

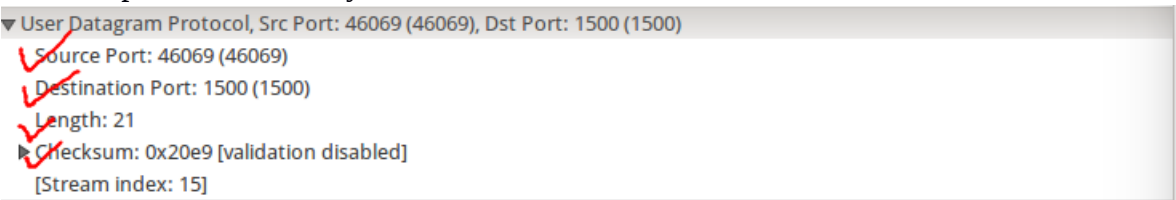
Introduction to Computer Network Proj3 Report

0416324 胡安鳳

Exp1 UDP packets

1.

Select the UDP packet. How many fields are there in UDP header? Name these fields.

Ans: 

This picture is captured from wireshark, as we can see, there are total of four fields in header, including Source Port, Destination Port, Length of the total UDP data(including the header, unit in byte) and the checksum for error detection(but UDP does not implement error recovery, it has to be done by the application in upper level).

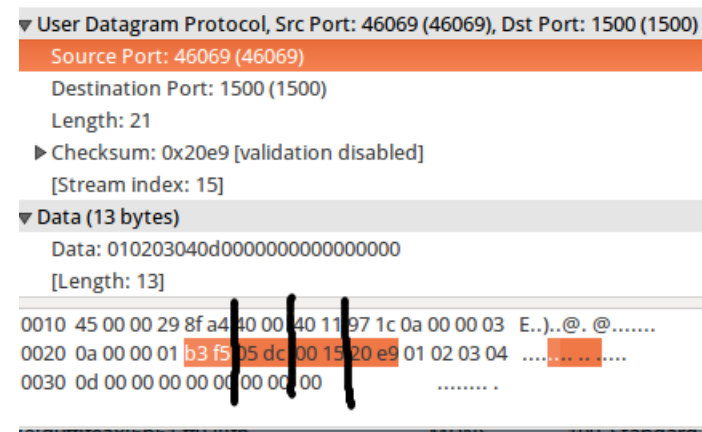
Validating from wikipedia, it is also 4 fields.

Offsets		0								1								2								3							
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Source port																Destination port															
4	32	Length																Checksum															

2.

Following the first operation, click on the header and observe the display at the bottom of Wireshark window. Determine the length(in bytes) of each field

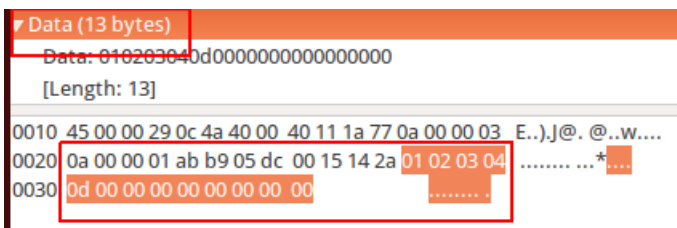
Ans:

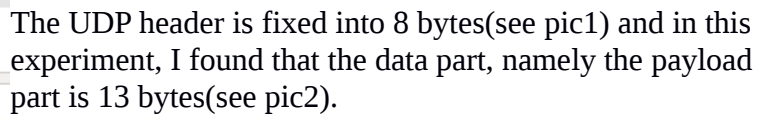


As the picture taken from wireshark, each field is equally divided into 16bits, namely 2 bytes.

3.

Focus on Length field. According to the format: total Length = header + payload, fill in the numbers and explain how you calculate. Ans as follows:





Ans:



Ans:

1.

```

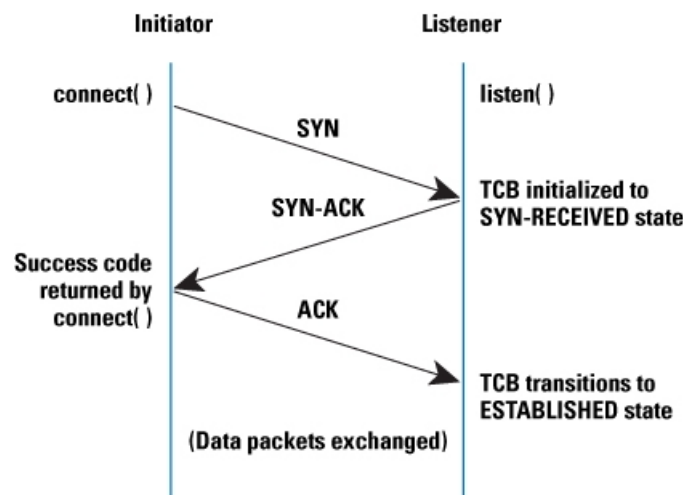
root@icn2017fall:~/proj3# ./tcpclient
Please Input Server Address: 10.0.0.1
Hi,this is server.
root@icn2017fall:~/proj3#

```

	IP Addr	Port
Client	10.0.0.3	8700
Server	10.0.0.1	38226

2.

Observing the experimental result, find the segments of “Three Way Handshake,” and answer the following questions.



(a) What are the Sequence and ACK numbers in each of the segments?

Ans:

10.0.0.3	10.0.0.1	TCP	74	38226 → 8700 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=2655519 TSecr=0 WS=512
10.0.0.1	10.0.0.3	TCP	74	8700 → 38226 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=2655520 TSecr=2655519 WS=512
10.0.0.3	10.0.0.1	TCP	66	38226 → 8700 [ACK] Seq=1 Ack=1 Win=29696 Len=0 TSval=2655521 TSecr=2655520

1st part SEQ1=0 ACK1= No (No one has transmitted data to client yet, the 3-way handshaking is just about to begin)

2nd part SEQ2=0 (I send my SEQ data to the client host)

ACK2=1 (I got client's SEQ0 data and I expected to get the client's next sequence which is 1)

3rd part SEQ3=1 (I send my SEQ data to the server host)

ACK3=1 (I got the server's SEQ0 data and I expected to get the client's next sequence which is 1)

(b) Do these three segments contain any data?

Ans:

No the real data transfer **happens when PSH flag is set**

Reference to:

<https://read01.com/zh-tw/dE6deN.html#.Wi0lqHaGOiY>

3. Find segments in which the FIN Flag field is set. Explain the purpose of these segments.

Ans:

The packet with FIN flag indicates the TCP connection should be terminated.

Why is the FIN flag in TCP called FIN?

FIN is an abbreviation for "Finish"

In the normal case, each side terminates its end of the connection by sending a special message with the **FIN (finish)** bit set. This message, sometimes called a FIN, serves as a connection termination request to the other device, while also possibly carrying data like a regular segment. The device receiving the FIN responds with an acknowledgment to the FIN to indicate that it was received. The connection as a whole is not considered terminated until both sides have finished the shut down procedure by sending a FIN and receiving an ACK.

Source: stackoverflow

4. Choose a packet from your experiment as an example.

10.0.0.3	10.0.0.1	TCP	74 38226 → 8700 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=2655519 TSecr=0 WS=512
10.0.0.1	10.0.0.3	TCP	74 8700 → 38226 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=2655520 TSecr=2655519 WS=512
10.0.0.3	10.0.0.1	TCP	66 38226 → 8700 [ACK] Seq=1 Ack=1 Win=29696 Len=0 TSval=2655521 TSecr=2655520

(a) What is the value of window size?

Window size is 29200 as the picture shown above.

(b) What is it used for?

It is used for flow controlling in TCP

reference: <https://wizardforcel.gitbooks.io/network-basic/content/7.html>

(c) Find the sequence numbers of the segments that can be sent by the TCP sender

ExpBonus

1.

Find and mark the interface names of switch on the topology you created in previous experiments. (Mark the names on the green labels of the following picture.)

Ans:

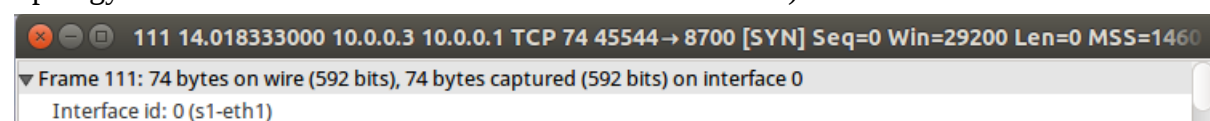
Consider the time sequence of wireshark from top to down of sending the data b/w h1,h3 with the following picture.

The data is transmitted in the blue line, either from h1 to h3 or from h3 to h1

Take an example of data from h1 to h3, the only road the data does not pass is s2h2, hence path of data transmission is h1s1 → s1s2 → s2s3 → s3h3 using this sequence, I can identify the label of each interface if the switch.

The following picture shows the result, and timing from start to end.

(Note, the time sequence does not listed in ascending order since there are many switch interfaces to be detected by the wireshark at the same time, however, we can still find the connectivity of the topology where each of the interface of the switch should be.)



First, h1s1 and the label is s1-eth1

122 14.021747000 10.0.0.3 10.0.0.1 TCP 66 45544 → 8700 [ACK] Seq=1 Ack=1 Win=2
 ▼ Frame 122: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 1
 Interface id: 1 (s1-eth2)

Then s1s2, first through s1-eth2

130 14.015688000 10.0.0.3 10.0.0.1 TCP 74 [TCP Spurious Retransmission] 45544 → 8700 [ACK] Seq=1 Ack=1 Win=2
 ▼ Frame 130: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 3
 Interface id: 3 (s2-eth2)

Then the s2-end part of s1s2, s2-eth2

140 14.013102000 10.0.0.3 10.0.0.1 TCP 74 [TCP Spurious Retransmission] 45544 → 8700 [SYN] Seq=1
 ▼ Frame 140: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 4
 Interface id: 4 (s2-eth3)

Then s2s3, first through s2-eth3

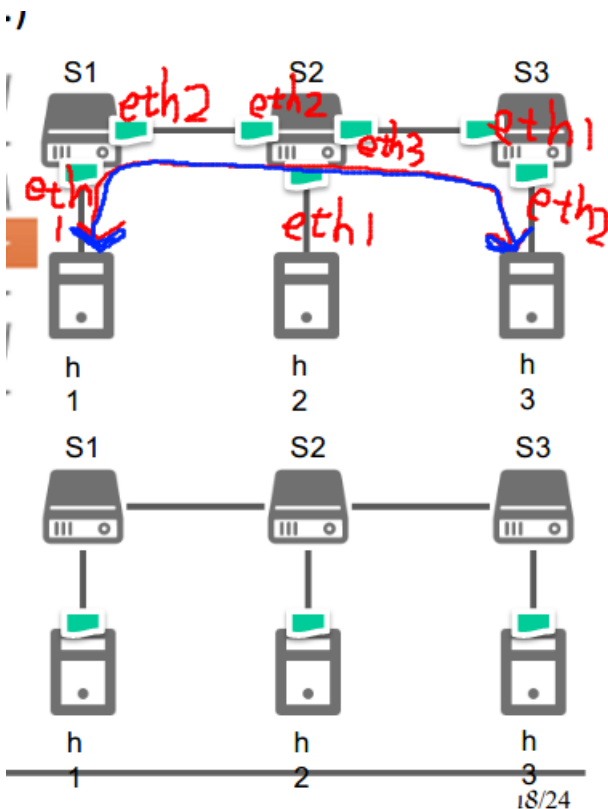
152 14.021635000 10.0.0.3 10.0.0.1 TCP 66 45544 → 8700 [ACK] Seq=1 Ack=1 Win=29696 Len=0 TS=1
 ▼ Frame 152: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 5
 Interface id: 5 (s3-eth1)

Then the s3-end part of s1s2, s3-eth1

166 14.025229000 10.0.0.3 10.0.0.1 TCP 66 [TCP Retransmission] 45544 → 8700 [FIN, ACK] Seq=10
 ▼ Frame 166: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 6
 Interface id: 6 (s3-eth2)

And finally reach end

This is what the answer should be



2.How do you observe the packets passed through the interface of h1?

Ans:

Use the filter with `dst addr or src addr==10.0.0.1` (h1 is the server) in wireshark