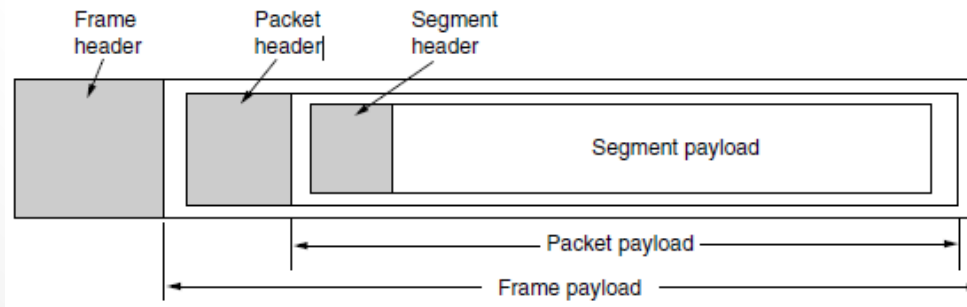


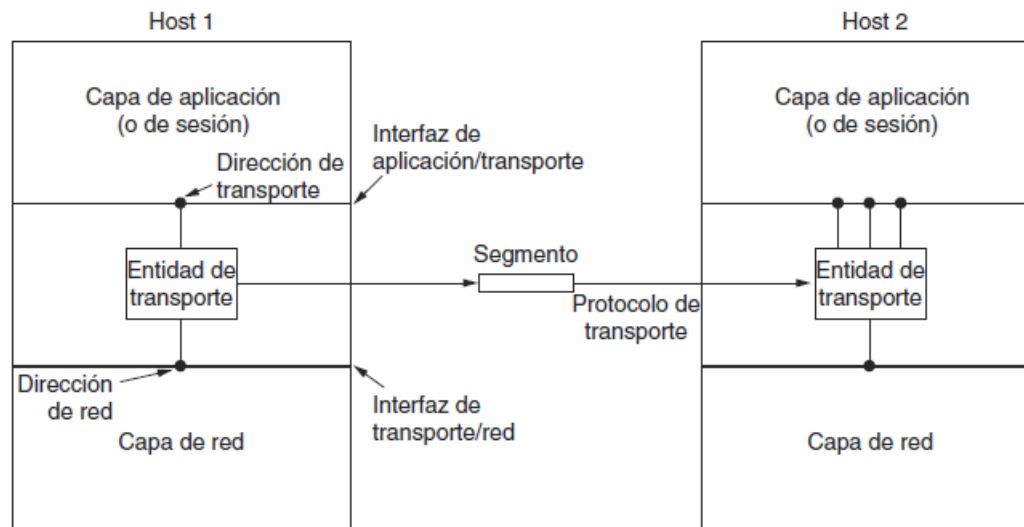
# Transporte

- La capa de red provee la entrega de paquetes punto a punto a través del uso de datagramas o circuitos virtuales.
- La capa de transporte se basa en la capa de red para proveer transporte de datos desde un proceso en una máquina de origen a un proceso en la máquina de destino.
- Ofrece las abstracciones que necesitan las aplicaciones para utilizar la red.



# Transporte

- El objetivo principal de la capa de transporte es proporcionar un servicio de transmisión de datos eficiente, confiable y económico a sus usuarios: procesos que normalmente son de la capa de aplicación.



# Servicios de Transporte

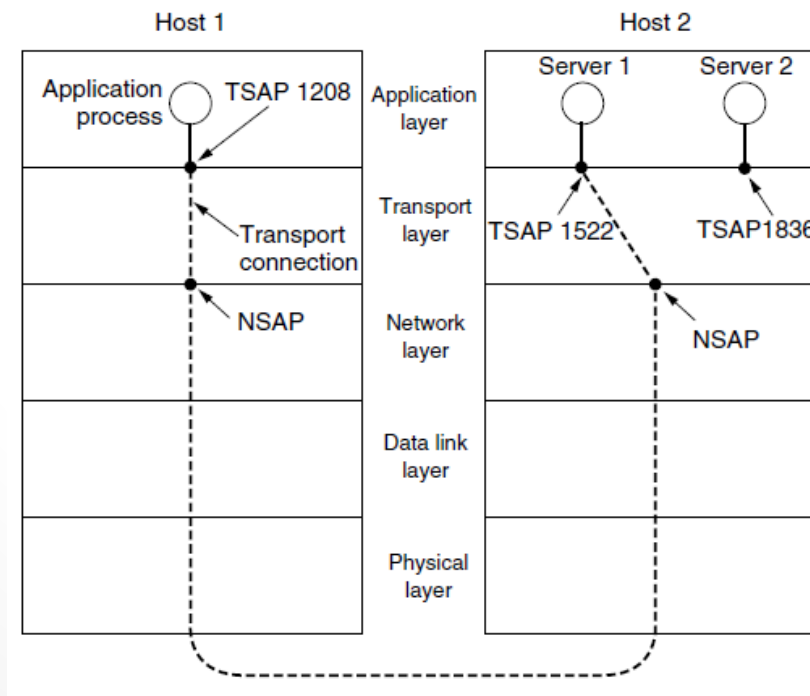
- Existen 2 tipos de servicios de transporte, los orientados a conexión y los sin conexión.
- Los orientados a conexión son similares a los servicios de red orientados a conexión:
  - Fases: establecimiento, transferencia de datos y liberación.
  - Direccionamiento y control de flujo similares.
- Los servicio de transporte sin conexión también son similares a los de la capa de red.

# Servicios de Transporte

- Si hay tanta similaridad, ¿Por qué hay 2 capas diferentes? La justificación es sutil e importante:
- El código de transporte se ejecuta por completo en las máquinas de los usuarios, y la capa de red se ejecuta en su mayor parte en los routers, en un contexto amplio, fuera del control de los usuarios y operados por las empresas portadoras.
- Es una forma de permitir que ante inconvenientes en la capa de red, se tengan herramientas para compensarlos desde la capa de transporte.

# Dirección de Transporte

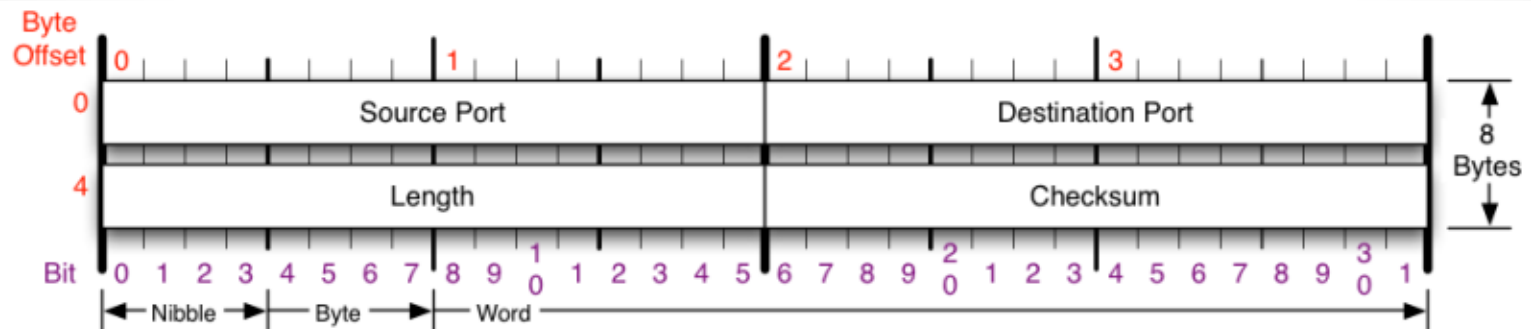
- La dirección de transporte distingue entre distintos procesos dentro de un mismo host.
- Nombre genérico: Transport Service Access Point.
- Puertos o ports.



# UDP

- [User Datagram Protocol](#), protocolo 17, RFC 768.
- Protocolo mínimo, simple, sin overhead:
  - Unreliable
  - Connectionless (stateless)
  - Message oriented
- Datagrams, sin garantía de llegada a destino, orden o duplicados.
- Ports: enteros de 16 bits.
- Soporta multicast.

# UDP Protocol Header



Checksum

Checksum of entire UDP segment and pseudo header (parts of IP header)

RFC 768

Please refer to RFC 768 for the complete User Datagram Protocol (UDP) Specification.

# UDP

- Usos:
  - Implementación sencilla
  - DNS, request-reply
  - RIP
  - SNMP, Simple Network Management
  - DHCP, Dynamic Host Configuration Protocol
  - VoIP
  - Video streaming
  - QUIC



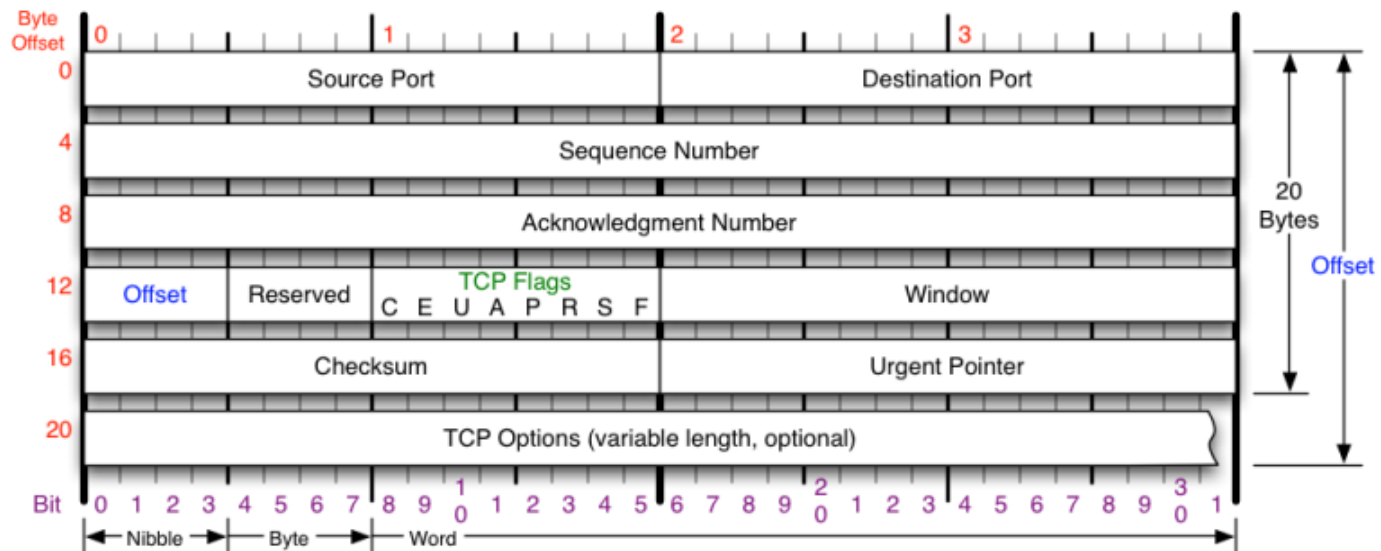
# TCP

- [Transmission Control Protocol](#), protocolo 6, RFC 793.
- Características:
  - Reliable
  - Connection oriented
  - Ordered byte stream
- Ports: enteros de 16 bits.
- [Control de congestión](#)

# TCP

- Números de secuencia de segmentos.
- Acknowledge de números de secuencia recibidos.
- Retransmisiones:
  - Buffers (window)
  - Timers
- Checksum de payload.
- Acknowledge de números de secuencia recibidos.
- Inicio y fin de conexión.

# TCP Protocol Header



## TCP Flags

C E U A P R S F

Congestion Window

C 0x80 Reduced (CWR)

E 0x40 ECN Echo (ECE)

U 0x20 Urgent

A 0x10 Ack

P 0x08 Push

R 0x04 Reset

S 0x02 Syn

F 0x01 Fin

## Congestion Notification

ECN (Explicit Congestion Notification). See RFC 3168 for full details, valid states below.

Packet State	DSB	ECN bits
Syn	0 0	1 1
Syn-Ack	0 0	0 1
Ack	0 1	0 0
No Congestion	0 1	0 0
No Congestion	1 0	0 0
Congestion	1 1	0 0
Receiver Response	1 1	0 1
Sender Response	1 1	1 1

## TCP Options

- 0 End of Options List
- 1 No Operation (NOP, Pad)
- 2 Maximum segment size
- 3 Window Scale
- 4 Selective ACK ok
- 8 Timestamp

## Checksum

Checksum of entire TCP segment and pseudo header (parts of IP header)

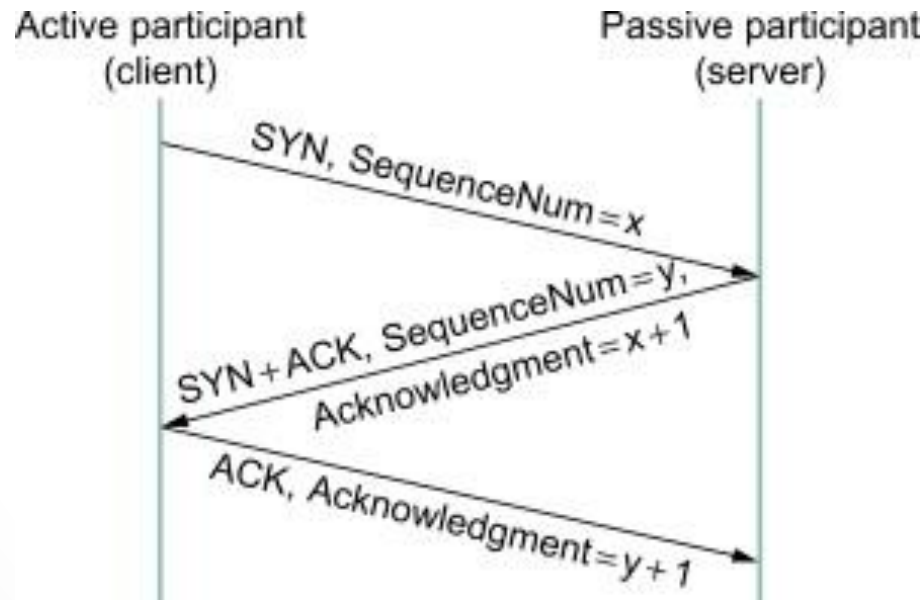
## Offset

Number of 32-bit words in TCP header, minimum value of 5. Multiply by 4 to get byte count.

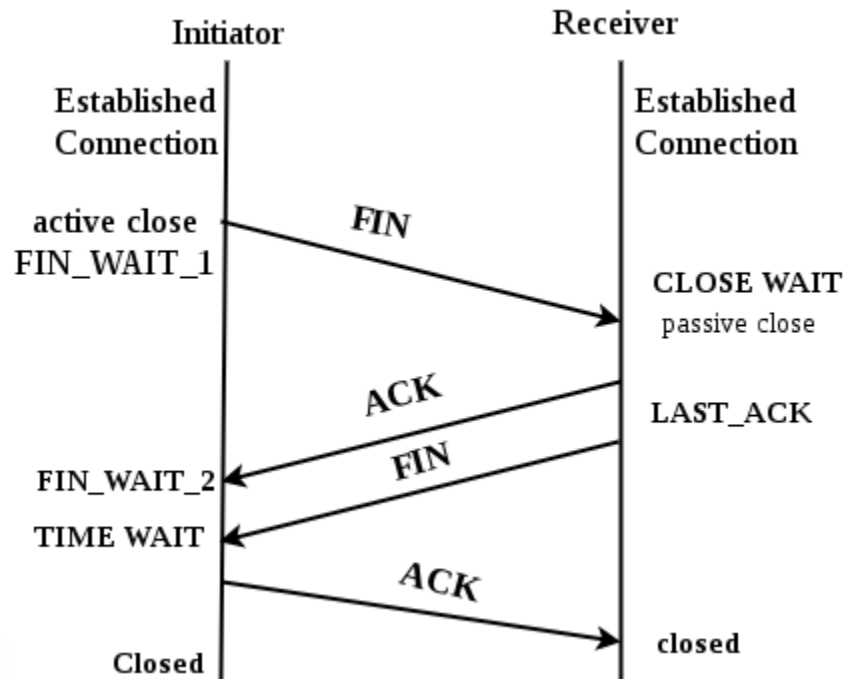
## RFC 793

Please refer to RFC 793 for the complete Transmission Control Protocol (TCP) Specification.

# Connection Start Tree Way Handshake



# Connection Close Handshake



# TCP

## Simplified State Diagram

