### The segments exchanged in the 3-way handshake

#### First 3-way handshake segment.

It can be identified by the SYN flag as well as the fact that it is the first segment which comes from the client (the process of establishing communication).

```
Frame 38: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface \Device\NPF_{02AAA61F-F837-4BA5-S}

Ethernet II, Src: ASUSTekCOMPU_b0:49:5b (18:31:bf:b0:49:5b), Dst: Intel_84:04:45 (c8:8a:9a:84:04:45)

Internet Protocol Version 4, Src: 192.168.87.112, Dst: 192.168.87.181
  Transmission Control Protocol, Src Port: 53766, Dst Port: 1235, Seq: 0, Len: 0
      Source Port: 53766
      Destination Port: 1235
      [Stream index: 2]
      [Stream Packet Number: 1]
      [Conversation completeness: Incomplete, DATA (15)]
      [TCP Segment Len: 0]
      Sequence Number: 0 (relative : Sequence Number (raw): 473755703
                                (relative sequence number)
      [Next Sequence Number: 1 (relative sequence number)]
      Acknowledgment Number: 0
      Acknowledgment number (raw): 0
      1000 .... = Header Length: 32 bytes (8)
   Flags: 0x002 (SYN)
      Window: 65535
      [Calculated window size: 65535]
      Checksum: 0x5b4c [unverified]
      [Checksum Status: Unverified]
      Urgent Pointer: 0
    > Options: (12 bytes), Maximum segment size, No-Operation (NOP), Window scale, No-Operation (NOP), No-Operation (
    | [Timestamps]
```

#### Second 3-way handshake segment.

The second segment in the process has a SYN, ACK flag and comes from the server (source port is 1235).

```
Frame 40: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface \Device\NPF_{02AAA61F-F837-4BA5-5}

Ethernet II, Src: Intel_84:04:45 (c8:8a:9a:84:04:45), Dst: ASUSTekCOMPU_b0:49:5b (18:31:bf:b0:49:5b)

Internet Protocol Version 4, Src: 192.168.87.181, Dst: 192.168.87.112

Transmission Control Protocol, Src Port: 1235, Dst Port: 53766, Seq: 0, Ack: 1, Len: 0

Source Port: 1235

Destination Port: 53766

[Stream index: 2]

[Conversation completeness: Incomplete, DATA (15)]

[TCP Segment Len: 0]

Sequence Number: 0 (relative sequence number)

Sequence Number (raw): 293795432

[Next Sequence Number: 1 (relative sequence number)]

Acknowledgment number: 1 (relative ack number)

Acknowledgment number: 473755704

1000 ... = Header Length: 32 bytes (8)

Flags: 0x012 (SYN, ACK)

Window: 65535

[Calculated window size: 65535]

Checksum: 0x309d [unverified]

Ungent Pointer: 0

Options: (12 bytes), Maximum segment size, No-Operation (NOP), Window scale, No-Operation (NOP), No-Operation (SEQ/ACK analysis]
```

#### Third 3-way handshake segment (the last one).

The 3<sup>rd</sup> segment in the process has ACK flag and comes from the client.

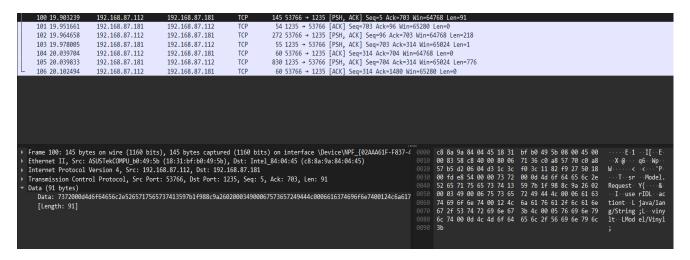
```
Frame 42: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface \Device\NPF_{02AAA61F-F837-4BA5-5} Ethernet II, Src: ASUSTekCOMPU_b0:49:5b (18:31:bf:b0:49:5b), Dst: Intel_84:04:45 (c8:8a:9a:84:04:45) Internet Protocol Version 4, Src: 192.168.87.112, Dst: 192.168.87.181
Transmission Control Protocol, Src Port: 53766, Dst Port: 1235, Seq: 1, Ack: 1, Len: 0
   Source Port: 53766
   Destination Port: 1235
    [Stream index: 2]
    [Stream Packet Number: 3]
 ▶ [Conversation completeness: Incomplete, DATA (15)]
    [TCP Segment Len: 0]
    Sequence Number: 1
                               (relative sequence number)
    Sequence Number (raw): 473755704
   [Next Sequence Number: 1 (relative sequence number)]
Acknowledgment Number: 1 (relative ack number)
   Acknowledgment number (raw): 293795433
   0101 .... = Header Length: 20 bytes (5)
   Flags: 0x010 (ACK)
   Window: 255
    [Calculated window size: 65280]
    [Window size scaling factor: 256]
   Checksum: 0x9324 [unverified]
    [Checksum Status: Unverified]
   Urgent Pointer: 0
   [Timestamps]
   [SEQ/ACK analysis]
```

## IP addresses and port numbers associated with the TCP socket established between client and server

Internet Protocol Version 4 section the source and the destination IP addresses are different as well as the ports in the Transmission Control Protocol section.

# Application layer "borrow" request message from client and response message from server

Inside the data section, it can be seen that this is the request message sent by the client (source IP address and port number).



The same situation for the server response.

