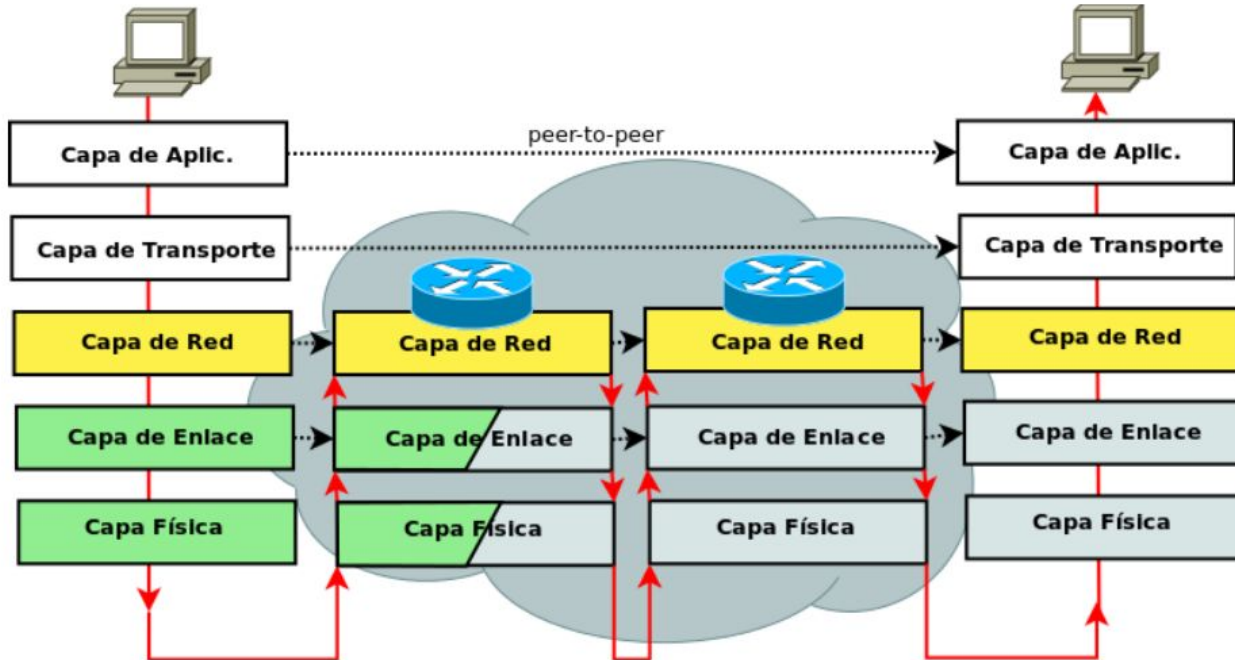


# Capa de transporte

# Introducción

La capa de transporte provee **comunicación lógica** entre procesos.



# Introducción

- Los protocolos de capa de transporte son mecanismos **para comunicar procesos**.
- Uso de **sockets**
- Multiplexación/demultiplexación

# Sockets

Un **socket** es una abstracción a través de la cual una aplicación puede enviar o recibir información.

- Para poder **enviar** información: un proceso debe escribir datos en un socket
- Para poder **recibir** información: un proceso debe leer datos desde un socket

Cada sistema operativo tiene su implementación.

# Sockets

Ya que puede haber más de un socket al mismo tiempo en un host, es necesario poder identificarlos de manera unívoca. Para lograr esto se usa la combinación de los datos:

- Protocolo (UDP o TCP)
- Número de puerto
- Dirección IP (capa de red)

# Sockets - Multiplexación/Demultiplexación

- Multiplexación: la capa de transporte del emisor arma segmentos enviando en las cabeceras los identificadores para el origen y destino.
- Demultiplexación: la capa de transporte del receptor revisa las cabeceras y de esta manera determina a cual socket debe entregarle los datos que contiene el segmento.

# Capa de transporte: Segmento TCP

Puerto origen					Puerto destino				
Número de secuencia									
Número de reconocimiento									
4 bits header length	reserved	urg	ack	psh	rst	syn	fin	tamaño de ventana	
TCP checksum					Puntero a datos urgentes				
Opciones									
Datos									

# Capa de transporte: Segmento (o datagrama) UDP

Puerto origen	Puerto destino
Tamaño	UDP checksum
Datos	



# Comando **ncat**

## **Servidor:**

TCP: `$ ncat -k -l 127.0.0.1 3000`

UDP: `$ ncat -u -l 127.0.0.1 3000`

- `ss -a | grep 3000`

# Comando **ncat**

## **Cliente:**

TCP: \$ ncat 127.0.0.1 3000

UDP: \$ ncat -u 127.0.0.1 3000