Control de Congestión en TCP

Teoría de la Comunicaciones

22 de Octubre de 2025

Referencias

- **Computer Networks - A Systems Approach** Peterson, Davie
- **RFC 5681 - TCP Congestion Control**
- **Computer Networks**: Tanenbaum, Wetherall
- **TCP/IP Illustrated, Volume 1: The Protocols** Stevens, Fall

RFC 5681: TCP Congestion Control

Abstract

This document defines TCP's four intertwined congestion control algorithms: slow start, congestion avoidance, fast retransmit, and fast recovery. In addition, the document specifies how TCP should begin transmission after a relatively long idle period, as well as discussing various acknowledgment generation methods. This document obsoletes RFC 2581.

Definiciones

- **Sender Maximum Segment Size** (SMSS): Máximo tamaï¿½o que puede tener los segmentos que arme
- **Receiver Maximum Segment Size** (RMSS): Máximo tamaï¿½o que puede recibir el receptor
- **Full-Sized Segment:** Un segmento con la cantidad máxima (SMSS) de bytes

Definiciones

- **Receiver Window** (RWND): Última Advertised Window recibida.
- **Congestion Window** (CWND): Variable que limita la cantidad de datos que puede enviar el emisor
- **Initial Window** (IW): Valor de CWND después del handshake.
- **Loss Window** (LW): Valor de CWND después de un timeout.
- **Restart Window** (RW): Valor de CWND después de un período idle.

Definiciones

- **Flight Size:** (LastByteSent - LastByteACKed)

- **Slow Start Threshold** (SSTHRESH): Umbral que define si se usa Slow Start o Congestion Avoidance.

- **Duplicate Acknowledgement:** Un ACK es duplicado, si:
  1. El receptor del ACK tiene datos sin confirmar.
  2. El ACK no tiene datos.
  3. No estï¿½n los flags SYN ni FIN
  4. El número de ACK es igual al máximo ACK recibido.
  5. La *advertised window* es igual a la última recibida.

La ventana de congestión

CWND

"Es una estimación de la cantidad de información que puedo meter en la red sin que se vea afectada su performance."

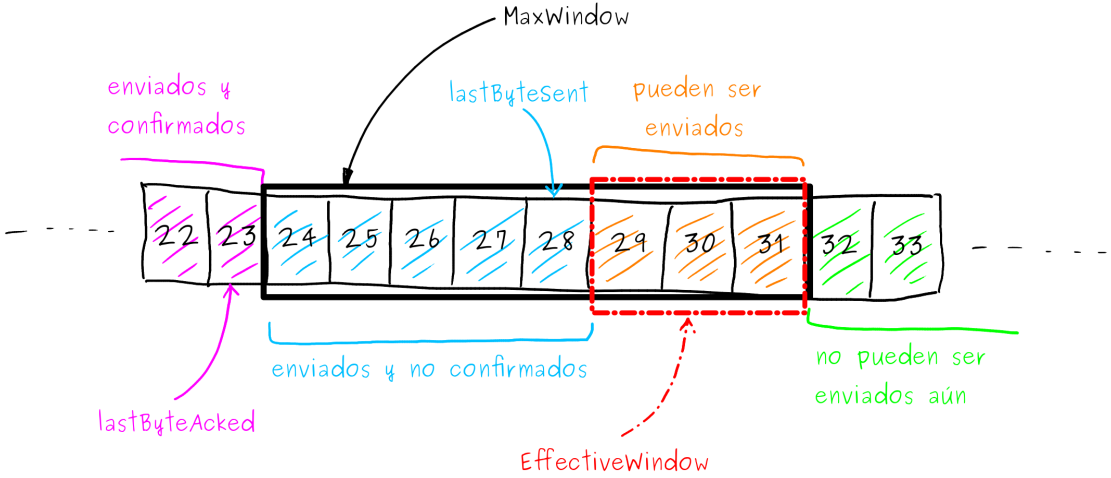
MaxWindow = Min(RWND, CWND)

"Limitada por red o por host"

EffectiveWindow = MaxWindow - (LastByteSent - LastByteAcked)

"Cuántos bytes puedo despachar."

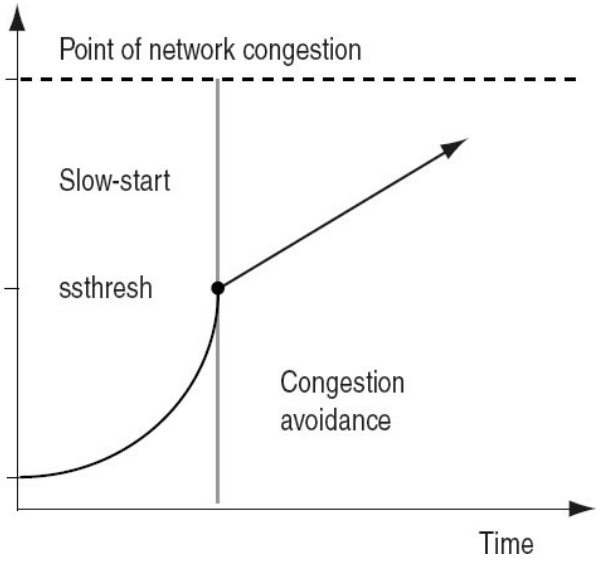
Ventana del emisor



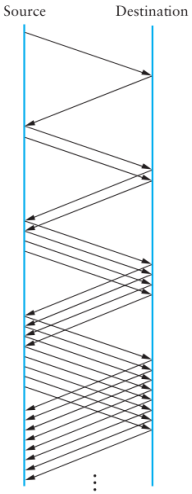
Algoritmos de Control de Congestión

- **Slow Start:** Comenzar enviando pocos datos
- **Congestion Avoidance:** Aumentar un SMSS por RTT
- **Fast Retransmit / Fast Recovery:** No esperar al time out, para recuperarse de un error.

Algoritmos: Slow Start + Congestion Avoidance



Algoritmos: Slow Start



Inicialmente:

- ★ $CWND = IW = 2 * SMSS$
- ★ $SSTHRESH = alta(max\ advertised\ window)$

si $CWND < SSTHRESH$:

Hacer $CWND+ = min(N, SMSS)$ **por cada ACK**

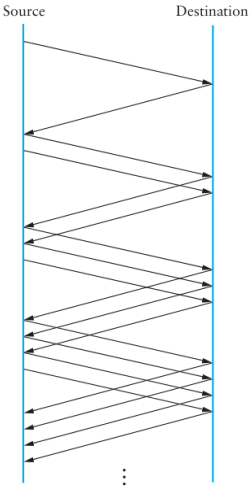
N es la cantidad de bytes reconocidos por el ACK

"Se usa la llegada de ACKs como retroalimentación positiva"

Ejercicio

Considere el efecto de usar Slow Start en una conexión TCP recién establecida ($IW = 2 * SMSS$, $SSTHRESH = 64KB$, $SMSS = 2KB$), que tiene un RTT de 10 mseg y sin congestión ni errores presentes en la red. Si la RWND es de 24 KB, ¿Cuánto tiempo transcurre antes de que pueda ser enviada la primera ventana de recepción llena?

Algoritmos: Congestion Avoidance



★ Aumentar la *CWND* en un *SMSS* por *RTT*

si $CWND > SSTHRESH$:

Hacer $CWND+ = SMSS * SMSS/CWND$
por cada ACK

Esta es la parte **Additive Increase**

Time out

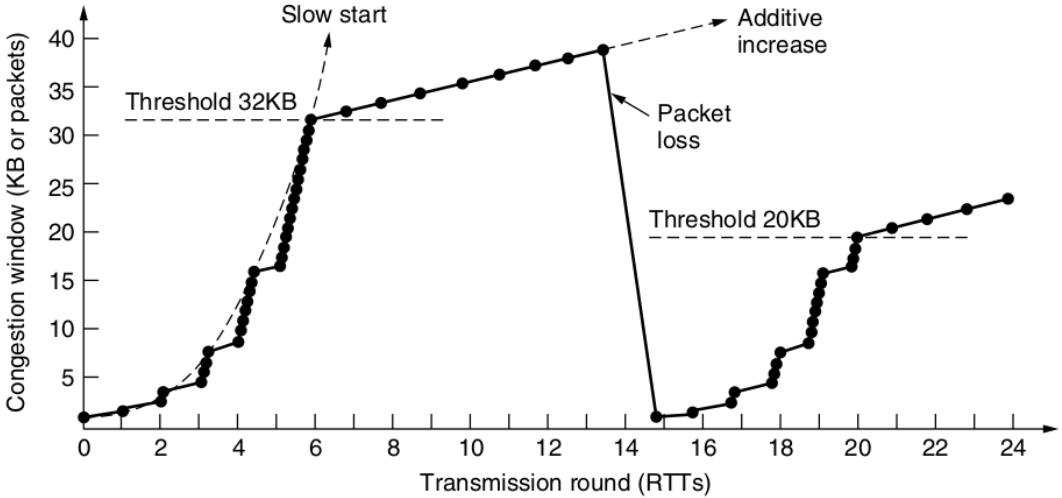
Ante un *time-out* se cambian los valores a:

- ★ $CWND = LW(1SMSS)$
- ★ $SSTHRESH = max(FlightSize/2, 2 * SMSS)$

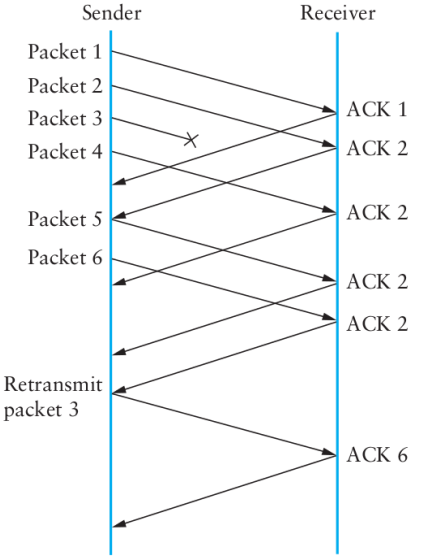
$\Rightarrow$ **Se comienza de nuevo con Slow Start**

“Se usa el time-out como retroalimentación negativa”

TCP Tahoe


Slow start
Additive increase
Threshold 32KB
Packet loss
Threshold 20KB
Congestion window (KB or packets)
Transmission round (RTTs)
0
2
4
6
8
10
12
14
16
18
20
22
24
5
10
15
20
25
30
35
40

Algoritmos: Fast Recovery / Fast Retransmit



Al 3er ACK duplicado, el emisor deberï¿½a retransmitir el segmento perdido.

- ★ SSTHRESH = max(FlightSize/2, 2 * SMSS)
- ★ CWND = SSTHRESH + 3 * SMSS

mientras no se reconozcan nuevo datos:
Hacer CWND += SMSS
por cada ACK Duplicado

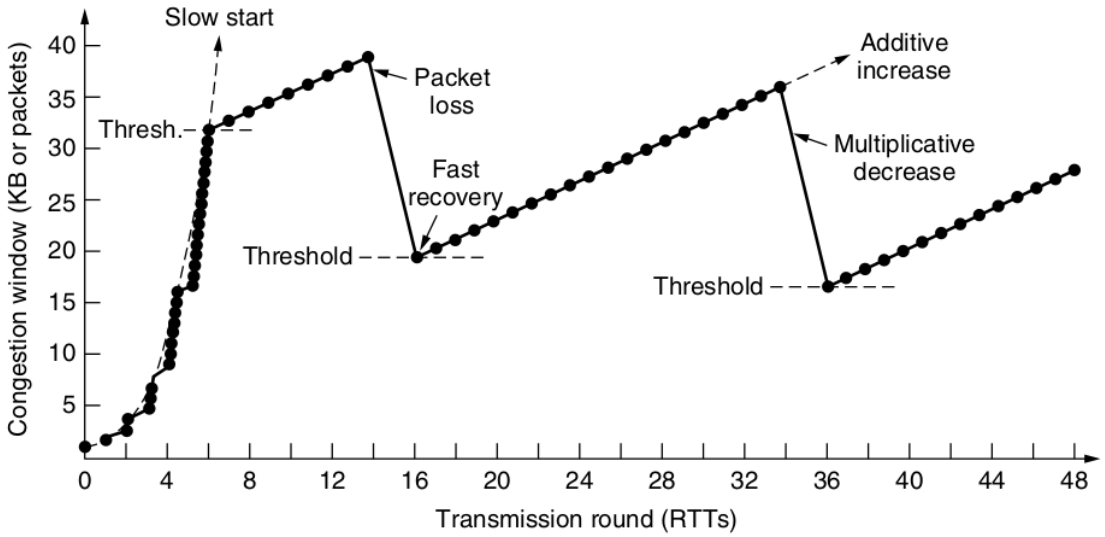
Termina cuando llega el primer ACK que reconoce nuevos datos.

- ★ CWND = SSTHRESH

⇒ **Se continúa con Congestion Avoidance**

"Se usa el 3er ACK duplicado como retroalimentación negativa"

TCP Reno


Slow start
Packet loss
Additive increase
Thresh.
Fast recovery
Multiplicative decrease
Threshold
Threshold
Congestion window (KB or packets)
Transmission round (RTTs)
0
4
8
12
16
20
24
28
32
36
40
44
48
5
10
15
20
25
30
35
40

Consideraciones adicionales

Reiniciando conexiones idle.

"Si una conexión no envía datos no tiene retroalimentaciones."

Si no hay actividad por mas de un RTO

★ CWND = RW = min(IW,cwnd)

Generando reconocimientos.

"Se puede esperar antes de enviar un ACK:"

★ A lo sumo 500ms o 2*SMSS bytes sin reconocer

Ejercicio

En una conexión recién establecida con RTT=200ms, el host receptor siempre anuncia una *AdvertisedWindow* de 16KB. La red está cargada al punto que si una ráfaga fuera de 16KB o mas, se perderían todos los segmentos de la misma.

a. ¿Cuánto vale la CWND luego de enviar un archivo de 40KB?
b. 3 segundos después del envió del archivo, se envía otro archivo de 30KB ¿Cuánto tiempo tarda?

Ejercicio

Dada una conexión TCP recién establecida entre dos host para la cual el RTT es de 50ms. Los dos host están separados por un sólo router que también conecta otras redes y está cargado a tal punto que cada vez que una ráfaga de paquetes es de 20KB o más, se descartan todos los paquetes de la ráfaga. El host emisor tiene que enviar un archivo bastante grande que se está transmitiendo por horas y el host receptor siempre anuncia una *AdvertisedWindow* de 28KB.

a. Si se define que una conexión alcanza el *estado estacionario* en el momento que el SSTHRESH converge a un valor a partir del cual ya no cambia más. ¿Cuánto tiempo tarda la conexión en alcanzar el estado estacionario? ¿Cuál es el valor del SSTHRESH es dicho momento?

- Finalizada la transferencia, se cierra la conexión, y se inicia una nueva en la que el host receptor siempre anuncia una *AdvertisedWindow* de 18KB. Si esta nueva conexión tuviera que transferir el mismo archivo, ¿tardaría más o menos tiempo que la anterior? *(Suponer las mismas condiciones de congestión en el router)*