

Wireshark : Three Way handshaking of TCP

Aim

Using Wireshark observe three way handshaking connection establishment, data transfer and three way handshaking connection termination in client server communication using TCP.

Theory

Wireshark

Wireshark is a network packet analyzer. A network packet analyzer will try to capture network packets and tries to display that packet data as detailed as possible. Here, we make use of the packet capturing ability of Wireshark to capture the 3 way handshaking signal packets in a TCP transmission.

What is three way handshaking and how does it work?

TCP provides reliable communication using the concept called Positive Acknowledgement with Re-transmission(PAR). A device using PAR resends the data unit until it receives an acknowledgement. If the data unit received at the receiver's end is damaged, then the receiver discards the segment. So the sender has to resend the data unit for which positive acknowledgement is not received. From this, we can understand that three segments are exchanged between sender(client) and receiver(server) for a reliable TCP connection to get established. These three steps can be explained as follows:

- Step 1(SYN): Here, the client wants to establish a connection with server, so it sends a segment with SYN(Synchronize Sequence Number) which informs the server that the client is likely to start communication and with what sequence number it starts the segments with.
- Step 2(SYN + ACK): In response, the server replies with a SYN-ACK. The acknowledgment number is set to one more than the received sequence number, and the sequence number that the server chooses for the packet is another random number.
- Step 3(ACK): Finally, the client sends an ACK back to the server. The sequence number is set to the received acknowledgement value , and the acknowledgement number is set to one more than the received sequence number.

Output

*Loopback: lo

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

→ Expression...

No.	Time	Source	Destination	Protocol	Length	Info
4	23.282126622	127.0.0.1	127.0.0.1	TCP	74	59212 → 8080 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM=1 TSval=609705113 TSecr=0 WS=...
5	23.282163631	127.0.0.1	127.0.0.1	TCP	74	8080 → 59212 [SYN, ACK] Seq=0 Ack=1 Win=65483 Len=0 MSS=65495 SACK_PERM=1 TSval=609705113 TSecr=...
6	23.282195471	127.0.0.1	127.0.0.1	TCP	66	59212 → 8080 [ACK] Seq=1 Ack=1 Win=65536 Len=0 TSval=609705113 TSecr=609705113
7	27.733534021	127.0.0.1	127.0.0.1	TCP	68	59212 → 8080 [PSH, ACK] Seq=1 Ack=1 Win=65536 Len=2 TSval=609709564 TSecr=609705113 [TCP ...
8	27.733581940	127.0.0.1	127.0.0.1	TCP	66	8080 → 59212 [ACK] Seq=1 Ack=3 Win=65536 Len=0 TSval=609709565 TSecr=609709564
9	27.733598038	127.0.0.1	127.0.0.1	TCP	68	8080 → 59212 [PSH, ACK] Seq=1 Ack=3 Win=65536 Len=2 TSval=609709565 TSecr=609709564 [TCP ...
10	27.733977873	127.0.0.1	127.0.0.1	TCP	66	59212 → 8080 [ACK] Seq=3 Ack=3 Win=65536 Len=0 TSval=609709565 TSecr=609709565
11	30.313491321	127.0.0.1	127.0.0.1	TCP	71	59212 → 8080 [PSH, ACK] Seq=3 Ack=3 Win=65536 Len=5 TSval=609712144 TSecr=609709565 [TCP ...
12	30.313530930	127.0.0.1	127.0.0.1	TCP	66	8080 → 59212 [ACK] Seq=3 Ack=8 Win=65536 Len=0 TSval=609712144 TSecr=609712144
13	30.313605887	127.0.0.1	127.0.0.1	TCP	71	8080 → 59212 [PSH, ACK] Seq=3 Ack=8 Win=65536 Len=5 TSval=609712145 TSecr=609712144 [TCP ...
14	30.313620625	127.0.0.1	127.0.0.1	TCP	66	59212 → 8080 [ACK] Seq=8 Ack=8 Win=65536 Len=0 TSval=609712145 TSecr=609712145
15	44.723536618	127.0.0.1	127.0.0.53	DNS	99	Standard query 0x6d4a A hr.client-channel.google.com OPT
16	44.723580376	127.0.0.1	127.0.0.53	DNS	99	Standard query 0xcbbc AAAA hr.client-channel.google.com OPT
17	44.724125298	127.0.0.53	127.0.0.1	DNS	115	Standard query response 0x6d4a A hr.client-channel.google.com A 74.125.130.189 OPT
18	44.724395718	127.0.0.53	127.0.0.1	DNS	127	Standard query response 0xcbbc AAAA hr.client-channel.google.com AAAA 2404:6800:4003:c01:...
19	48.073611845	127.0.0.1	127.0.0.1	TCP	66	59212 → 8080 [FIN, ACK] Seq=8 Ack=8 Win=65536 Len=0 TSval=609729905 TSecr=609712145
20	48.073735280	127.0.0.1	127.0.0.1	TCP	71	8080 → 59212 [PSH, ACK] Seq=8 Ack=9 Win=65536 Len=5 TSval=609729905 TSecr=609729905 [TCP ...
21	48.073781034	127.0.0.1	127.0.0.1	TCP	54	59212 → 8080 [RST] Seq=9 Win=0 Len=0
22	66.317975673	127.0.0.1	127.0.0.53	DNS	100	Standard query 0xfbee AAAA connectivity-check.ubuntu.com OPT

▶ Frame 20: 71 bytes on wire (568 bits), 71 bytes captured (568 bits) on interface 0

▶ Ethernet II, Src: 00:00:00:00:00:00 (00:00:00:00:00:00), Dst: 00:00:00:00:00:00 (00:00:00:00:00:00)

▶ Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1

▶ Transmission Control Protocol, Src Port: 8080, Dst Port: 59212, Seq: 8, Ack: 9, Len: 5

0000	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	E
0010	00 39 df b9 40 09 00 06 5d 03 7f 00 00 01 7f 00	9 - 0 @ -]	
0020	00 01 1f 90 e7 4c 46 39 80 2d 81 72 b6 67 80 18LF9 - r g	
0030	02 00 fe 2d 00 00 01 08 0a 24 57 bd 71 24 57-SW q\$W	
0040	bd 71 68 65 6c 6c 6f		qhello

Result

Using Wireshark observed three way handshaking connection establishment, data transfer and three way handshaking connection termination in client server communication using TCP.