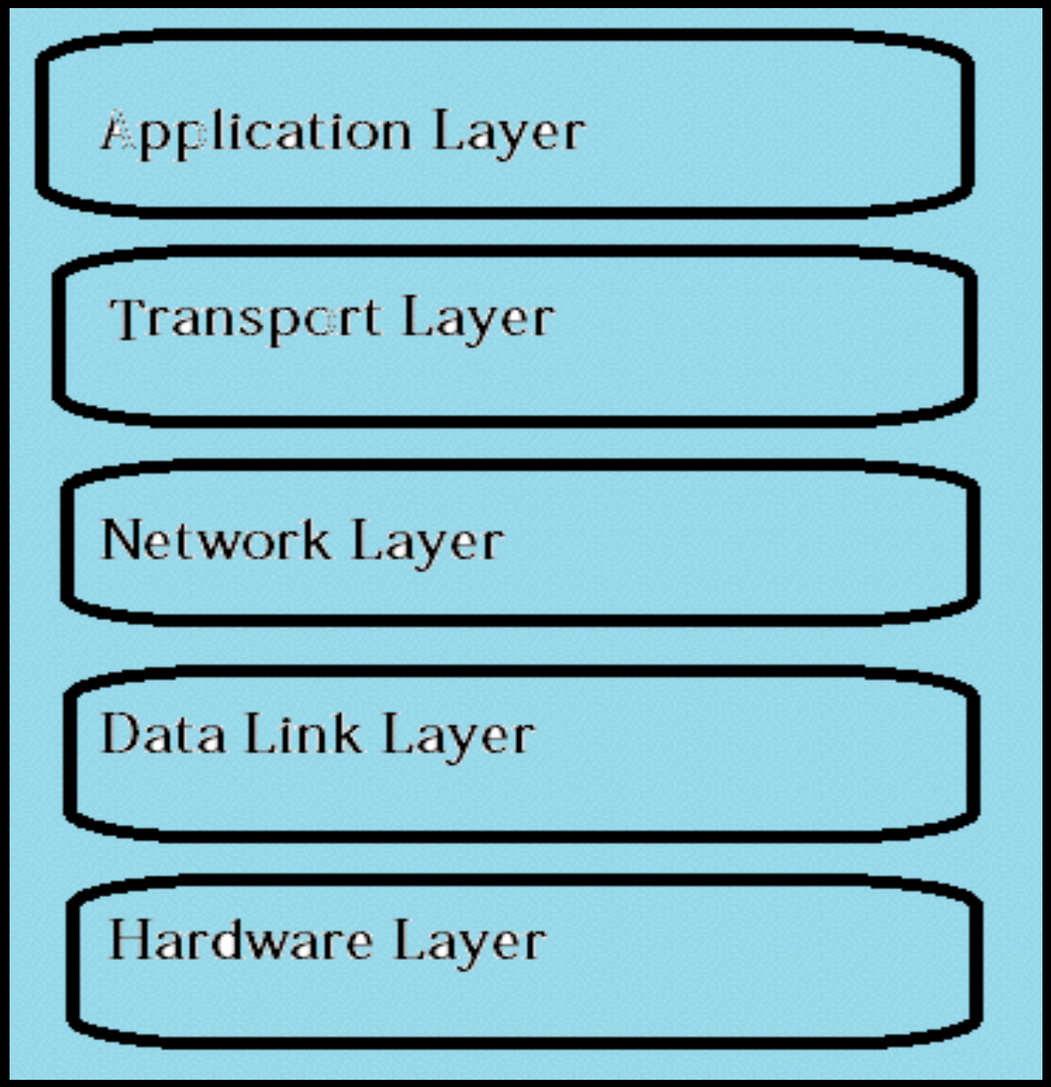
TCP 3 Way Handshake

SYN-ACK-SYN
More precise, SYN - SYN-ACK - SYN.

Situations

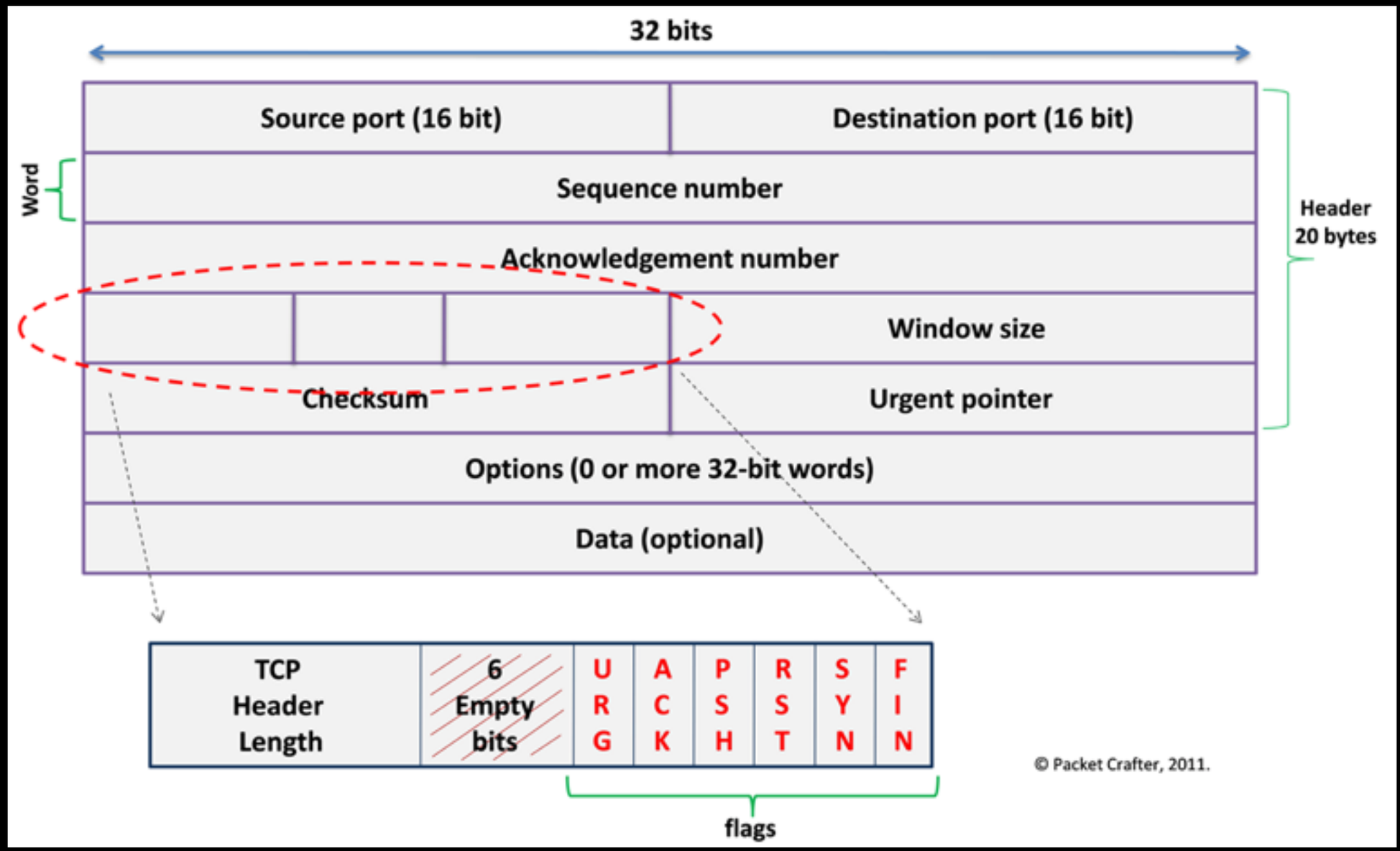
- There are two scenarios where a three-way handshake will take place:

1. Establishing a connection (an active open)
2. Terminating a connection (an active close)



Network stack

Transport layer can use TCP or UDP.
We will be discussing how TCP establishes and ends connections.

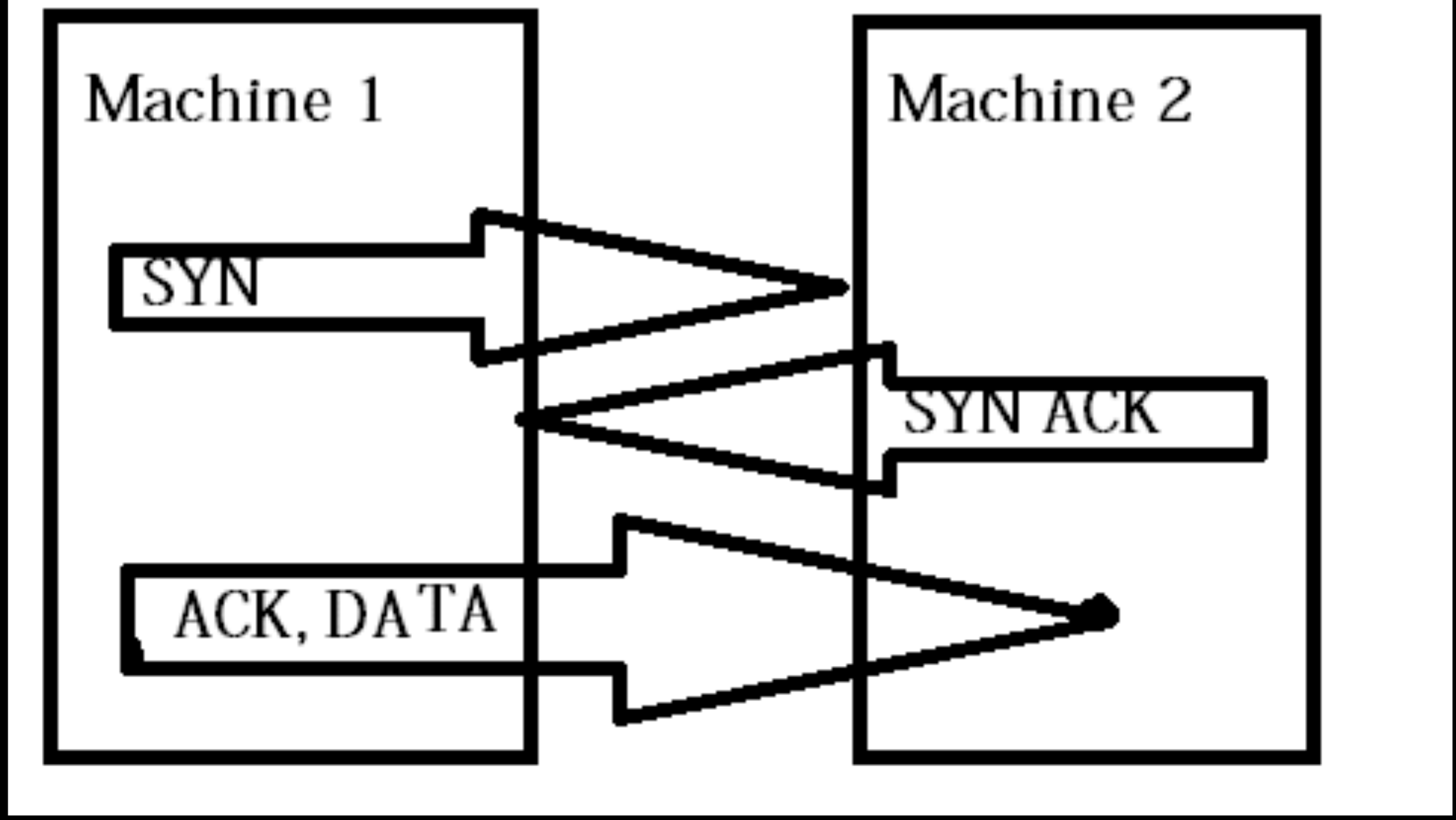


TCP Packet Format

Our focus is on the six flags, specially, - ACK, SYN and FIN.

Control Bits

- URG: Urgent Pointer field significant
- ACK: Acknowledgement field significant
- PSH: Push Function
- RST: Reset the connection
- SYN: Synchronize sequence numbers
- FIN: No more data from sender



Pictorial representation

Situation 1: Establishing a connection

Frame 1: SYN from Client

- Client sends a SYN segment (TCP ….S)
- Client specifies its initial sequence number (ISN, in our case it is 8221821), which is incremented by 1, for ex: 8221821+1=8221822, and that is sent to the server.
- To initialize a connection, the client and server must synchronize each other's sequence numbers.

```
1    2.0785 NTW3 --> BDC3 TCP ....S., len: 4, seq: 8221822-8221825, ack: 0,
win: 8192, src: 1037  dst:  139 (NBT Session)  NTW3 -->  BDC3 IP

TCP: ....S., len: 4, seq: 8221822-8221825, ack: 0, win: 8192, src: 1037
dst:  139 (NBT Session)

   TCP: Source Port = 0x040D
   TCP: Destination Port = NETBIOS Session Service
   TCP: Sequence Number = 8221822 (0x7D747E)
   TCP: Acknowledgement Number = 0 (0x0)
   TCP: Data Offset = 24 (0x18)
   TCP: Reserved = 0 (0x0000)
   TCP: Flags = 0x02 : ....S.

      TCP: ..0..... = No urgent data
      TCP: ...0.... = Acknowledgement field not significant
      TCP: ....0... = No Push function
      TCP: .....0.. = No Reset
      TCP: ......1. = Synchronize sequence numbers
      TCP: .......0 = No Fin

   TCP: Window = 8192 (0x2000)
   TCP: Checksum = 0xF213
   TCP: Urgent Pointer = 0 (0x0)
   TCP: Options

         TCP: Option Kind (Maximum Segment Size) = 2 (0x2)
         TCP: Option Length = 4 (0x4)
         TCP: Option Value = 1460 (0x5B4)

   TCP: Frame Padding

00000:  02 60 8C 9E 18 8B 02 60 8C 3B 85 C1 08 00 45 00   .`.....`.;....E.
00010:  00 2C 0D 01 40 00 80 06 E1 4B 83 6B 02 D6 83 6B   .,..@....K.k...k
00020:  02 D3 04 0D 00 8B 00 7D 74 7E 00 00 00 00 60 02   .......}t~....`.
00030:  20 00 F2 13 00 00 02 04 05 B4 20 20                .........
```

Frame 1: SYN from Client(NTW3)

Frame 2: SYN-ACK from Server

- Server sends an ACK and a SYN on this segment (TCP .A..S.)
- It is doing two things:
  1. Acknowledging the request of the client for synchronization.
  2. The server is also sending its request to the client for synchronization of its sequence numbers.
- The process of acknowledging the client's request allows the server to increment the client's sequence number by one and uses it as its acknowledgement number.

```
2   2.0786 BDC3 --> NTW3  TCP .A..S., len: 4, seq: 1109645-1109648, ack:
8221823, win: 8760, src: 139 (NBT Session)  dst: 1037 BDC3 --> NTW3  IP

TCP: .A..S., len:    4, seq:   1109645-1109648, ack:   8221823, win: 8760,
src:  139 (NBT Session)  dst: 1037

   TCP: Source Port = NETBIOS Session Service
   TCP: Destination Port = 0x040D
   TCP: Sequence Number = 1109645 (0x10EE8D)
   TCP: Acknowledgement Number = 8221823 (0x7D747F)
   TCP: Data Offset = 24 (0x18)
   TCP: Reserved = 0 (0x0000)
   TCP: Flags = 0x12 : .A..S.

      TCP: ..0..... = No urgent data
      TCP: ...1.... = Acknowledgement field significant
      TCP: ....0... = No Push function
      TCP: .....0.. = No Reset
      TCP: ......1. = Synchronize sequence numbers
      TCP: .......0 = No Fin

   TCP: Window = 8760 (0x2238)
   TCP: Checksum = 0x012D
   TCP: Urgent Pointer = 0 (0x0)
   TCP: Options

         TCP: Option Kind (Maximum Segment Size) = 2 (0x2)
         TCP: Option Length = 4 (0x4)
         TCP: Option Value = 1460 (0x5B4)

   TCP: Frame Padding

00000:  02 60 8C 3B 85 C1 02 60 8C 9E 18 8B 08 00 45 00   .`.;...`......E.
00010:  00 2C 5B 00 40 00 80 06 93 4C 83 6B 02 D3 83 6B   .,[.@....L.k...k
00020:  02 D6 00 8B 04 0D 00 10 EE 8D 00 7D 74 7F 60 12   ...........}t`.
00030:  22 38 01 2D 00 00 02 04 05 B4 20 20               "8.-......
```

Frame 2: SYN-ACK from Server(BDC3)

Frame 3: ACK from Client

- Client sends ACK on this segment (TCP .A….).
- The client is acknowledging the request from the server for synchronization.
- The client uses the same algorithm the server implemented in providing an acknowledgement number.
- The client's acknowledgment of the server's request for synchronization completes the process of establishing a reliable connection, thus the three-way handshake.

```
3   2.787 NTW3 --> BDC3  TCP .A...., len: 0, seq: 8221823-8221823, ack:
1109646, win: 8760, src: 1037  dst:  139 (NBT Session)  NTW3 --> BDC3  IP

TCP: .A...., len:    0, seq:   8221823-8221823, ack:   1109646, win: 8760,
src: 1037  dst:  139 (NBT Session)

   TCP: Source Port = 0x040D
   TCP: Destination Port = NETBIOS Session Service
   TCP: Sequence Number = 8221823 (0x7D747F)
   TCP: Acknowledgement Number = 1109646 (0x10EE8E)
   TCP: Data Offset = 20 (0x14)
   TCP: Reserved = 0 (0x0000)
   TCP: Flags = 0x10 : .A....

      TCP: ..0..... = No urgent data
      TCP: ...1.... = Acknowledgement field significant
      TCP: ....0... = No Push function
      TCP: .....0.. = No Reset
      TCP: ......0. = No Synchronize
      TCP: .......0 = No Fin

   TCP: Window = 8760 (0x2238)
   TCP: Checksum = 0x18EA
   TCP: Urgent Pointer = 0 (0x0)
   TCP: Frame Padding

00000:  02 60 8C 9E 18 8B 02 60 8C 3B 85 C1 08 00 45 00   .`.....`.;....E.
00010:  00 28 0E 01 40 00 80 06 E0 4F 83 6B 02 D6 83 6B   .(..@....O.k...k
00020:  02 D3 04 0D 00 8B 00 7D 74 7F 00 10 EE 8E 50 10   .......}t....P.
00030:  22 38 18 EA 00 00 20 20 20 20 20 20               "8....
```

Frame 3:ACK from Client(NTW3)

Situation 2:Terminating a Connection

- Although the three-way handshake only requires three packets to be transmitted over our networked media, the termination of this reliable connection will necessitate the transmission of four packets.

- Because a TCP connection is full duplex (that is, data can be flowing in each direction independent of the other), each direction must be terminated independently.

Frame 4: ACK-FIN by client

- Client a FIN that is accompanied by an ACK (TCP .A…F)
- This segment has two functions:

1. FIN: It will inform the server that it has no more data to send

2. ACK: It is essential in identifying the specific connection they have established

```
4   16.0279 NTW3 --> BDC3 TCP .A...F, len: 0, seq: 8221823-8221823,
ack:3462835714, win: 8760, src: 2337  dst: 139 (NBT Session)  NTW3 --> BDC3
IP

TCP: .A...F, len:   0, seq: 8221823-8221823, ack:  1109646, win: 8760, src:
1037  dst:  139 (NBT Session)

   TCP: Source Port = 0x040D
   TCP: Destination Port = NETBIOS Session Service
   TCP: Sequence Number = 8221823 (0x7D747F)
   TCP: Acknowledgement Number = 1109646 (0x10EE8E)
   TCP: Data Offset = 20 (0x14)
   TCP: Reserved = 0 (0x0000)
   TCP: Flags = 0x11 : .A...F

      TCP: ..0..... = No urgent data
      TCP: ...1.... = Acknowledgement field significant
      TCP: ....0... = No Push function
      TCP: .....0.. = No Reset
      TCP: ......0. = No Synchronize
      TCP: .......1 = No more data from sender

   TCP: Window = 8760 (0x2238)
   TCP: Checksum = 0x236C
   TCP: Urgent Pointer = 0 (0x0)

00000:  00 20 AF 47 93 58 00 A0 C9 22 F5 39 08 00 45 00   . .G.X..."9..E.
00010:  00 28 9B F5 40 00 80 06 21 4A C0 5E DE 7B C0 5E   .(..@...!J.^.{.^
00020:  DE 57 09 21 05 48 0B 20 96 AC CE 66 AE 02 50 11   .W.!.H. ...f..P.
00030:  22 38 23 6C 00 00                                 "8#l..
```

Frame 4: ACK-FIN by client

Frame 5: ACK from Server

- The server acknowledges the FIN that was transmitted from the client.

```
5    16.0281 BDC3 --> NTW3 TCP .A...., len:    0, seq: 1109646-1109646,
ack: 8221824, win:28672, src: 139  dst: 2337 (NBT Session) BDC3 -->  NTW3
IP

TCP: .A...., len:    0, seq: 1109646-1109646, ack: 8221824, win:28672, src:
139  dst: 2337 (NBT Session)

   TCP: Source Port = 0x040D
   TCP: Destination Port = NETBIOS Session Service
   TCP: Sequence Number = 1109646 (0x10EE8E)
   TCP: Acknowledgement Number = 8221824 (0x7D7480)
   TCP: Data Offset = 20 (0x14)
   TCP: Reserved = 0 (0x0000)
   TCP: Flags = 0x10 : .A....

      TCP: ..0..... = No urgent data
      TCP: ...1.... = Acknowledgement field significant
      TCP: ....0... = No Push function
      TCP: .....0.. = No Reset
      TCP: ......0. = No Synchronize
      TCP: .......0 = No Fin

   TCP: Window = 28672 (0x7000)
   TCP: Checksum = 0xD5A3
   TCP: Urgent Pointer = 0 (0x0)
   TCP: Frame Padding

00000:  00 A0 C9 22 F5 39 08 00 02 03 BA 84 08 00 45 00   ...".9........E.
00010:  00 28 D2 82 00 00 3F 06 6B BD C0 5E DE 57 C0 5E   .(....?.k..^.W.^
00020:  DE 7B 05 48 09 21 CE 66 AE 02 0B 20 96 AD 50 10   .{.H.!.f... ..P.
00030:  70 00 D5 A3 00 00 90 00 01 00 86 00               p...........
```

Frame 5: ACK from Server

Frame 6: ACK-FIN from Server

- Even though TCP has established connections between the two computers, the connections are still independent of one another.
- Therefore, the server must also transmit a FIN (TCP .A...F) to the client.
- The acknowledgement number stays the same.

```
6   17.0085 BDC3 --> NTW3 TCP .A...F, len: 0, seq: 1109646-1109646, ack:
8221824, win:28672, src: 139 dst: 2337 (NBT Session) BDC3 -->  NTW3   IP

TCP: .A...F, len:  0, seq: 1109646-1109646, ack: 8221824, win:28672, src:
139  dst: 2337 (NBT Session)

   TCP: Source Port = 0x0548
   TCP: Destination Port = 0x0921
   TCP: Sequence Number = 1109646 (0x10EE8E)
   TCP: Acknowledgement Number = 8221824 (0x7D7480)
   TCP: Data Offset = 20 (0x14)
   TCP: Reserved = 0 (0x0000)
   TCP: Flags = 0x11 : .A...F

      TCP: ..0..... = No urgent data
      TCP: ...1.... = Acknowledgement field significant
      TCP: ....0... = No Push function
      TCP: .....0.. = No Reset
      TCP: ......0. = No Synchronize
      TCP: .......1 = No more data from sender

   TCP: Window = 28672 (0x7000)
   TCP: Checksum = 0xD5A2
   TCP: Urgent Pointer = 0 (0x0)
   TCP: Frame Padding

00000:  00 A0 C9 22 F5 39 08 00 02 03 BA 84 08 00 45 00   ...".9........E.
00010:  00 28 D2 94 00 00 3F 06 6B AB C0 5E DE 57 C0 5E   .(....?.k..^.W.^
00020:  DE 7B 05 48 09 21 CE 66 AE 02 0B 20 96 AD 50 11   .{.H.!.f... ..P.
00030:  70 00 D5 A2 00 00 02 04 05 B4 86 00               p...........
```

Frame 6: ACK-FIN from Server

Frame 7: ACK from Client

- The client responds in the same format as the server, by ACKing the server's FIN and incrementing the sequence number by 1.

```
7   17.0085 NTW3 --> BDC3 TCP .A...., len: 0, seq: 8221824-8221824, ack:
1109647, win: 8760, src: 2337  dst: 139 (NBT Session) NTW3 --> BDC3 IP

TCP: .A...., len:    0, seq: 8221824-8221824, ack: 1109647, win: 8760, src:
2337  dst: 139   (NBT Session)

   TCP: Source Port = 0x0921
   TCP: Destination Port = 0x0548
   TCP: Sequence Number = 8221824 (0x7D7480)
   TCP: Acknowledgement Number = 1109647 (0x10EE8F)
   TCP: Data Offset = 20 (0x14)
   TCP: Reserved = 0 (0x0000)
   TCP: Flags = 0x10 : .A....

      TCP: ..0..... = No urgent data
      TCP: ...1.... = Acknowledgement field significant
      TCP: ....0... = No Push function
      TCP: .....0.. = No Reset
      TCP: ......0. = No Synchronize
      TCP: .......0 = No Fin

   TCP: Window = 8760 (0x2238)
   TCP: Checksum = 0x236B
   TCP: Urgent Pointer = 0 (0x0)

00000:  00 20 AF 47 93 58 00 A0 C9 22 F5 39 08 00 45 00   . .G.X..."9..E.
00010:  00 28 BA F5 40 00 80 06 02 4A C0 5E DE 7B C0 5E   .(..@....J.^.{.^
00020:  DE 57 09 21 05 48 0B 20 96 AD CE 66 AE 03 50 10   .W.!.H. ...f..P.
00030:  22 38 23 6B 00 00                                 "8#k..
```

Frame 7: ACK from Client

Credits: http://support.microsoft.com/kb/172983

–Arnav Sharma