

7 Introduction to TCP connection management

7.1 Introduction

When an application process in a client host seeks a reliable data connection with a process in another host (say, server), the client-side TCP then proceeds to establish a TCP connection with the TCP at the server side. A TCP connection is a point-to-point, full-duplex logical connection with resources allocated only in the end hosts. The TCP connection between the client and the server is established in the following manner and is illustrated in **Figure 7-1**.

1. The TCP at the client side first sends a special TCP segment, called the SYN packet, to the TCP at the server side.
2. Upon receiving the SYN packet, the server allocates TCP buffer and variables to the connection. Also, the server sends a connection-granted segment, called the SYN-ACK packet, to the TCP at the client side.
3. Upon receiving the SYN-ACK segment, the client also allocates buffers and variables to the connection. The client then acknowledges the server's connection granted segment with an ACK of its own.

This connection establishment procedure is often referred to as the three-way handshake. The special TCP segments can be identified by the values in the fields SYN, ACK and FIN in the TCP header (see **Figure 7-2**). We also note that the TCP connection is uniquely identified by the source and destination port numbers (see **Figure 7-2**) exchanged during TCP connection establishment and the source and destination IP addresses.

Once a TCP connection is established, the application processes can send data to each other. The TCP connection can be terminated by either of the two processes. Suppose that the client application process seeks to terminate the connection. Then, the following handshake ensures that the TCP connection is torn down.

1. The TCP at the client side sends a special TCP segment, called the FIN packet, to the TCP at the server side.
2. When the server receives the FIN segment, it sends the client an acknowledgement segment in return and its own FIN segment to terminate the full-duplex connection.
3. Finally, the client acknowledges the FIN-ACK segment (from the server) with an ACK of its own. At this point, all the resources (i.e., buffers and variables) in the two hosts are deallocated.

During the life of a TCP connection, the TCP protocol running in each host makes transitions through various TCP states. **Figure 7-1** illustrates the typical TCP states visited by the client and the server during connection establishment and data communication.

TCP is defined in RFCs 793, 1122, 7323 and, 2018. A recommended text book reference for TCP is Chapter 3: Transport layer, of Computer Networking: A top-down approach, by James Kurose and Keith Ross (Pearson).

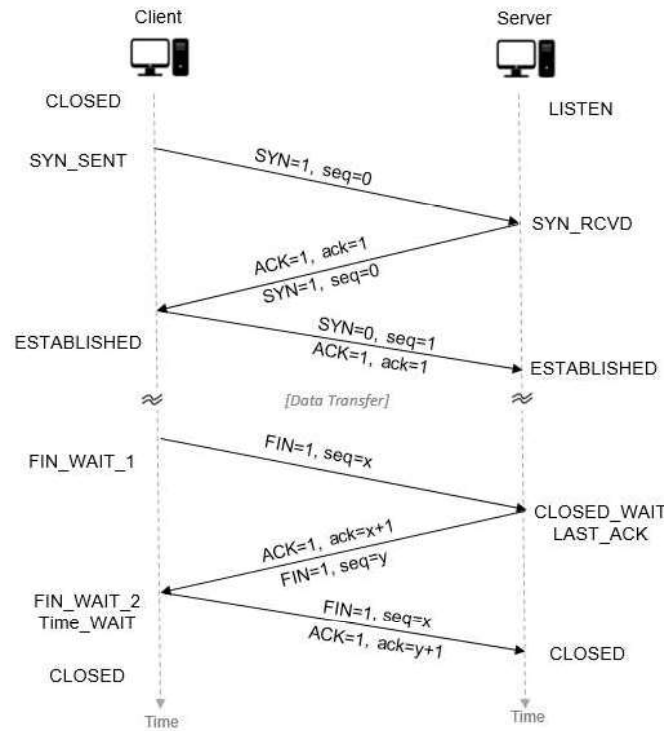


Figure 7-1: TCP connection establishment between a client and a server

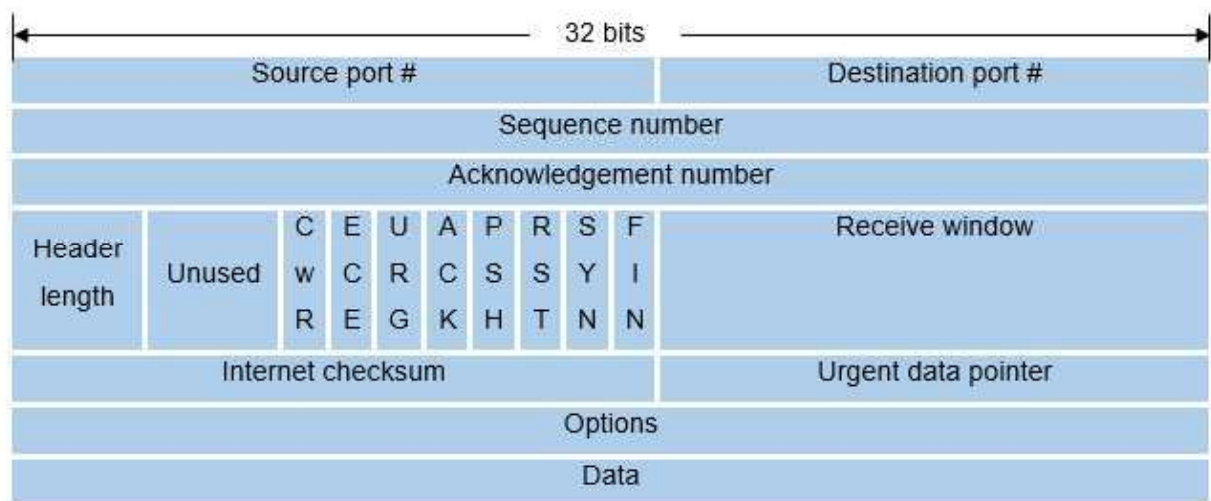


Figure 7-2: TCP Header

7.2 Network Setup

Open NetSim and click **Examples > Experiments > Introduction-to-TCP-connection-management > Sample-1** as shown below **Figure 7-3**.

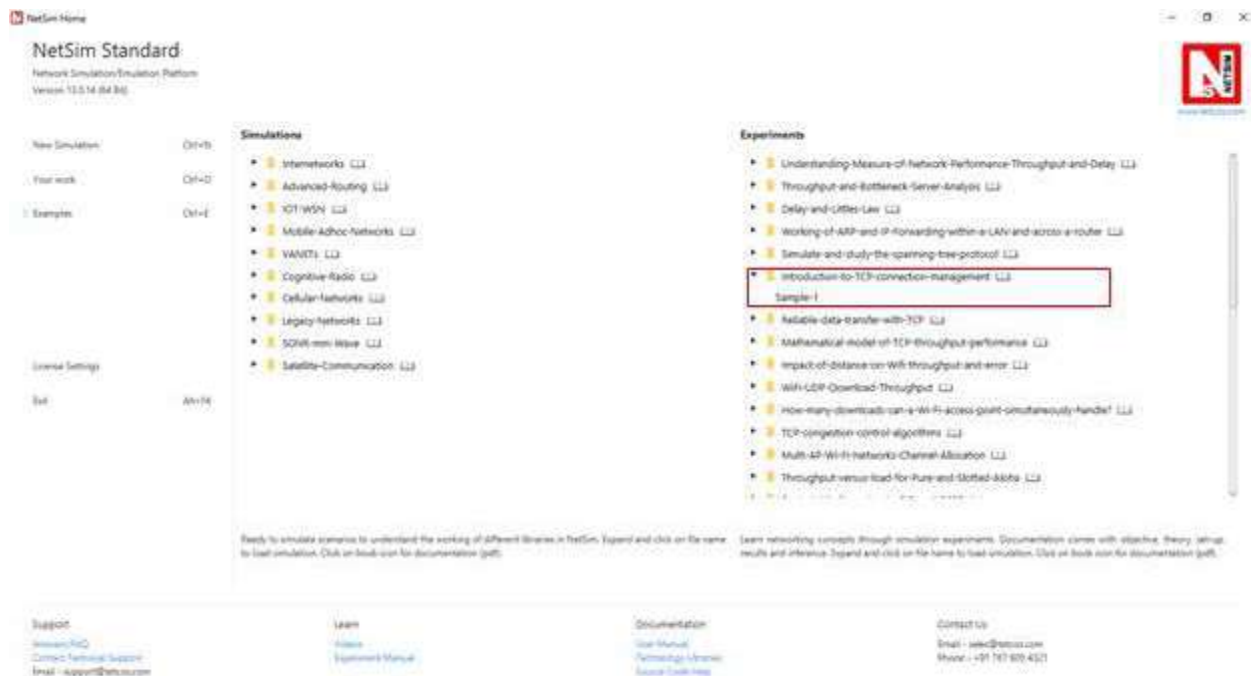


Figure 7-3: Experiments List

NetSim UI displays the configuration file corresponding to this experiment as shown below **Figure 7-4**

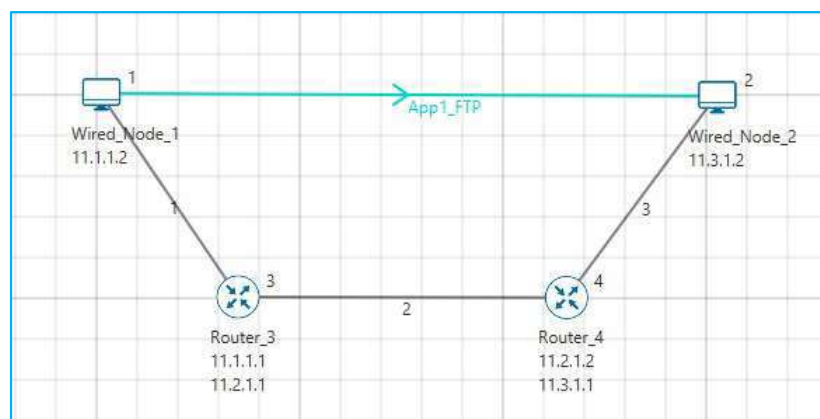


Figure 7-4: An FTP application between a client and a server connected in a network

7.3 Procedure

The following set of procedures were done to generate this sample.

Step 1: A network scenario is designed in NetSim GUI comprising of 2 Wired Nodes and 2 Routers in the **“Internetworks”** Network Library.

Step 2: In the General Properties of Wired Node 1 i.e. Source, Wireshark Capture is set to Online. Transport Layer properties Congestion plot is set to true.

Note: Accept default properties for Routers as well as the Links.

Step 3: Right-click the link ID (of a wired link) and select Properties to access the link's properties. Set Max Uplink Speed and Max Downlink Speed to **10 Mbps**. Set Uplink BER and Downlink BER to **0**. Set Uplink Propagation Delay and Downlink Propagation Delay as **100** microseconds for the links 1 and 3 (between the Wired Node's and the routers). Set Uplink Propagation Delay and Downlink Propagation Delay as **50000** microseconds for the backbone link connecting the routers, i.e., 2.

Step 4: Right click on the Application Flow **App1 FTP** and select Properties or click on the Application icon present in the top ribbon/toolbar.

An FTP Application is generated from Wired Node 1 i.e. Source to Wired Node 2 i.e. Destination with File Size set to 14600 Bytes and File Inter Arrival Time set to 10 Seconds.

Step 5: Click on Display Settings > Device IP check box in the NetSim GUI to view the network topology along with the IP address.

Step 6: Enable the plots and click on Run simulation. The simulation time is set to 10 seconds.

7.4 Output

We have enabled Wireshark capture in Wired Node 1. The PCAP file is generated at the end of the simulation and is shown in **Figure 7-5**.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	0.0.0.0	0.0.0.0	IPv4	20	
2	0.000000	11.1.1.2	11.1.1.1	TCP	44	82 → 36934 [SYN] Seq=0 Win=65535 Len=0 MSS=1460
3	0.100335	11.3.1.2	11.1.1.2	TCP	44	36934 → 82 [SYN, ACK] Seq=0 Ack=1 Win=4380 Len=0 MSS=1460
4	0.100335	11.1.1.2	11.1.1.1	TCP	40	82 → 36934 [ACK] Seq=1 Ack=1 Win=4380 Len=0
5	0.100335	11.1.1.2	11.1.1.1	TCP	1500	82 → 36934 [<None>] Seq=1 Win=4380 Len=1460
6	0.100335	11.1.1.2	11.1.1.1	TCP	1500	82 → 36934 [<None>] Seq=1461 Win=4380 Len=1460
7	0.100335	11.1.1.2	11.1.1.1	TCP	1500	82 → 36934 [<None>] Seq=2921 Win=4380 Len=1460
8	0.204208	11.3.1.2	11.1.1.2	TCP	40	36934 → 82 [ACK] Seq=1 Ack=1461 Win=4381 Len=0
9	0.204208	11.1.1.2	11.1.1.1	TCP	1500	82 → 36934 [<None>] Seq=4381 Win=5840 Len=1460
10	0.204208	11.1.1.2	11.1.1.1	TCP	1500	82 → 36934 [<None>] Seq=5841 Win=5840 Len=1460
11	0.205430	11.3.1.2	11.1.1.2	TCP	40	36934 → 82 [ACK] Seq=1 Ack=2921 Win=4381 Len=0
12	0.205430	11.1.1.2	11.1.1.1	TCP	1500	82 → 36934 [<None>] Seq=7301 Win=7300 Len=1460
13	0.205430	11.1.1.2	11.1.1.1	TCP	1500	82 → 36934 [<None>] Seq=8761 Win=7300 Len=1460
14	0.206651	11.3.1.2	11.1.1.2	TCP	40	36934 → 82 [ACK] Seq=1 Ack=4381 Win=4381 Len=0
15	0.206651	11.1.1.2	11.1.1.1	ICP	1500	82 → 36934 [<None>] Seq=10221 Win=8760 Len=1460
16	0.206651	11.1.1.2	11.1.1.1	TCP	1500	82 → 36934 [<None>] Seq=11681 Win=8760 Len=1460
17	0.308027	11.3.1.2	11.1.1.2	TCP	40	36934 → 82 [ACK] Seq=1 Ack=5841 Win=4381 Len=0
18	0.308027	11.1.1.2	11.1.1.1	TCP	1500	82 → 36934 [<None>] Seq=13141 Win=10220 Len=1460
19	0.309249	11.3.1.2	11.1.1.2	TCP	40	36934 → 82 [ACK] Seq=1 Ack=7301 Win=4381 Len=0
20	0.310471	11.3.1.2	11.1.1.2	TCP	40	36934 → 82 [ACK] Seq=1 Ack=8761 Win=4381 Len=0
21	0.311692	11.3.1.2	11.1.1.2	TCP	40	36934 → 82 [ACK] Seq=1 Ack=10221 Win=4381 Len=0
22	0.312914	11.3.1.2	11.1.1.2	TCP	40	36934 → 82 [ACK] Seq=1 Ack=11681 Win=4381 Len=0
23	0.314136	11.3.1.2	11.1.1.2	TCP	40	36934 → 82 [ACK] Seq=1 Ack=13141 Win=4381 Len=0
24	0.411846	11.3.1.2	11.1.1.2	TCP	40	36934 → 82 [ACK] Seq=1 Ack=14601 Win=4381 Len=0
25	0.411846	11.1.1.2	11.1.1.1	TCP	40	82 → 36934 [FIN] Seq=14601 Win=18980 Len=0
26	0.512161	11.3.1.2	11.1.1.2	TCP	40	36934 → 82 [FIN, ACK] Seq=1 Ack=14601 Win=4381 Len=0
27	0.512215	11.3.1.2	11.1.1.2	TCP	40	36934 → 82 [FIN] Seq=2 Win=4381 Len=0
28	0.512215	11.1.1.2	11.1.1.1	TCP	40	82 → 36934 [ACK] Seq=14602 Ack=3 Win=18980 Len=0

Figure 7-5: Wireshark Packet capture at Wired_Node_1

1. The 3-way handshake of TCP connection establishment and TCP connection termination is observed in the packet capture (Figure 7-5).

2. Data is transferred only after the TCP connection is established.
3. We can access the packet header details of the TCP segments (SYN, SYN-ACK, FIN, FINACK) in Wireshark.