**Computer Network HW2**

**Shujun Bian GTID: 903172758**

**Problem1:**

a) How is the “Token” calculated in MPTCP? And what is it used for?

Token is a truncated cryptographic hash of the key that is used to identify which MPTCP connection current subflows is joining.

MPTCP needs to be able to link each subflow to an existing MPTCP connection. For this, MPTCP assigns a locally unique token to each connection.

b) Why does MPTCP include data sequence numbers in the options field? In other words why is a single sequence number space not enough?

Because we need to decide the sequence of the data between different TCP connections, if we use single sequence number to do that, there might be some gap exists in sequence numbering space while some DPI don’t allow that. Moreover, Some middleboxes may change the sequence number space in the networks.

c) What is the difference between the receive window of regular TCP and MPTCP?

In regular TCP, there is only one TCP connection, so the sever would include the size of window in each packet/Ack, then the sender would know the size of the receive window.

But in MPTCP, an unqiue window is shared among all subflows and the information of the window is transmitted inside the window field of the regular TCP header. While the middleboxes change the size of the window, it would use the largest window received at MPTCP-level on each flow.

d) Explain the two choices for where to put the MPTCP-specific parameters and

options. Which one was chosen by the designers and why?

One of the choices is using TLVs to encode data and control information inside payload. Another is to use TCP options field to encode MPTCP-specific parameters.

The designers chose the latter. Because using TCP options is a normal way of extending TCP and should be able to go through middleboxes or fallback. If we using TLVs inside the payload, TCP segments contain TLVs including the data and not only the data, so middlebox would make it difficulty to deal with the parameters.

e) The sender has 9 data segments to send over three MPTCP subflows. Each subflow is responsible for sending 3 segments. Suppose the Initial data sequence number is 1 and each segment contains 1 byte of data. Give a reasonable sequence mapping on these three subflows. Just consider the data transfer phase without showing connection or termination.

If three subflows initial sequence number is 123, 456, and 789.

So the mapping would be like:

Dseq=1, seq=123 [1->123] Dseq=4, seq=124 [4->124] Dseq=7, seq=125[7->125]

Dseq=2, seq=456 [2->456] Dseq=5, seq=457 [5->457] Dseq=8, seq=458[7->458]

Dseq=3, seq=789 [3->789] Dseq=6, seq=790 [6->790] Dseq=9, seq=791[9->791]

f) One possibility for implementing MPTCP is to build it as at the application layer (above the sockets interface) and build it above TCP. 1) Draw an architecture picture that shows how this is accomplished. 2) Show the header structure for an IP packet carrying MPTCP data in such an architecture. 3) Would such a system have trouble going through middleboxes (why or why not)? 4) Explain one important disadvantage of such a set up.

|  |
| --- |
| Application |
| Multipath TCP |
| socket |
| |  |  |  |  | | --- | --- | --- | --- | | TCP | TCP | …... | TCP | |

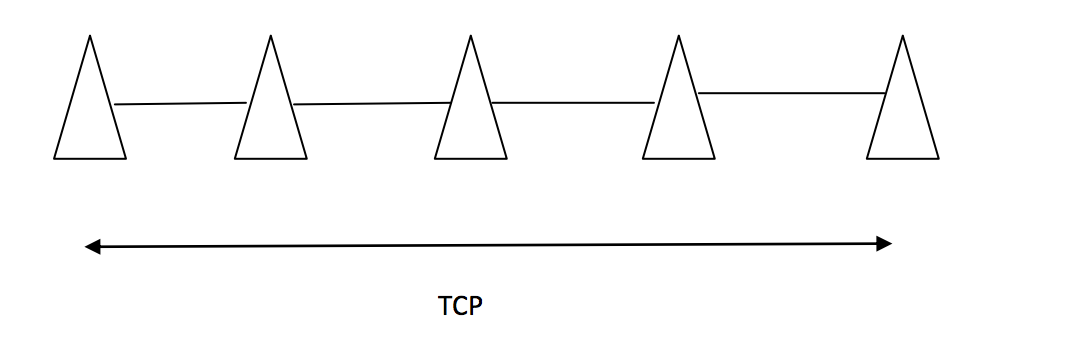
|  |  |  |  |
| --- | --- | --- | --- |
| IP header | TCP header | |  | | --- | | MPTCP header |   Payload |

Yes. TCP segments contain TLVs including the data and not only the data, and the middlebox might change something in the payload. So even it go through the middlebox well, it would be difficult to tell the original MPTCP header.

It changes the current network protocol. We need extra program to deal with this MPTCP, which is a real big overhead.

**Problem2:**

Consider the following 4 hop network path with an end-to-end TCP connection running



Assume the source (leftmost node) has 3 data packets to send and would like to send them as fast as possible. Also assume it takes one time unit for each packet to traverse a single hop (transmission time). Propagation, processing and queueing delays are negligible Finally, assume that each packet is acknowledged by the destination (the rightmost node).

a) Considering that links are all wired links, how long does it take from when the first packet starts to send until the ACK for the third packet is received. (You may assume no packet loss).

Since there is no packet loss, it would take 4 time units start from the leftmost node to rightmost mode and it needs two extra time units to send total 3 packets. So it needs 10 time units in all.

b) Now assume the links are wireless and are using WiFi. .

Determine the fastest that the transfer of the same data can take place. The fastest transfer will occur when the wireless medium is perfectly scheduled such that no collisions do occur.

Assume we send data 1,2,3 and get ACK 4,5,6. Here is how it works.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 123 |  |  |  |  |
| 1 | 23 | 1 |  |  |  |
| 2 | 23 |  | 1 |  |  |
| 3 | 3 | 2 |  | 1 |  |
| 4 | 3 |  | 2 |  | 1 |
| 5 |  | 3 |  | 2 | 4 |
| 6 |  |  | 3 |  | 45 |
| 7 |  |  |  | 3 | 45 |
| 8 |  |  |  |  | 456 |
| 9 |  |  |  | 4 | 56 |
| 10 |  |  | 4 |  | 56 |
| 11 |  | 4 |  | 5 |  |
| 12 | 4 |  | 5 |  | 6 |
| 13 | 4 | 5 |  | 6 |  |
| 14 | 45 |  | 6 |  |  |
| 15 | 45 | 6 |  |  |  |
| 16 | 456 |  |  |  |  |

So at least, we need 16 time units to finish the transmission.

c) Repeat a) and b) for a network with two hops and compare your results.

If there is only two hops. Then we only need 2 \* 2 + 2 = 6 time units for wire network.

|  |  |  |  |
| --- | --- | --- | --- |
|  | 123 |  |  |
| 1 | 23 | 1 |  |
| 2 | 23 |  | 4 |
| 3 | 3 | 2 |  |
| 4 | 3 |  | 45 |
| 5 |  | 3 |  |
| 6 |  |  | 456 |
| 7 |  | 4 | 56 |
| 8 | 4 |  | 56 |
| 9 | 4 | 5 | 6 |
| 10 | 45 |  | 6 |
| 11 | 45 | 6 |  |
| 12 | 456 |  |  |

It needs 12 time units for wireless network.