byte word format in binary. The given value for the HLEN field is invalid because the given value is not in binary format.

c. 0011

Ans:

The header length should be between 20 and 60 bytes. HLEN field contains the value in 4-byte word format in binary. So, the header length is $0011_2 * 4_{10} = 3_{10} * 4_{10} = 12_{10} bytes$. So, the given value for the HLEN field is invalid.

d. 0101

Ans:

The header length should be between 20 and 60 bytes. HLEN field contains the value in 4-byte word format in binary. So, the header length is $0101_2 * 4_{10} = 5_{10} * 4_{10} = 20_{10} bytes$. So, the given value for the HLEN field is valid.

4. In an IP packet, the value in the HLEN field is 1100, and the value of the total length is 111111000. How many bytes of data is the packet carrying? Are there any options? If so, what is the length of the options? **(3 points)**

Ans:

HLEN value = $1100_2 = 12_{10}$

Header Length = $12 * 4 = 48\ bytes$

Total Length = $111111000_2 = 1 * 2^8 + 1 * 2^7 + 1 * 2^6 + 1 * 2^5 + 1 * 2^4 + 1 * 2^3 = 504_{10}\ bytes$

Data Length = $Total\ Length - Header\ Length = 504 - 48 = 456\ bytes$

- 456 bytes of data is the packet carrying.
- The minimum header length is 20 bytes, and the given header length is greater than that. So, yes, there are options present, and the length of the options is $28\ bytes\ (= 48 - 20)$.

5. The total IP datagram length is 70 bytes, out of which data length is 34 bytes. Is this example a valid IP datagram or not? Give your supporting reasons. **(2 points)**

Ans:

IP datagram total length = 70 bytes

Data length = 34 bytes

Header length = $Total\ Length - Data\ Length = 70 - 34 = 36\ bytes$, which is between 20 and 60 bytes, so this is a valid example of an IP datagram.

6. An IP datagram is divided into three fragments. All fragments are equal in size and have a base header of 20 bytes. The size of data in each fragment is 800 bytes. The first and last fragments can be divided further, but the second cannot be fragmented further. Give D, M, and fragmentation offset values of each fragment. (4 points)

Ans:

The size of data in each fragment is 800 bytes, so the data bytes sent in the first fragment will be from 000 to 799, in the second fragment will be from 800 to 1599, and in the third fragment will be from 1600 to 2399. If the D value is 1, then the datagram must not fragment further and if the D value is 0, then the datagram can be fragmented further. If the M value is 1, then the datagram is not the last and if the M value is 0, then the datagram is the last.

The values for D, M, and fragmentation offset for each fragment is given below:

#Fragment	D	M	Fragmentation Offset
1	0	1	$\frac{000}{8} = 0$

#Fragment	D	M	Fragmentation Offset
2	1	1	$\frac{800}{8} = 100$
3	0	0	$\frac{1600}{8} = 200$

7. A fragment has arrived with the first few hexadecimal digits, as shown below:

4500 003C 0001 8370……

This is the second fragment. How many bytes of data does this fragment contain? What is the offset of the next fragment? **(3 points)**

Ans:

The format of the IP datagram is given as below:

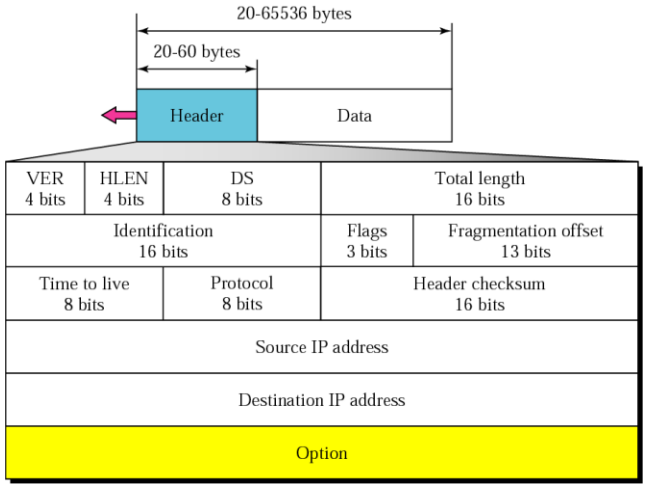


Figure 2:IP Datagram

From the above figure and the received data in hexadecimal digit, we have the following information:

HLEN = $5_{16} = 5_{10}$

Header Length = $5 * 4 = 20\ bytes$

Total Length = $003C_{16} = 3 * 16^1 + 12 * 16^0 = 60_{10} bytes$

Data Length = $60 - 20 = 40\ bytes$

- 40 bytes of data this fragment contains.

The flag and fragmentation offset = $8370_{16} = 1000\ 0011\ 0111\ 0000_2$

First bit = 1_2 (reserved)

Second bit, D = 0_2

Third bit, M = 0_2, it means this datagram is the last datagram.

The fragmentation offset (next 13 bits) = $0\ 0011\ 0111\ 0000_2 = 1 * 2^9 + 1 * 2^8 + 1 * 2^6 + 1 * 2^5 + 1 * 2^4 = 880_{10}$

- As we can see, the value of the M bit is 0, which means this is the last fragment, so there will not be any next fragment. So, the next fragment cannot be calculated.

8. The first 32 bits of an IP datagram are shown below. Is it a valid IP datagram? Explain your answer? **(2 points)**

0001 1010 0000 0000 0000 0000 0001 1110…

Ans:

From Fig 2 and the given data we can have the following information:

VER = 0001_2 (not a valid version)

HLEN = $1010_2 = 10_{10}$

Header Length = $10 * 4 = 40\ bytes$

Total Length = $0000\ 0000\ 0001\ 1110_2 = 1 * 2^4 + 1 * 2^3 + 1 * 2^2 + 1 * 2^1 = 30_{10}\ bytes$ (not a valid total length)

From the above information, we can see that the total length is less than the header length, and the VER should be either 0100_2 or 0110_2 but given is 0001_2, which are not possible, so it is an invalid IP datagram.

9. An IP packet has arrived with the first few hexadecimal digits as shown below:

4600 0040 0001 0000 1217……

The initial 'Time to Live" value in the hexadecimal format is BC. How many hops have this packet already traveled? How many hops can this packet travel before being dropped? **(3 points)**

Ans:

From Fig 2 and the given data we can have the following information:

Received Time to Live = $12_{16} = 1 * 16^1 + 2 * 16^0 = 18_{10}$

Initial Time to Live = $BC_{16} = 11 * 16^1 + 12 * 16^0 = 188_{10}$

- This packet has traveled $170\ hops\ (= 188 - 18)$ already.
- This packet can travel $18\ hops$ more before being dropped.

10. When does the TCP sliding windows shrink? Why is it not recommended? **(2 points)**

Ans:

- When the right-side wall of the window is moved towards the left of the original position, then it is called the shrinking of the sliding window. TCP sliding windows shrink when the following condition is not met:

$$\boldsymbol{new\ ack + new\ rwnd \geq last\ ack + last\ rwnd}$$

Or

$$\boldsymbol{new\ rwnd \geq (last\ ack + last\ rwnd) - new\ ack}$$

Where,

$new\ ack \rightarrow new\ acknowledgment\ value$

$last\ ack \rightarrow last\ acknowledgment\ value$

$new\ rwnd \rightarrow new\ receiving\ window$

$last\ rwnd \rightarrow last\ receiving\ window$

- Shrinking of the window means that the bytes which are already send are now outside of the window and the receiver has not received those. Those bytes will be sent again which will create overhead in the system and hinder the throughput of the TCP protocol. So, shrinking is not recommended and this can be avoided by satisfying the condition given above.

11. Given following TCP header dump in hexadecimal format. Give your answers in decimal format **(5 points)**

0325 0091 0000 0321 0000 3467 5001 08BE 0000 0000

a. What is the source port number?

Ans:

The format of TCP Segment is given as:

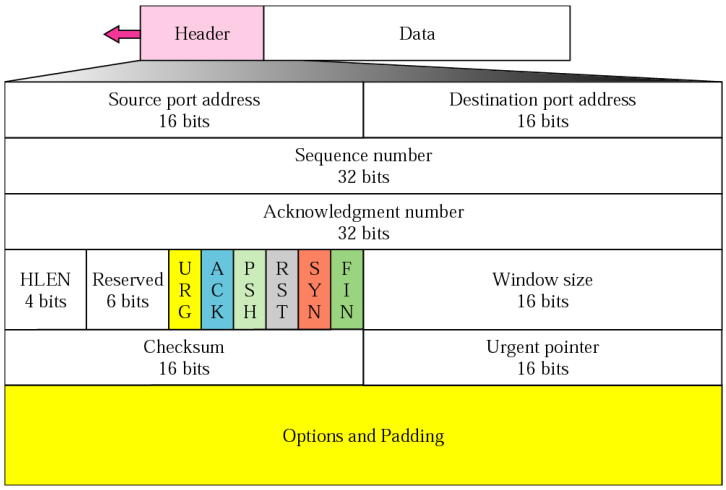


Figure 3: TCP Segment

From Fig 3 and the data given above we have the following information:

Source Port Number = $0325_{16} = 3 * 16^2 + 2 * 16^1 + 5 * 16^0 = 805_{10}$

The source port number is 805.

b. What is a sequence number?

Ans:

Sequence number = $0000\ 0321_{16} = 3 * 16^2 + 2 * 16^1 + 1 * 16^0 = 801_{10}$

The sequence number is 801.

c. What is header length?

Ans:

HLEN = $5_{16} = 5_{10}$

Header Length = $5 * 4 = 20\ bytes$

The header length is 20 bytes.

d. What is the use of segment (which bit is set in the control field and give your answer based on that bit)

Ans:

Control filed = $001_{16} = 0000\ 0000\ 0001_2$

So, the last bit, FIN = 1

So, this segment is used to terminate the connection and the request is sent from client to server.

e. What is the window size?

Ans:

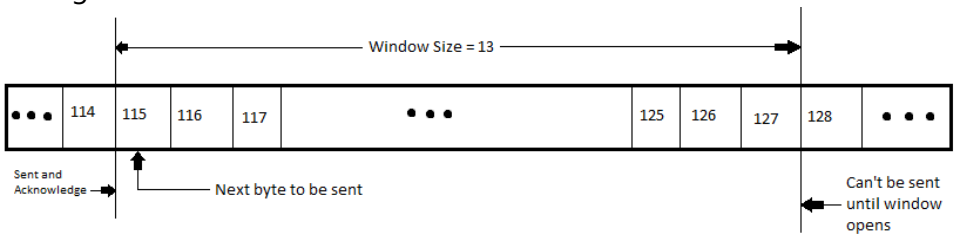
Window size = $08BE_{16} = 8 * 16^2 + 11 * 16^1 + 14 * 16^0 = 2238_{10}\ bytes$

The window size if 2238 bytes.

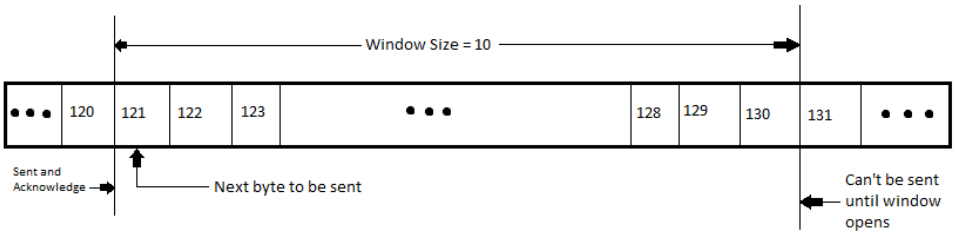
12. The TCP sliding window values of rwnd and cwnd are 18 and 13, respectively. The last acknowledgment number was 115. A segment with the acknowledgment number 121 and the rwnd of 10 has just been received. Draw a diagram showing the window before and after. The assumption cwnd has not changed. **(4 points)**

Ans:

- **Before:**
  ACK = 115
  RWND = 18
  CWND = 13
  Window size = min (18, 13) = 13
  The window is given as below:



- **After:**
  ACK = 121
  RWND = 10
  CWND = 13
  Window size = min (18, 10) = 10
  The window is given as below:



13. The UDP header in the hexadecimal format is 0223 000E 00AA E217. **(3 points)**

a. What is the source port number?

Ans:

The format of User Datagram is given as below:

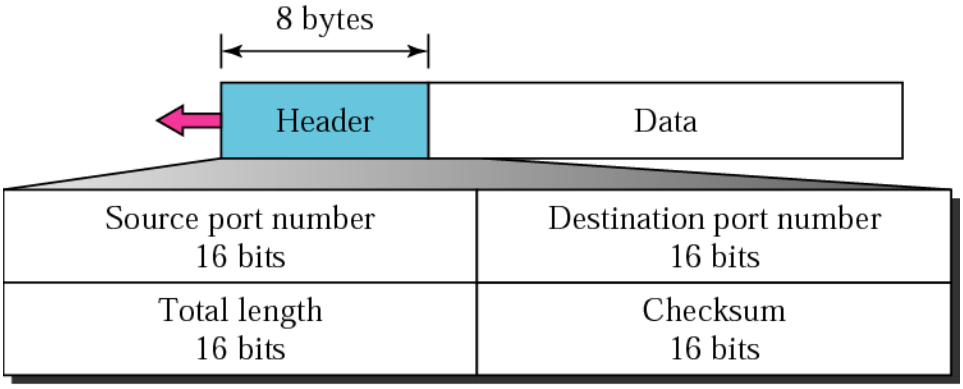


Figure 4: User Datagram

From Fig 4 and the given data we have the following information:

Source Port Number = $0223_{16} = 2 * 16^2 + 2 * 16^1 + 3 * 16^0 = 547_{10}$

The source port number is 547.

b. What is the destination port number?

Ans:

Destination Port Number = $000E_{16} = 14_{10}$

The destination port number is 14.

c. What is the total length of the user datagram?

Ans:

The total length of the user datagram = $00AA_{16} = 10 * 16^1 + 10 * 16^0 = 170_{10}\ bytes$

The total length of the user datagram is 170 bytes.