

Universidad Industrial de  
Santander  
Escuela de Sistemas

# Redes de Computadoras Capa de Transporte

Prof. Gilberto Díaz  
[gilberto.diaz@uis.edu.co](mailto:gilberto.diaz@uis.edu.co)

# Protocolo TCP

TCP utiliza un conjunto de algoritmos que envían paquetes a la red sin ningún tipo de reserva previa

Sin embargo, estos algoritmos son capaces de reaccionar ante determinados eventos que se producen en la red.

Entre ellos se destacan:

El Control de Congestión  
Recuperación de Segmentos Perdidos.

# Protocolo TCP: Control de Congestión

Los algoritmos para el control de congestión son:

*Slow Start / Congestion Avoidance*

*Fast Retransmit / Fast Recovery*

# Protocolo TCP: Control de Congestión

Para implantar estos algoritmos se agregan las siguientes variables:

*RWND* (*Receiver Window*) - Receptor

*CWND* (*Congestion Window*) - Emisor

La ventana más pequeña de estas dos gobierna el flujo transmitido.

# Protocolo TCP: Control de Congestión

Para implantar estos algoritmos se agregan las siguientes variables:

***SSTHRESH*** (*Slow Start Threshold*) Se utiliza para determinar que algoritmo se usa: *slow start* o *congestion avoidance*

# Protocolo TCP: Control de Congestión RFC 2581

Al comienzo de la transmisión TCP lentamente determina la capacidad de la red (utiliza *slow start*) y evitar el envío de una cantidad inapropiada de datos

La ventana inicial (*Initial Window, IW*) debe ser menor o igual que  $2 * SSMS$  (*Sender Maximum Segment Size*)

# Protocolo TCP: Control de Congestión RFC 2581

*Slow start* también se utiliza luego de que una pérdida es detectada

Durante *slow start* TCP incrementa la ventana *CWND* en al menos *SMSS* bytes por cada ACK recibido que reconoce nuevos datos

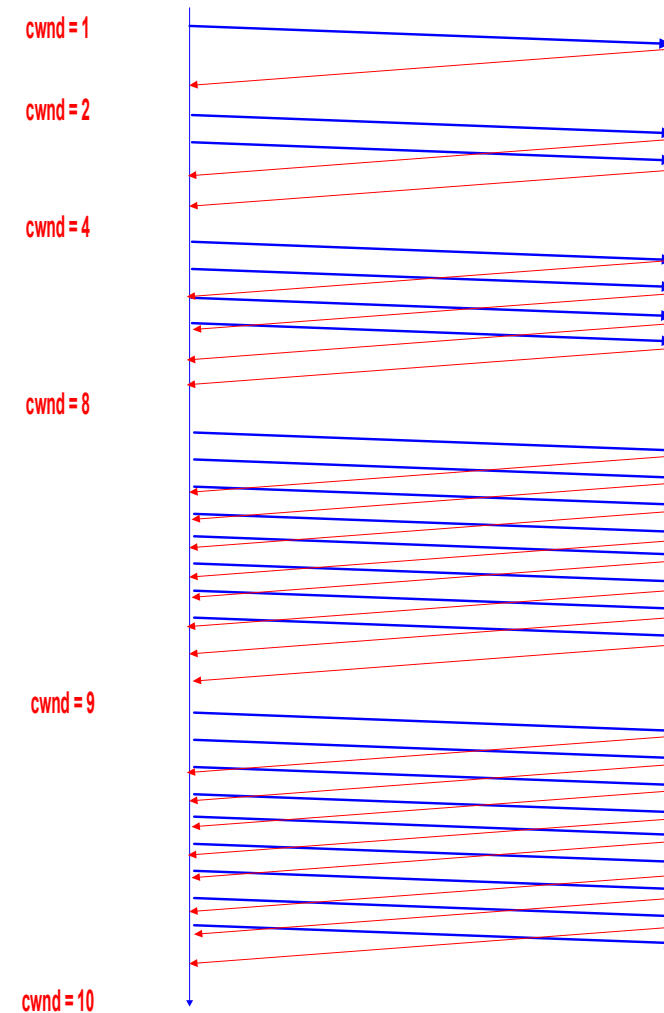
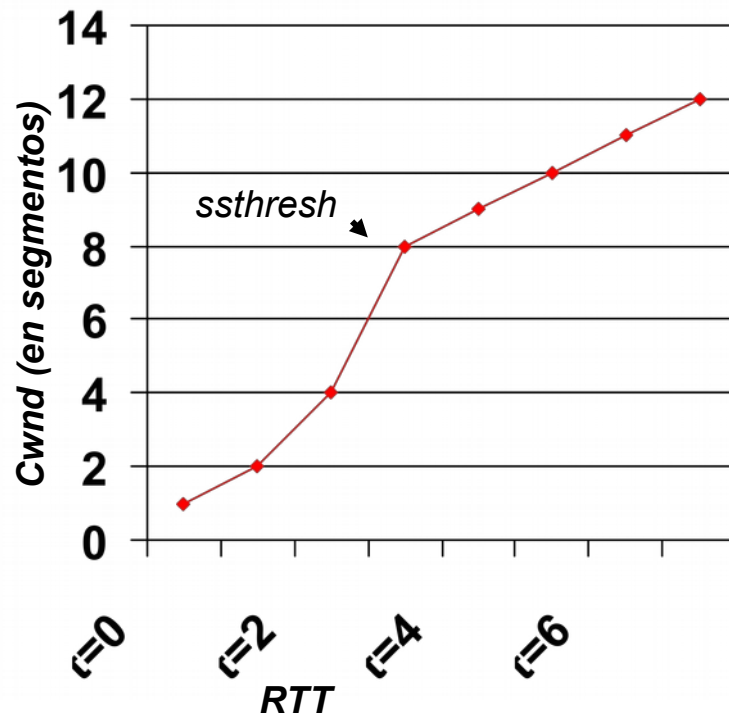
*slow start* finaliza cuando *CWND* excede a *SSTRESH*

# Protocolo TCP: Control de Congestión RFC 2581

## Slow Start + Congestion Avoidance

Asumimos :

*ss\_thresh = 8*





# Protocolo TCP: Control de Congestión RFC 2581

## Detección de congestión

Si tenemos *packet loss*, el ACK para ese paquete no se recibirá

El paquete eventualmente puede demorarse  
No ACK se interpreta como señal de congestión

# Protocolo TCP: Control de Congestión RFC 2581

## Detección de congestión

*Timeout* significa congestión

Cada vez que se detecta congestión

$$SSTRESH = CWND/2$$

$$CWND = 1$$

Y se entra en *slow start* nuevamente



# Protocolo TCP: Control de Congestión RFC 2581

