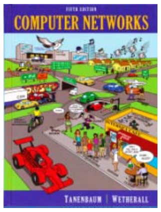
Solutions – TCP



The solutions below are based on our capture and use of tools. Your answers will differ in the details if they are based on your own capture and use of tools in a different network setting. Nonetheless, we expect our solutions to help you understand whether your answers are correct.

Step 3: TCP Segment Structure

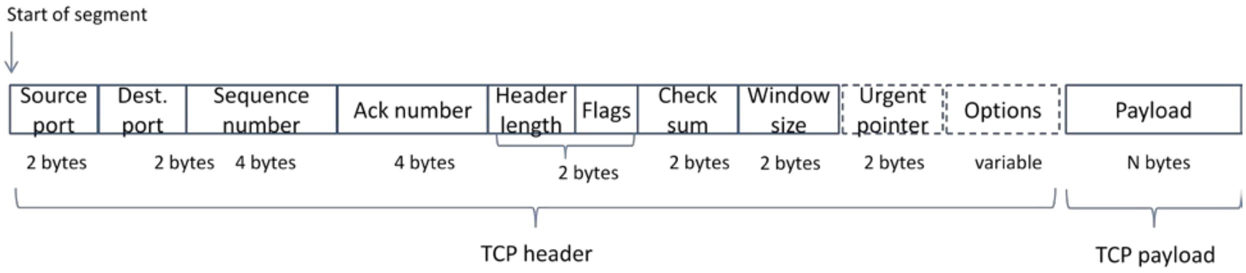


Figure 1: Structure of a TCP segment

This drawing differs from the text drawing in Fig. 6-36 in only minor respects:

- The Header length and Flags fields are combined into a 2 byte quantity. It is not easy to determine their bit lengths with Wireshark.
- The Urgent Pointer field is shown as dotted. This field is typically not used, and so does not show up in Wireshark and we do not expect you to have it in your drawing. You can notice its existence in Wireshark, however, by observing the zero bytes in the segment that are skipped over as you select the different fields.
- The Options field is shown dotted, as it may or may not be present for the segments in your trace. Most often it will be present, and when it is then its length will be a multiple of four bytes.
- The Payload is optional. It is present for the segment you viewed, but not present on an Ack-only segment, for example.

Step 4: TCP Connection Setup/Teardown

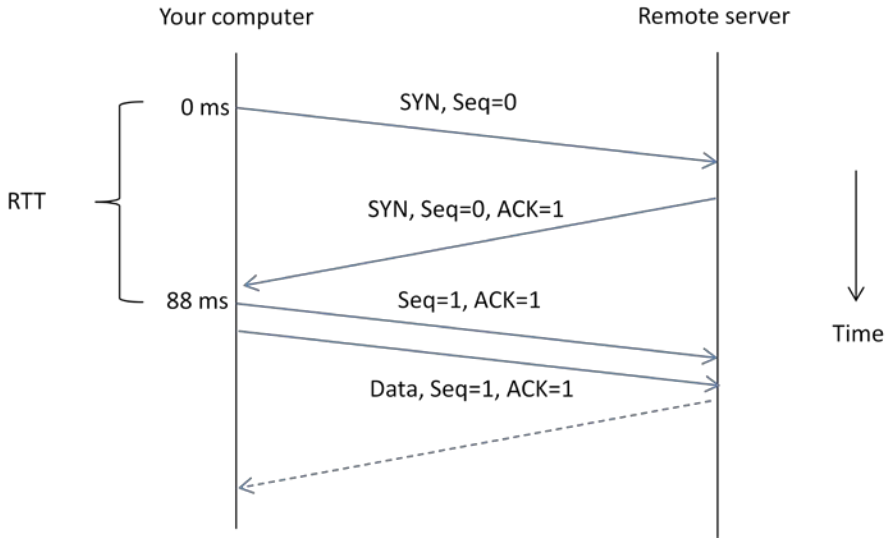


Figure 2: Time sequence diagram for the TCP three-way handshake

There are several features to note:

- The initial SYN has no ACK number, only a sequence number. All subsequent packets have ACK numbers.
- The initial sequence numbers are shown as zero in each direction. This is because our Wireshark is configured to show relative sequence numbers. The actual sequence number is some large 32-bit number, and it is different for each end.
- The ACK number is the corresponding sequence number plus 1.
- Our computer sends the third part of the handshake (the ACK) and then sends data right away in a different packet. It would be possible to combine these packets, but they are typically separate (because one is triggered by the OS and one by the application).
- For the Data segment, the sequence number and ACK stay with the previous values. The sequence number will advance as the sender sends more data. The ACK number will advance as the sender receives more data from the remote server.
- The three packets received and sent around 88ms happen very close together compared to the gap between the first and second packet. This is because they are local operations; there is no network delay involved.
- The RTT is 88ms in our trace. If you use a local web server, the RTT will be very small, likely a few milliseconds. If you use a major web server that may be provided by a content distribution network, the RTT will likely be tens of milliseconds. If you use a geographically remote server, the RTT will likely be hundreds of milliseconds.

Answers to the questions:

1. Our TCP Options are Maximum Segment Size, Window Scale, SACK permitted, and Timestamps. Each of these Options is used in both directions. There are also the NOP and End of Option List formatting options.

Here is an example of a FIN teardown:

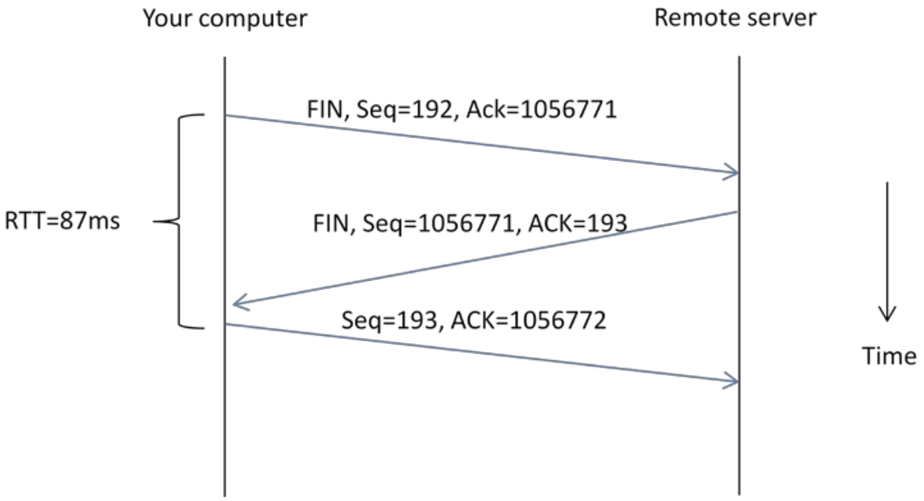


Figure 3: Time sequence diagram for FIN teardown

Points to note:

- The teardown is initiated by the computer; it might also be initiated by the server.
- Like the SYN, the FIN flag occupies one sequence number. Thus when the sequence number of the FIN is 192, the corresponding Ack number is 193.
- Your sequence numbers will vary. Our numbers are relative (as computed by Wireshark) but clearly depend on the resource that is fetched. You can tell that it is around 1 MB long.
- The RTT in the FIN exchange is similar to that in the SYN exchange, as it should be. Your RTT will vary depending on the distance between the computer and server as before.

Here is an example of a RST teardown:

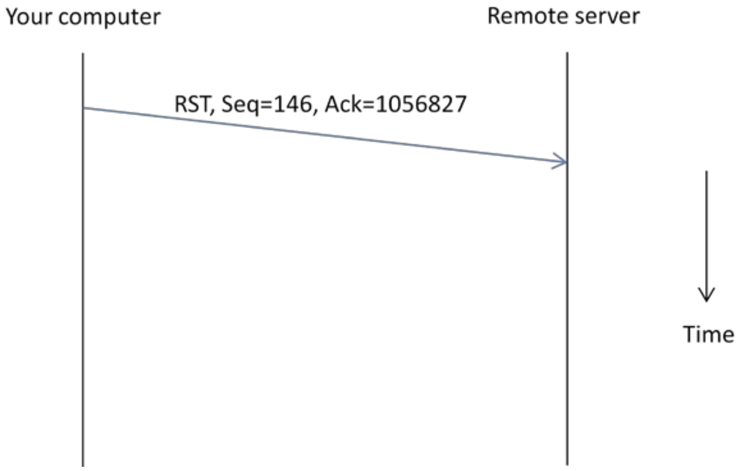


Figure 4: Time sequence diagram for RST teardown

Points to note:

- The teardown is initiated by the computer; it might also be initiated by the server.
- The teardown is abrupt – a single RST in this case, and then it is closed, which the other end must accommodate.
- The sequence and Ack numbers do not really matter here. They are simply the (relative Wireshark) values at the end of the connection.
- Since there is no round trip exchange, no RTT can be estimated.

Step 5: TCP Data Transfer

Answers to the questions:

1. Our rate is 250 packet/sec and 2.5 Mbps. Your rate will differ, but it is likely that the packet and bit rates will differ by a factor of around 10,000 for packets of size roughly 1000 bytes.
2. Our download packets are1434 bytes long, of which 1368 bytes are the TCP payload carrying contents. Thus 95% of the download is content. Your number should be similarly high for large packets of around 1 KB, since the TCP header is only 20-40 bytes.
3. Our rate is 120 packets/sec and 60,000 bits/sec. Your rate will vary, but we expect the ACK packet rate to be around half of the data packet rate for the typical pattern of one delayed ACK per two data packets received. The ACK bit rate will be at least an order of magnitude below the data bit rate because the packets are much smaller, around 60 bytes.
4. The Ack number tells the next expected sequence number. Thus it will be X plus the number of TCP payload bytes in the data segment.

[END]