


Capa Transporte





Campos - función



Mecanismos de entrega confiable



Control de errores



Control de Flujo



Control de congestión

# Puertos, su objetivo



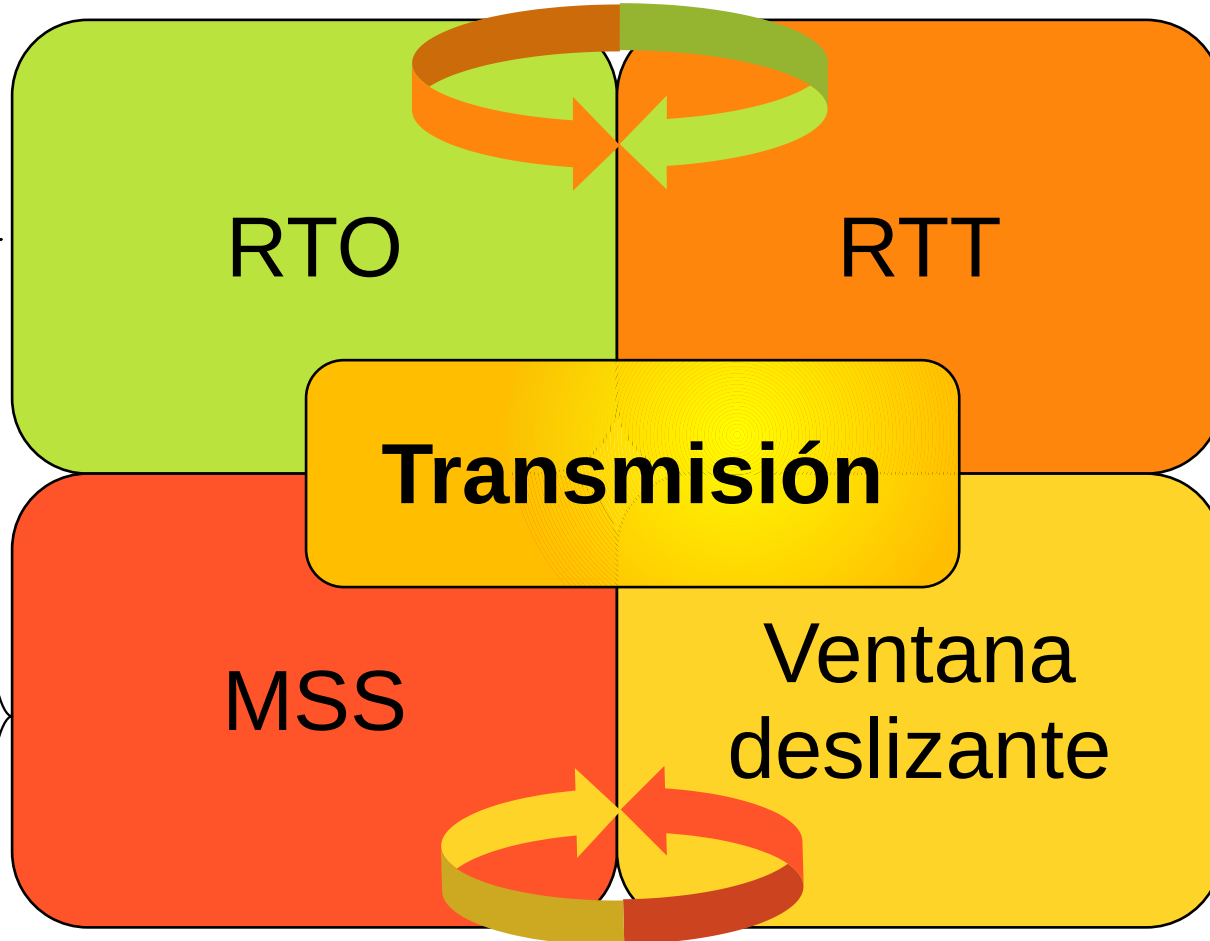
- ¿Por qué se necesita un número de puerto para identificar un servicio si ya tenemos la dirección IP del servidor?
- Cuando ingreso a un sitio web, ¿Debo especificar siempre el número de puerto en el navegador?
  - ¿Por qué si? ¿Por qué no?
- Si recibe un segmento y no existe un proceso en el puerto destino
  - ¿Qué pasa en TCP?
  - ¿Y en en UDP?

# Conceptos en la transmisión



¿Por qué el cálculo del tiempo de Espera de Retransmisión se basa en un RTT promedio entre muy corto y muy largo?

¿Cuáles son las implicancias de tener un **RTO** demasiado corto o demasiado largo?



¿Cómo se relacionan los timestamps en el encabezado TCP con la estimación del **RTT** y su ajuste dinámico?

¿Cuáles son los **campos** que utiliza?

¿Cuándo y cómo se lleva a cabo la **negociación del MSS** entre el cliente y el servidor?

¿Por qué es importante acordar un **MSS** adecuado para la transmisión de datos eficiente?

¿Cuál es la función de la **ventana deslizante** en TCP y cómo afecta la eficiencia de la transmisión de datos?

¿Cómo se relaciona el tamaño de la ventana deslizante con el MSS y la cantidad de datos que el emisor puede enviar antes de recibir un ACK?

# Transferencia Fiable en TCP



Adaptación de envío por la capacidad del **receptor**

**Ventana de dinámica por conexión**, independiente de cada extremo

Emisor envía sin esperar ACK según tamaño **ventana deslizante**, se desliza al recibir un ACK en orden

Ventana = 0

Envíos periódicos con Byte 0

Adaptación de envío por la congestión de la **red** percibida

**Ventana de dinámica por conexión**, independiente de cada extremo

Cálculo de la velocidad de transmisión a medida que recibe ACKs

3 Acks duplicados y vencimiento de RTO

Uso de algoritmos al detectar una congestión:  
slow start congestion avoidance

## Control de Flujo

## Control de congestión

- Capacidad de envío
  - Mínimo entre la congestión, el flujo y los errores detectados

# Algoritmo de control de congestión con 3 componentes:



Funcionamiento:

• Slow start

• Congestion avoidance

• Fast retransmit

• TCP Tahoe  
• TCP Reno

(obligatorio) **arranque lento**

- **inicio:** tamaño bajo y va aumentando
- Crece **exponencialmente**
- Ante una congestión:
  - pone en 1 el MSS e inicia nuevamente
  - Guarda en una variable umbral
  - Sigüientes transmisiones sin pasar el umbral
  - Llega al umbral pasa al estado congestion avoidance

(obligatorio) **evitar congestión**  
**(Congestion avoidance)**

- con el valor del umbral calcula la vel.de transmisión que se incrementa **linealmente**

(opcional) **recuperación rápida:** Intenta recuperarse más rápido que slow start

¿Lo usan?

¿qué cambio tienen en la eficiencia?

# TCP vs UDP



Multiplexación/Demultiplexación

Detección de errores

Control de errores

Multiplexación/Demultiplexación

Detección de errores

Control de errores

Orientado a conexión

Control de flujo

Control de congestión

Orientado a conexión

Control de flujo

Control de congestión