# Transport Layer Protocols



# Objectives

Upon completion of this section, trainees will be able to:

- Describe the common differences between TCP and UDP.
- Describe the forms of data to which TCP and UDP are applied.
- Identify well known TCP and UDP based port numbers.

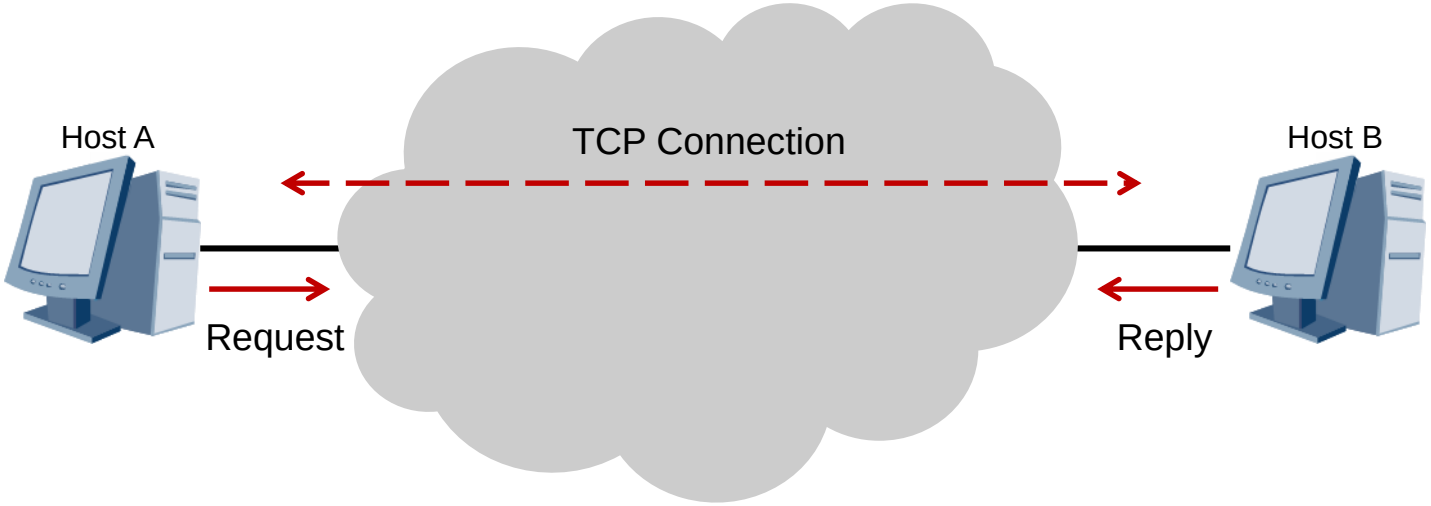
# Foreword

Tra le funzioni più importanti del livello 4 ISO/OSI vi sono il processo di **error-recovery** e il **controllo di flusso**.

In questo livello abbiamo due protocolli principali che sono TCP (orientato alla connessione) e UDP (senza connessione). Un protocollo orientato alla connessione richiede uno scambio di messaggi PRIMA che inizi lo scambio dei dati. Un protocollo senza connessione inizia a scambiare dati subito.

Vengono usati i numeri di porta per scambiare i dati tra applicazioni in esecuzione su calcolatori remoti collegati alla rete.

# Transmission Control Protocol

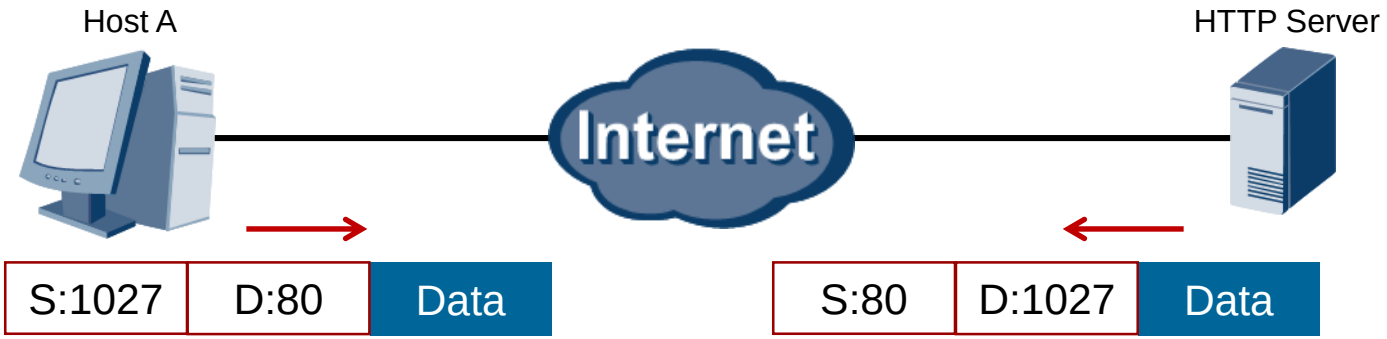



- A connection is established before data is sent.

# Transmission Control Protocol

| | | | | |
|---|---|---|---|---|
| Eth | IP | UDP | Data | Eth |
| Eth | IP | TCP | Web page | Eth |
| Eth | IP | TCP | Wire transfer | Eth |

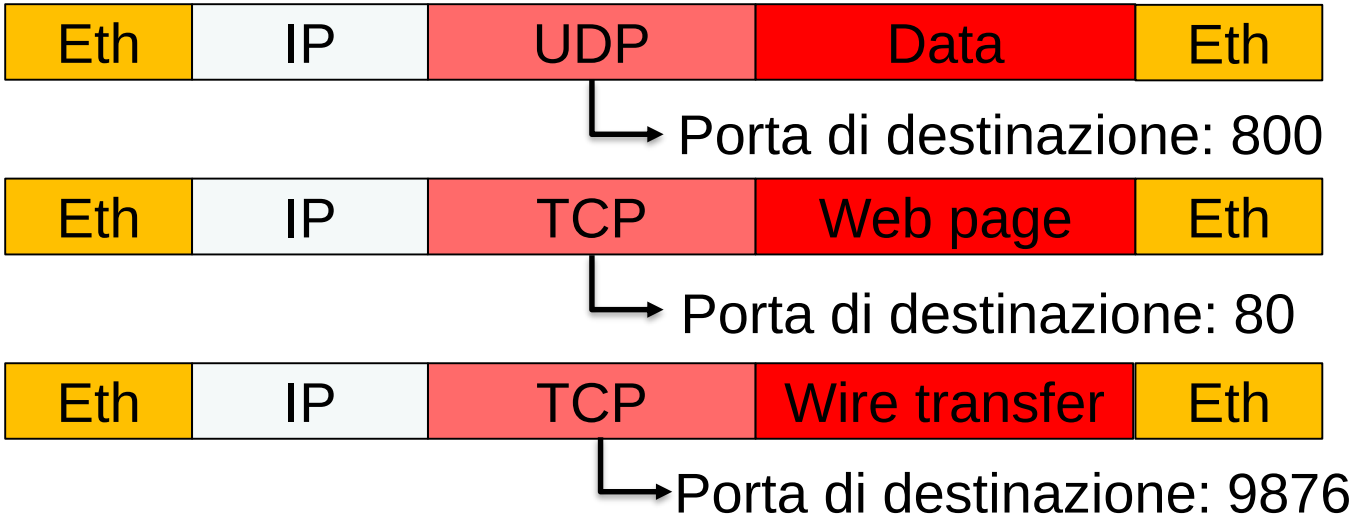
# TCP Ports




| Protocol | Port |
|---|---|
| FTP | 20 - 21 |
| HTTP | 80 |
| TELNET | 23 |
| SMTP | 25 |

- Ports represent individual services such as those listed above.

# Transmission Control Protocol

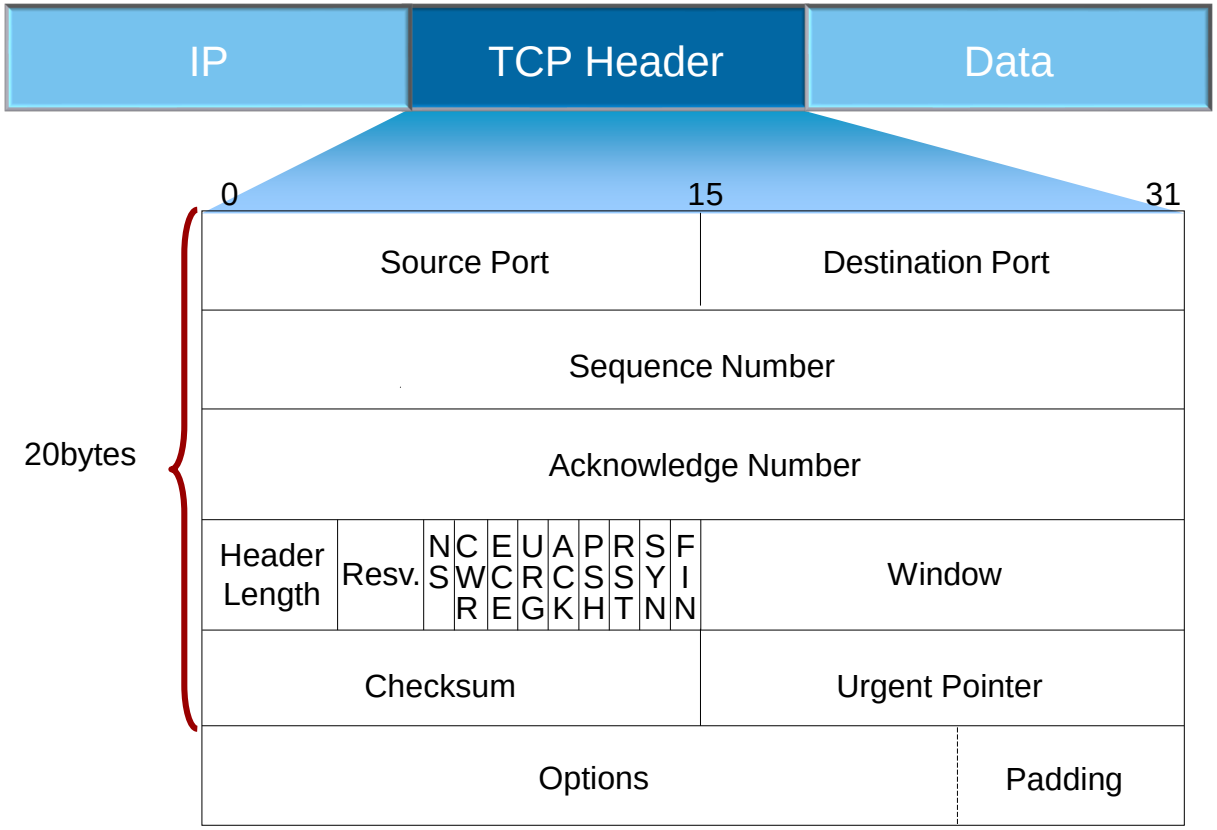







HUAWEI

# TCP Header

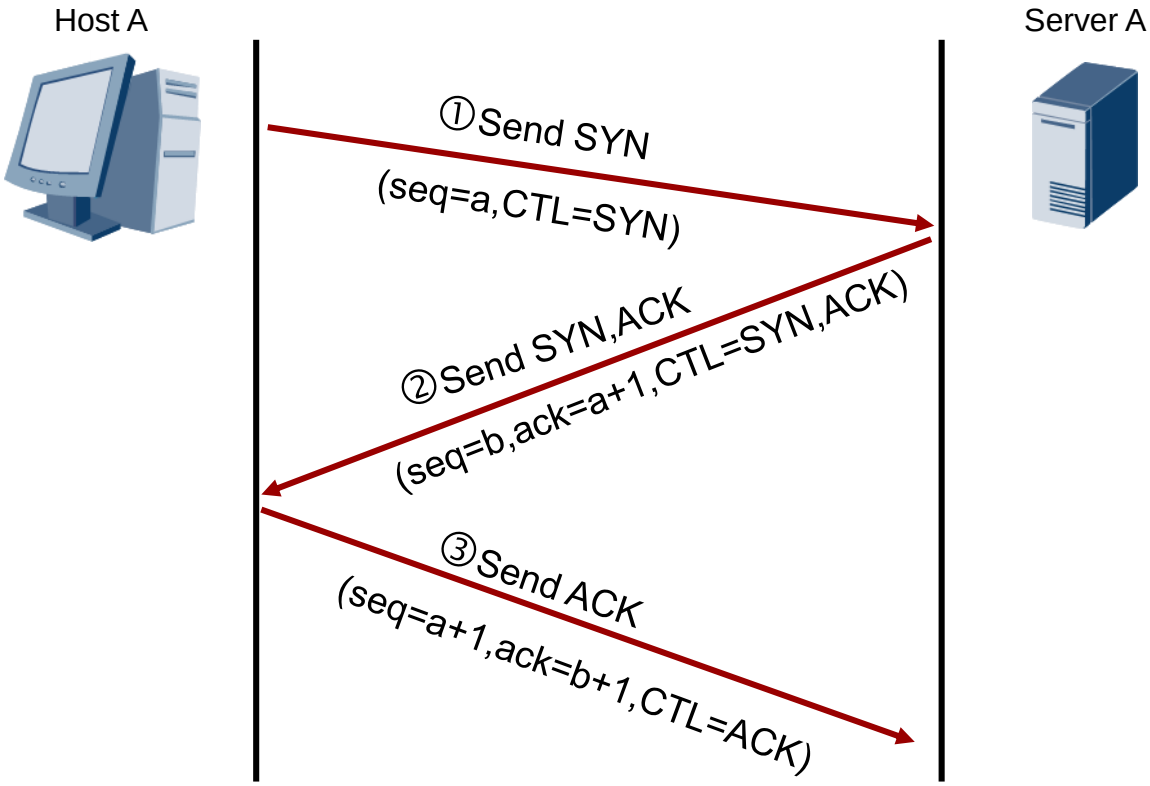






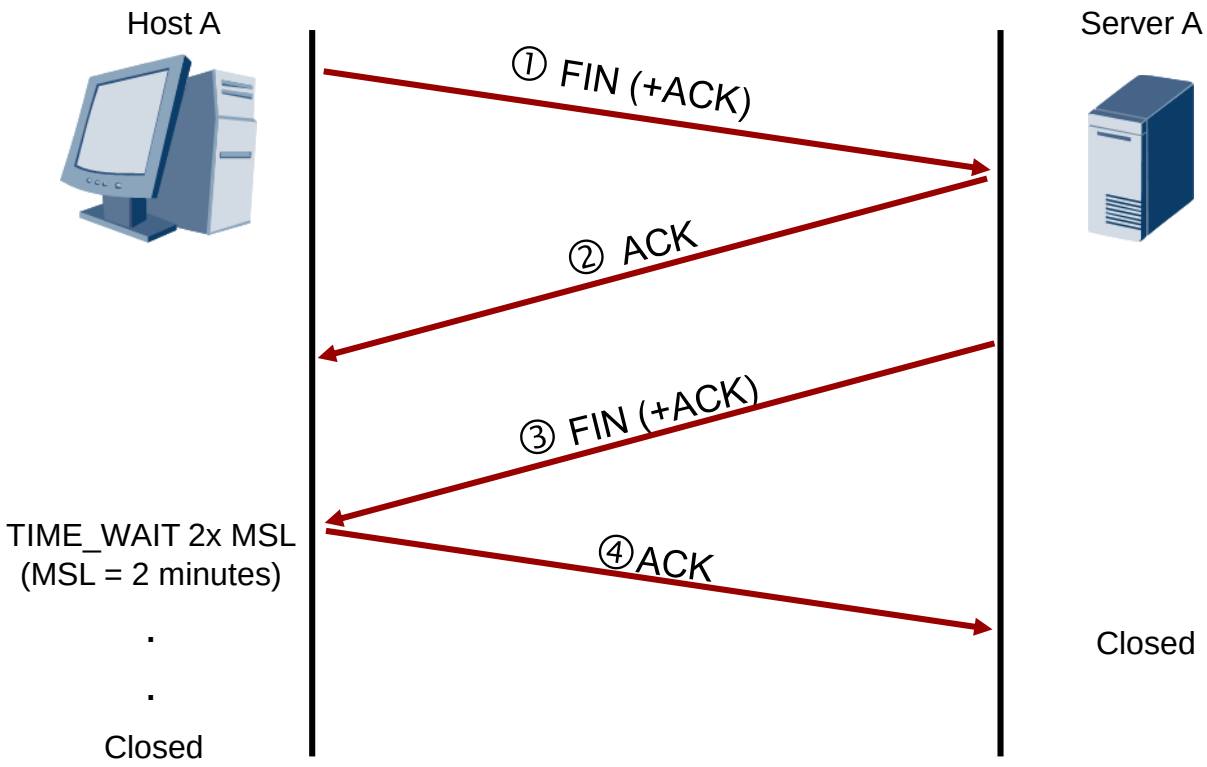

HUAWEI

# TCP Connection Establishment




- A TCP connection is established after a three-way handshake.

# TCP Connection Termination




- Host A will ensure ACK is received by Server A before closing.

# TCP Connection Establishment

**LISTEN**: in attesa di una connessione da un host e porta remoti;

**SYN-SENT**: in attesa di una richiesta di connessione in risposta a quella che è stata appena inviata;

**SYN-RECEIVED**: in attesa di un ack dopo avere inviato e ricevuto una richiesta di connessione;

**ESTABLISHED**: connessione aperta e dati consegnabili all'utente finale;

**FIN-WAIT-1**: in attesa di una richiesta di terminazione della connessione dal peer remoto o di un ack alla richiesta di terminazione precedente;

**FIN-WAIT-2**: in attesa di una richiesta di terminazione della connessione dal peer remoto.

# TCP Connection Establishment

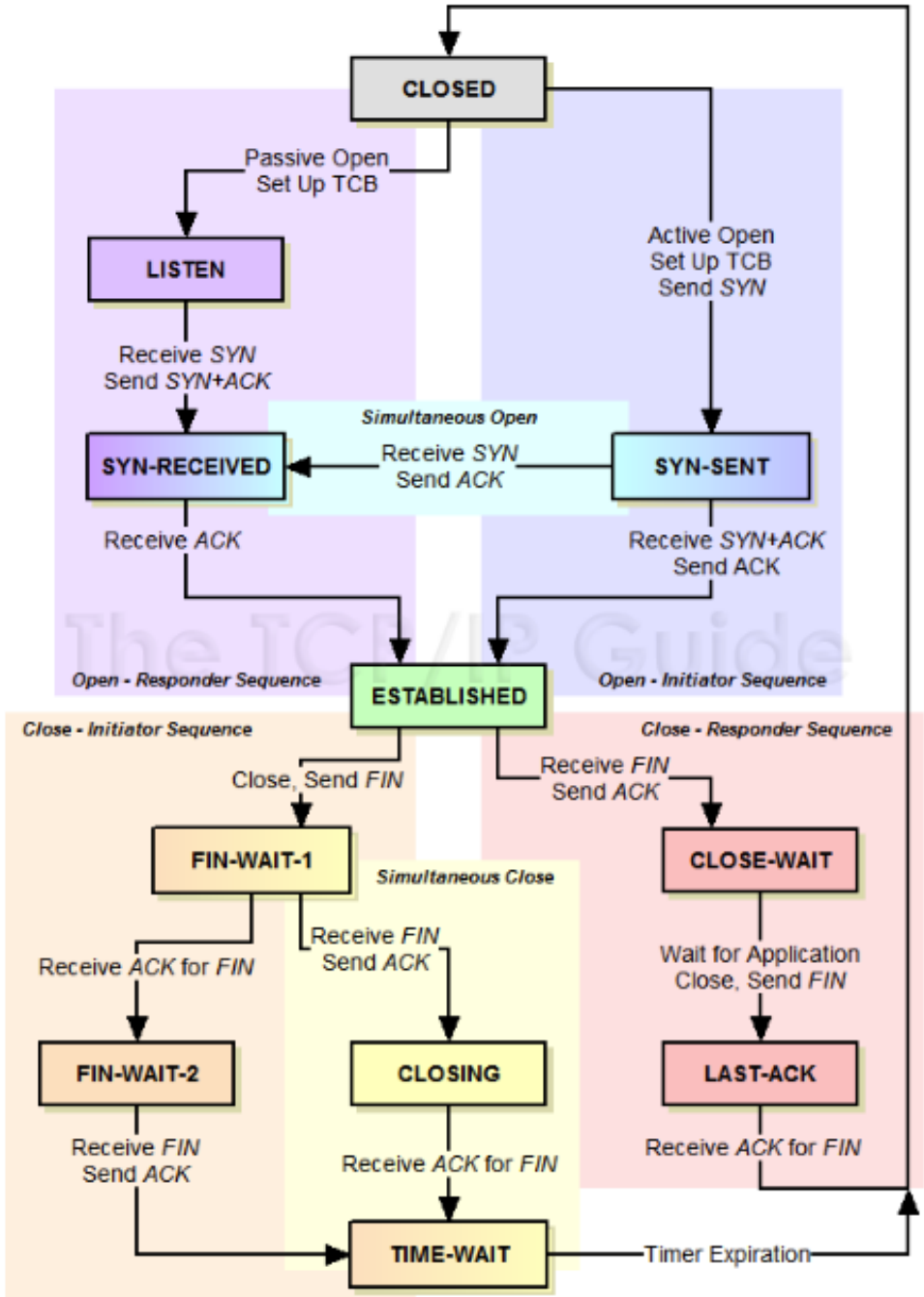
**CLOSE-WAIT**: in attesa di una richiesta di terminazione della connessione dall'utente locale;

**CLOSING**: in attesa di un ack alla chiusura dal peer remoto;

**LAST-ACK**: in attesa di un ack alla richiesta di disconnessione inviata al peer remoto.

**TIME-WAIT**: tempo di attesa per sincerarsi che il peer remoto abbia ricevuto l'ack alla sua richiesta di disconnessione;

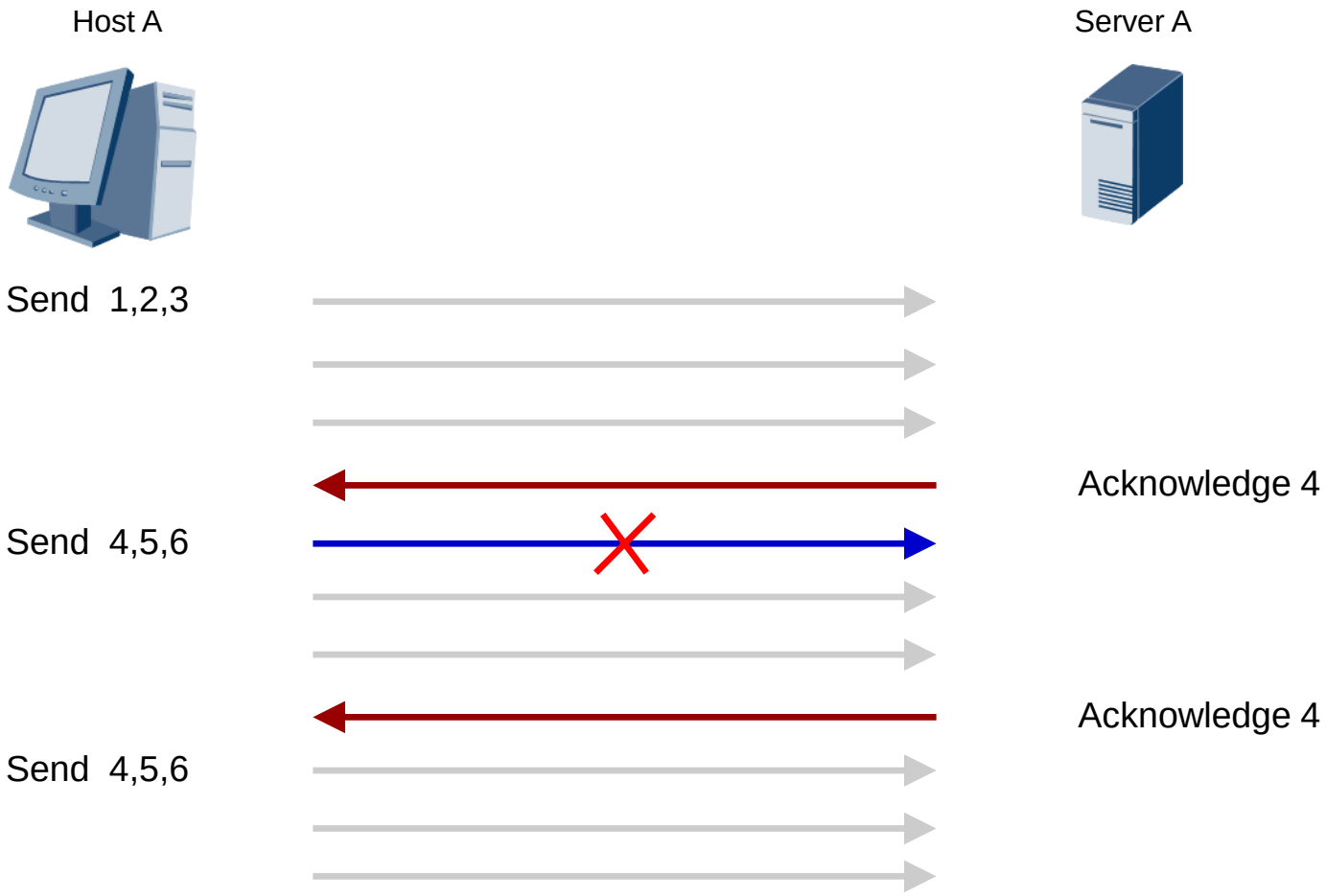
**CLOSED**: nessuno stato di connessione.



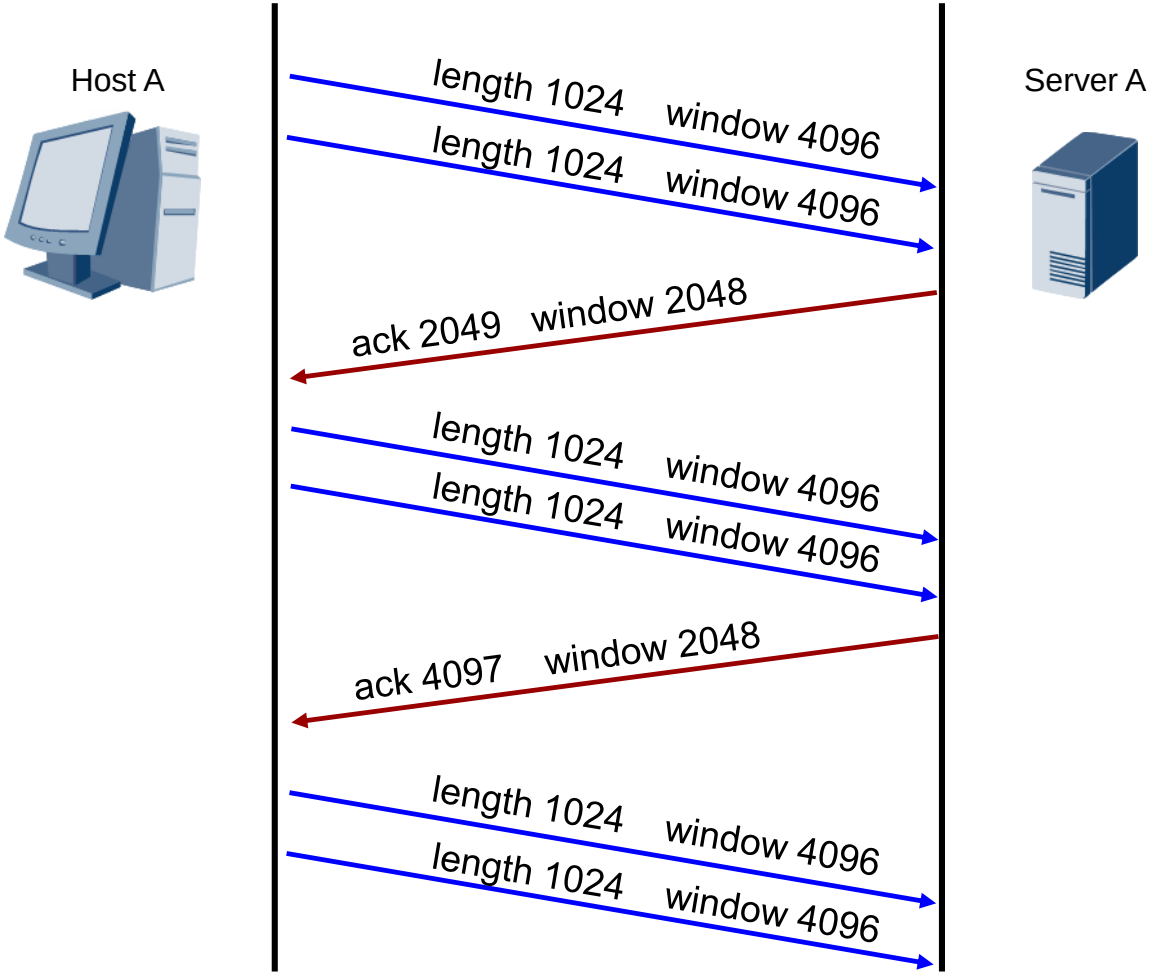
# TCP Connection Establishment

```
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:5900            0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:5901          0.0.0.0:*               LISTEN
tcp        0      0 192.168.188.11:5550     0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:5902          0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:5903          0.0.0.0:*               LISTEN
tcp        0      0 192.168.122.1:53        0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:25            0.0.0.0:*               LISTEN
tcp        0     64 192.168.188.11:22       192.168.188.253:61557   ESTABLISHED
tcp        0      0 192.168.188.11:37204    216.58.205.74:443       ESTABLISHED
```

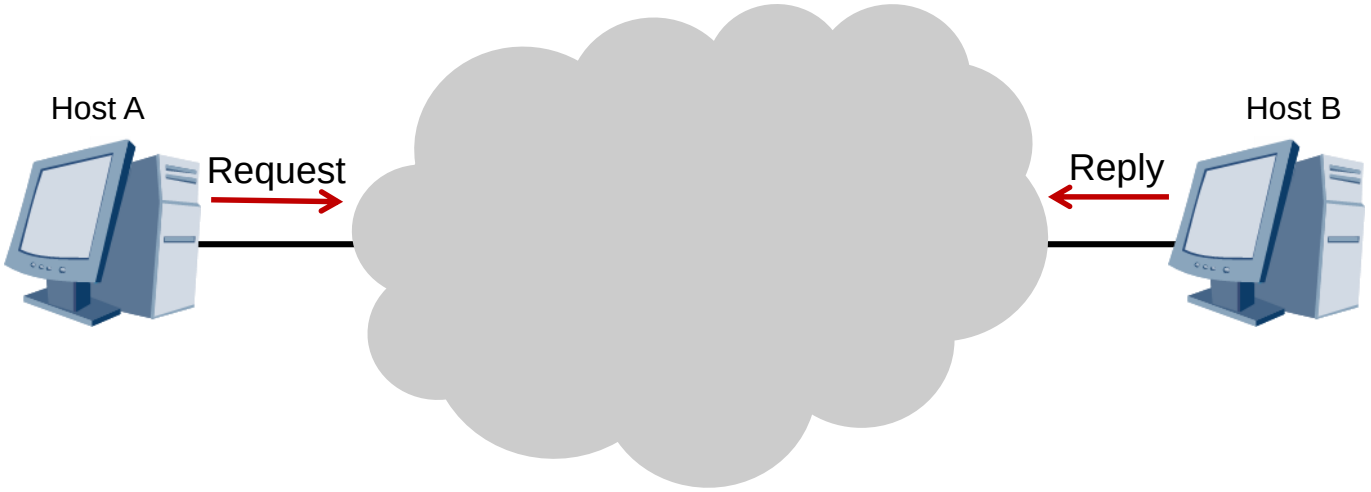
# TCP Transmission Process



# TCP Flow Control

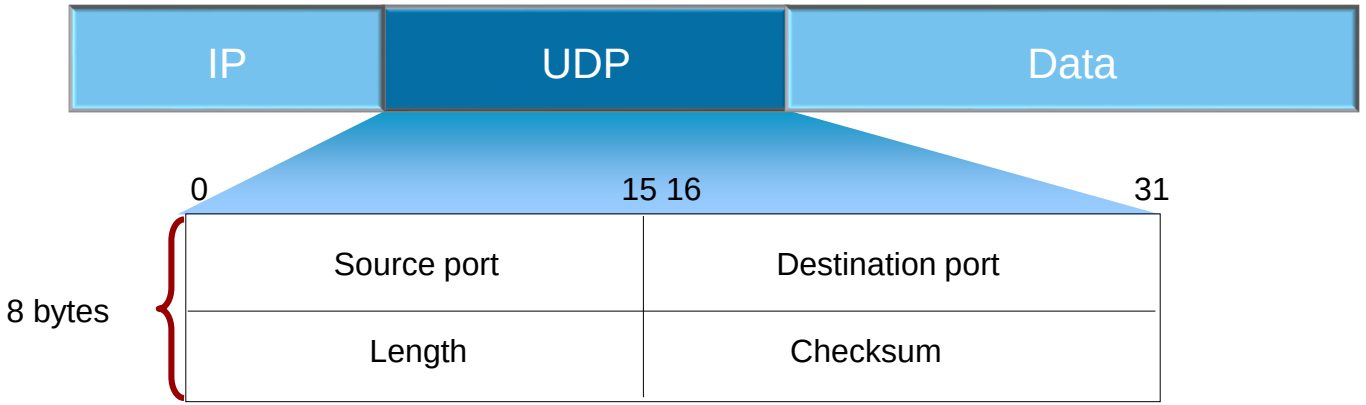


# User Datagram Protocol



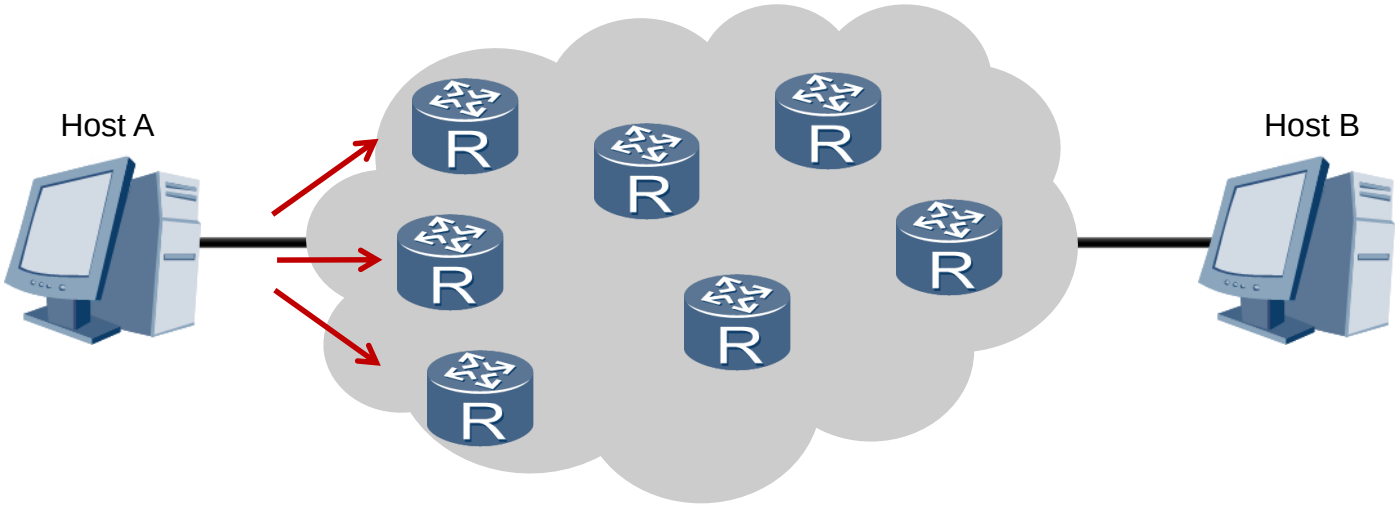

- UDP based data is sent without establishing a connection.

# UDP Datagram Format



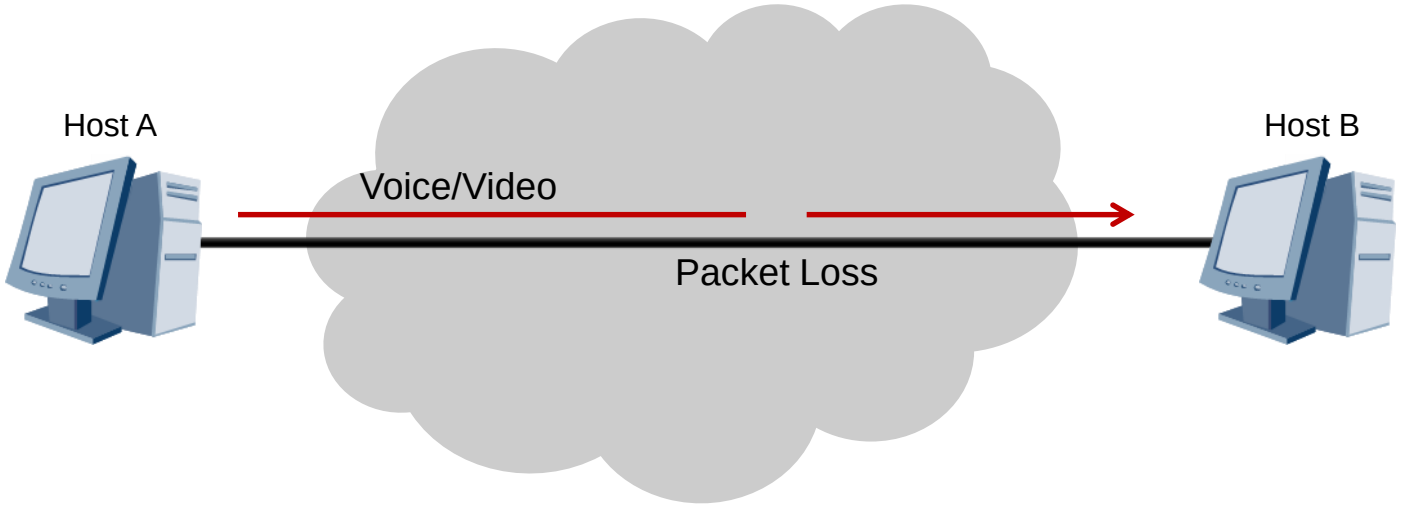

- UDP achieves minimal overhead for each datagram.
- Datagram delivery is not guaranteed with UDP.

# UDP Forwarding Behavior




- UDP susceptible to the possibility of datagram duplication or non-orderly delivery of datagrams.

# UDP Forwarding Behavior



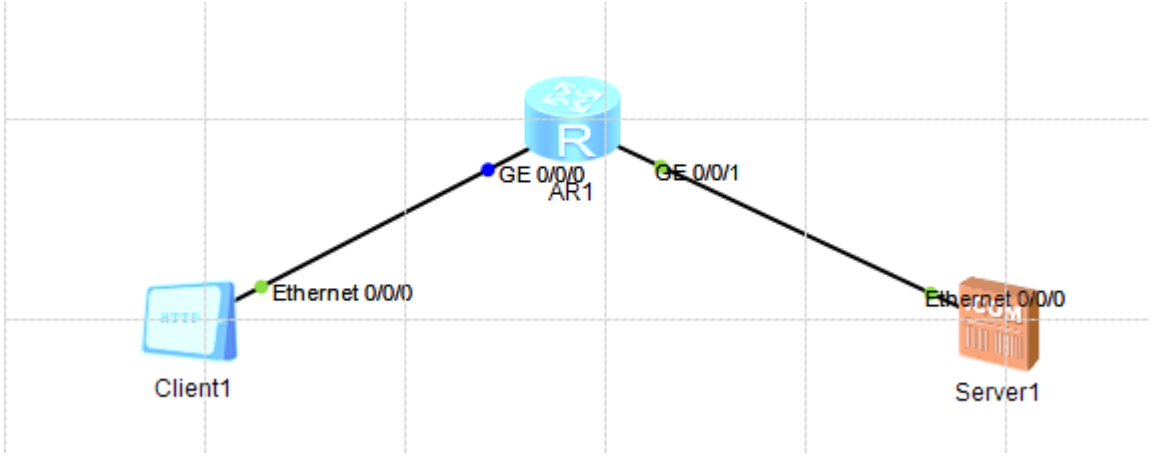

- There are no acknowledgements, therefore lost packets are not retransmitted, this however is beneficial to delay sensitive data.

# TCP vs UDP

| Funzione | Descrizione | UDP o TCP |
|---|---|---|
| Multiplexing tramite port | Sulla base del numero della porta è possibile scegliere la corretta applicazione alla quale sono destinati i dati. | TCP e UDP |
| Error recovery (reliability) | Consiste nel numerare e confermare i dati con l'uso dei campi sequence ed acknowledgement nell'header | TCP |
| Flow control | La quantità di dati che viene scambiata tra mittente e destinatario è negoziata dinamicamente per evitare congestioni sulla rete | TCP |
| Ordered data transfer and data segmentation | I dati provenienti dai layers superiori sono segmentati ed inviati al destinatario. Questi li riceve e li passa ai sui layer superiori nello stesso ordine in cui sono stati spediti. | TCP |
| Connection establishment and termination | Processo utilizzato per inizializzare i campi sequence ed acknowledgment. | TCP |

# Esempio – web!

Mini-Lab_basic:05-transport_01




| | | | |
|---|---|---|---|
| 192.168.1.1 | 192.168.2.1 | TCP | 58 2052 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 |
| 192.168.2.1 | 192.168.1.1 | TCP | 58 80 → 2052 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 |
| 192.168.1.1 | 192.168.2.1 | TCP | 54 2052 → 80 [ACK] Seq=1 Ack=1 Win=8192 Len=0 |
| 192.168.1.1 | 192.168.2.1 | HTTP | 211 GET / HTTP/1.1 Continuation |
| 192.168.2.1 | 192.168.1.1 | HTTP | 361 HTTP/1.1 200 OK (text/html) |
| 192.168.1.1 | 192.168.2.1 | TCP | 54 2052 → 80 [ACK] Seq=158 Ack=308 Win=7885 Len=0 |
| 192.168.1.1 | 192.168.2.1 | TCP | 54 2052 → 80 [FIN, ACK] Seq=158 Ack=308 Win=7885 Len=0 |
| 192.168.2.1 | 192.168.1.1 | TCP | 54 80 → 2052 [ACK] Seq=308 Ack=159 Win=8034 Len=0 |
| 192.168.2.1 | 192.168.1.1 | TCP | 54 80 → 2052 [FIN, ACK] Seq=308 Ack=159 Win=8034 Len=0 |
| 192.168.1.1 | 192.168.2.1 | TCP | 54 2052 → 80 [ACK] Seq=159 Ack=309 Win=7884 Len=0 |

# Esempio – web!

```
192.168.1.1     192.168.2.1     TCP      58 2052 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460
192.168.2.1     192.168.1.1     TCP      58 80 → 2052 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460
192.168.1.1     192.168.2.1     TCP      54 2052 → 80 [ACK] Seq=1 Ack=1 Win=8192 Len=0
192.168.1.1     192.168.2.1     HTTP     211 GET / HTTP/1.1 Continuation
192.168.2.1     192.168.1.1     HTTP     361 HTTP/1.1 200 OK  (text/html)
192.168.1.1     192.168.2.1     TCP      54 2052 → 80 [ACK] Seq=158 Ack=308 Win=7885 Len=0
192.168.1.1     192.168.2.1     TCP      54 2052 → 80 [FIN, ACK] Seq=158 Ack=308 Win=7885 Len=0
192.168.2.1     192.168.1.1     TCP      54 80 → 2052 [ACK] Seq=308 Ack=159 Win=8034 Len=0
192.168.2.1     192.168.1.1     TCP      54 80 → 2052 [FIN, ACK] Seq=308 Ack=159 Win=8034 Len=0
192.168.1.1     192.168.2.1     TCP      54 2052 → 80 [ACK] Seq=159 Ack=309 Win=7884 Len=0
```

```
Transmission Control Protocol, Src Port: 2052, Dst Port: 80, Seq: 0, Len: 0
    Source Port: 2052
    Destination Port: 80
    [Stream index: 3]
    [TCP Segment Len: 0]
    Sequence number: 0    (relative sequence number)
    Acknowledgment number: 0
    Header Length: 24 bytes
  > Flags: 0x002 (SYN)
    Window size value: 8192
    [Calculated window size: 8192]
    Checksum: 0xd1ea [unverified]
    [Checksum Status: Unverified]
    Urgent pointer: 0
  > Options: (4 bytes), Maximum segment size
```

# Esempio – web!

```
192.168.1.1     192.168.2.1     TCP      58 2052 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460
192.168.2.1     192.168.1.1     TCP      58 80 → 2052 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460
192.168.1.1     192.168.2.1     TCP      54 2052 → 80 [ACK] Seq=1 Ack=1 Win=8192 Len=0
192.168.1.1     192.168.2.1     HTTP    211 GET / HTTP/1.1 Continuation
192.168.2.1     192.168.1.1     HTTP    361 HTTP/1.1 200 OK  (text/html)
192.168.1.1     192.168.2.1     TCP      54 2052 → 80 [ACK] Seq=158 Ack=308 Win=7885 Len=0
192.168.1.1     192.168.2.1     TCP      54 2052 → 80 [FIN, ACK] Seq=158 Ack=308 Win=7885 Len=0
192.168.2.1     192.168.1.1     TCP      54 80 → 2052 [ACK] Seq=308 Ack=159 Win=8034 Len=0
192.168.2.1     192.168.1.1     TCP      54 80 → 2052 [FIN, ACK] Seq=308 Ack=159 Win=8034 Len=0
192.168.1.1     192.168.2.1     TCP      54 2052 → 80 [ACK] Seq=159 Ack=309 Win=7884 Len=0
```

```
> Internet Protocol Version 4, Src: 192.168.2.1, Dst: 192.168.1.1
v Transmission Control Protocol, Src Port: 80, Dst Port: 2052, Seq: 0, Ack: 1, Len: 0
    Source Port: 80
    Destination Port: 2052
    [Stream index: 3]
    [TCP Segment Len: 0]
    Sequence number: 0    (relative sequence number)
    Acknowledgment number: 1    (relative ack number)
    Header Length: 24 bytes
  > Flags: 0x012 (SYN, ACK)
    Window size value: 8192
    [Calculated window size: 8192]
    Checksum: 0xb42c [unverified]
    [Checksum Status: Unverified]
    Urgent pointer: 0
  > Options: (4 bytes), Maximum segment size
  > [SEQ/ACK analysis]
```





HUAWEI

# Esempio – web!

```
192.168.1.1     192.168.2.1     TCP      58 2052 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460
192.168.2.1     192.168.1.1     TCP      58 80 → 2052 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460
192.168.1.1     192.168.2.1     TCP      54 2052 → 80 [ACK] Seq=1 Ack=1 Win=8192 Len=0
192.168.1.1     192.168.2.1     HTTP     211 GET / HTTP/1.1 Continuation
192.168.2.1     192.168.1.1     HTTP     361 HTTP/1.1 200 OK  (text/html)
192.168.1.1     192.168.2.1     TCP      54 2052 → 80 [ACK] Seq=158 Ack=308 Win=7885 Len=0
192.168.1.1     192.168.2.1     TCP      54 2052 → 80 [FIN, ACK] Seq=158 Ack=308 Win=7885 Len=0
192.168.2.1     192.168.1.1     TCP      54 80 → 2052 [ACK] Seq=308 Ack=159 Win=8034 Len=0
192.168.2.1     192.168.1.1     TCP      54 80 → 2052 [FIN, ACK] Seq=308 Ack=159 Win=8034 Len=0
192.168.1.1     192.168.2.1     TCP      54 2052 → 80 [ACK] Seq=159 Ack=309 Win=7884 Len=0
```

```
> Internet Protocol Version 4, Src: 192.168.1.1, Dst: 192.168.2.1
v Transmission Control Protocol, Src Port: 2052, Dst Port: 80, Seq: 1, Ack: 1, Len: 0
    Source Port: 2052
    Destination Port: 80
    [Stream index: 3]
    [TCP Segment Len: 0]
    Sequence number: 1    (relative sequence number)
    Acknowledgment number: 1    (relative ack number)
    Header Length: 20 bytes
  > Flags: 0x010 (ACK)
    Window size value: 8192
    [Calculated window size: 8192]
    [Window size scaling factor: -2 (no window scaling used)]
    Checksum: 0xcbe9 [unverified]
    [Checksum Status: Unverified]
    Urgent pointer: 0
  > [SEQ/ACK analysis]
```

# Esempio – web!

```
192.168.1.1     192.168.2.1     TCP      58 2052 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460
192.168.2.1     192.168.1.1     TCP      58 80 → 2052 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460
192.168.1.1     192.168.2.1     TCP      54 2052 → 80 [ACK] Seq=1 Ack=1 Win=8192 Len=0
192.168.1.1     192.168.2.1     HTTP     211 GET / HTTP/1.1 Continuation
192.168.2.1     192.168.1.1     HTTP     361 HTTP/1.1 200 OK  (text/html)
192.168.1.1     192.168.2.1     TCP      54 2052 → 80 [ACK] Seq=158 Ack=308 Win=7885 Len=0
192.168.1.1     192.168.2.1     TCP      54 2052 → 80 [FIN, ACK] Seq=158 Ack=308 Win=7885 Len=0
192.168.2.1     192.168.1.1     TCP      54 80 → 2052 [ACK] Seq=308 Ack=159 Win=8034 Len=0
192.168.2.1     192.168.1.1     TCP      54 80 → 2052 [FIN, ACK] Seq=308 Ack=159 Win=8034 Len=0
192.168.1.1     192.168.2.1     TCP      54 2052 → 80 [ACK] Seq=159 Ack=309 Win=7884 Len=0
```

```
> Internet Protocol Version 4, Src: 192.168.1.1, Dst: 192.168.2.1
v Transmission Control Protocol, Src Port: 2052, Dst Port: 80, Seq: 1, Ack: 1, Len: 0
    Source Port: 2052
    Destination Port: 80
    [Stream index: 3]
    [TCP Segment Len: 0]
    Sequence number: 1    (relative sequence number)
    Acknowledgment number: 1    (relative ack number)
    Header Length: 20 bytes
  > Flags: 0x010 (ACK)
    Window size value: 8192
    [Calculated window size: 8192]
    [Window size scaling factor: -2 (no window scaling used)]
    Checksum: 0xcbe9 [unverified]
    [Checksum Status: Unverified]
    Urgent pointer: 0
  > [SEQ/ACK analysis]
```

# Esempio – web!

| | | | |
|---|---|---|---|
| 192.168.1.1 | 192.168.2.1 | TCP | 58 2052 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 |
| 192.168.2.1 | 192.168.1.1 | TCP | 58 80 → 2052 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 |
| 192.168.1.1 | 192.168.2.1 | TCP | 54 2052 → 80 [ACK] Seq=1 Ack=1 Win=8192 Len=0 |
| 192.168.1.1 | 192.168.2.1 | HTTP | 211 GET / HTTP/1.1 Continuation |
| 192.168.2.1 | 192.168.1.1 | HTTP | 361 HTTP/1.1 200 OK (text/html) |
| 192.168.1.1 | 192.168.2.1 | TCP | 54 2052 → 80 [ACK] Seq=158 Ack=308 Win=7885 Len=0 |
| 192.168.1.1 | 192.168.2.1 | TCP | 54 2052 → 80 [FIN, ACK] Seq=158 Ack=308 Win=7885 Len=0 |
| 192.168.2.1 | 192.168.1.1 | TCP | 54 80 → 2052 [ACK] Seq=308 Ack=159 Win=8034 Len=0 |
| 192.168.2.1 | 192.168.1.1 | TCP | 54 80 → 2052 [FIN, ACK] Seq=308 Ack=159 Win=8034 Len=0 |
| 192.168.1.1 | 192.168.2.1 | TCP | 54 2052 → 80 [ACK] Seq=159 Ack=309 Win=7884 Len=0 |

```
> Frame 31: 211 bytes on wire (1688 bits), 211 bytes captured (1688 bits) on interface 0
> Ethernet II, Src: HuaweiTe_16:43:b6 (54:89:98:16:43:b6), Dst: HuaweiTe_a5:35:48 (00:e0:fc:a5:35:48)
> Internet Protocol Version 4, Src: 192.168.1.1, Dst: 192.168.2.1
> Transmission Control Protocol, Src Port: 2052, Dst Port: 80, Seq: 1, Ack: 1, Len: 157
> Hypertext Transfer Protocol
> Hypertext Transfer Protocol
```

# Esempio – servizi udp!

```
> Frame 4993: 110 bytes on wire (880 bits), 110 bytes captured (880 bits) on interface 0
> Ethernet II, Src: Xensourc_09:50:8c (00:16:3e:09:50:8c), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
> Internet Protocol Version 4, Src: 193.205.130.233, Dst: 193.205.130.255
v User Datagram Protocol, Src Port: 137, Dst Port: 137
    Source Port: 137
    Destination Port: 137
    Length: 76
    Checksum: 0x65ff [unverified]
    [Checksum Status: Unverified]
    [Stream index: 366]
> NetBIOS Name Service
```

```
> Frame 8468: 198 bytes on wire (1584 bits), 198 bytes captured (1584 bits) on interface 0
> Ethernet II, Src: FujitsuT_cf:7a:7f (00:19:99:cf:7a:7f), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
> Internet Protocol Version 4, Src: 192.168.81.138, Dst: 255.255.255.255
v User Datagram Protocol, Src Port: 17500, Dst Port: 17500
    Source Port: 17500
    Destination Port: 17500
    Length: 164
    Checksum: 0x7f97 [unverified]
    [Checksum Status: Unverified]
    [Stream index: 56]
> Dropbox LAN sync Discovery Protocol
```

# Summary

- What is the purpose of the acknowledgement field in the TCP header?
- Which TCP code bits are involved in a TCP three-way handshake?

# Thank you

www.huawei.com