# Video - TCP 3-Way Handshake (7 min)

I have some screenshots of a Wireshark packet capture that shows the process of a TCP 3-Way Handshake and the termination of a TCP conversation. Let's analyze these screenshots to get an idea of how it's working.

TCP is a connection-oriented protocol, meaning an end-to-end connection needs to be established first before data can be sent or received. The TCP 3-Way Handshake initiates that connection. When the connection finally needs to be terminated, for instance, let's say it's a connection to a web server, and you close the web browser, the connection is terminated with two 2-Way Handshakes.

A TCP 3-Way Handshake involves three steps, a [SYN], a [SYN, ACK], and an [ACK]. SYN stands for Synchronization, an ACK stands for Acknowledgment. First, the initiating host sends a Synchronization segment, the responding host sends an Acknowledgment and its own Synchronization segment, and then the initial host sends an Acknowledgment segment, hence, the [SYN], [SYN, ACK], and [ACK]. We can see that here, at the top of this screenshot. If we look at the Packet List window, at Packets 10, 11, and 12, we can see a [SYN], a [SYN, ACK], and an [ACK]. This is the 3-Way Handshake. If we look at the initial Packet in the 3-Way Handshake, the [SYN] segment up here at the top, we can see that the Sequence number is 0. The beginning of a 3-Way Handshake is Sequence number 0 because it's the first Packet in the connection or conversation between two hosts, or in this case, a host in the server. The Sequence number is actually a 32-bit random number called the ISN or Initial Sequence Number. This random number, or ISN, is chosen randomly at the beginning of each TCP conversation. This helps to protect against TCP connection hijacking attacks. Wireshark takes that 32-bit random number and converts it to 0. It then increments the Sequence numbers and the Acknowledgments from there. This makes it easier to read and follow the segments in order using the Wireshark program.

Let's look at a few of the details for this initial [SYN] segment. We go down to the Packet Details window, and we can see Sequence number: 0, that it's a (relative sequence number). If we look at the Flags, we see that the Syn bit has been Set. You can see it here with a 1. In the next Packet, Packet number 11, the server responds to the initial Synchronization segment. I'll go to the next screenshot, and now Packet 11 is highlighted. The server responds with an Acknowledgment acknowledging Sequence number 0, and sending Acknowledgment 1, so the initial Sequence number, with relative Sequence number 0, has been incremented and Acknowledgement 1 has been sent. We can see in the Protocol Details window, Acknowledgment number: 1 and that it's the (relative ack number). The server has also sent its own Synchronization segment, and that number is 0 since it's the initial conversation going the other way. If we look in the Details window, we can see that the Sequence number is 0, and that's a (relative sequence number) from the server to the host. If we look here at the Flags, both the SYN bit and the ACK bit have both been Set.

If we go to the next screenshot, in Packet 12, which is Step 3 in the 3-Way Handshake, host 10.1.1.1 responds with an Acknowledgment, or [ACK], and if we look in the Protocol Details window, we see the Acknowledgment number is 1, incrementing the server Synchronization segment by 1. You can see here that the Acknowledgment bit has been Set, but notice that the Syn bit has Not been set. This the final phase in the 3-Way Handshake.

Let's take a look at how the TCP connection terminates. I'll go to the next screenshot, and you can see, in Packet 16, the server is communicating to the host at 10.1.1.1, and has sent a segment with a Finish, or FIN, and an Acknowledgment, or ACK. In this segment, we have a [FIN, ACK]. The FIN ends the conversation. The Acknowledgment Flag has been Set since the 3-Way Handshake was first established, and in every segment sent thereafter, the Acknowledgment Flag is Set. You can see that in the next Packet, Packet number 17, the host has replied to the server with an Acknowledgment acknowledging that the conversation has ended. This is a 2-Way Handshake. A [FIN, ACK], and an [ACK]. If we look ahead, in the Packet List window, to Packet 18, you can see that host 10.1.1.1 then sends the server its own FIN and Acknowledgment, and then the server replies with its own [ACK]. So you have two 2-Way Handshakes to terminate the connection. If I go back to the previous screenshot, and take a look at the Protocol Details or the Packet Details, we can see here in the TCP segment, the Flags, notice the 1 for the Acknowledgment and then the 1 for the Finish, or Fin Flag, here that's been Set. Notice that the Acknowledgments went as high as number 374, indicating that these screenshots were probably generated from two separate Packet captures in Wireshark. You can see in these last two screenshots how the conversation ends with two 2-Way Handshakes, a [FIN, ACK], and an [ACK], and then another one going the other way.