TEMA 3
CAPA DE TRANSPORTE EN INTERNET

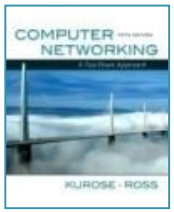
Fundamentos de Redes

2015/2016

- Bibliografía Básica:

Capítulo 10, Pedro García Teodoro, Jesús Díaz Verdejo y Juan Manuel López Soler. ***TRANSMISIÓN DE DATOS Y REDES DE COMPUTADORES***, Ed. Pearson, 2ª Ed. ,Pearson, 2014, ISBN: 978-0-273-76896-8

- Para saber más…



Capítulo 3 James F. Kurose y Keith W. Ross. ***COMPUTER NETWORKING. A TOP-DOWN APPROACH***, 5ª Edición, Addison-Wesley, 2010, ISBN: 9780136079675.

- Agradecimientos:

Transparencias originales de **Juan Manuel López Soler, Pedro García Teodoro, Jorge Navarro Ortiz**, Departamento TSTC, UGR.

1. **Introducción.**
2. Protocolo de datagrama de usuario (UDP).
3. Protocolo de control de transmisión (TCP).
    1. Multiplexación/demultiplexación.
    2. Control de conexión.
    3. Control de errores y de flujo.
    4. Control de congestión.
4. Extensiones TCP.
5. Ejercicios.

INTRODUCCIÓN

Funciones y servicios de la capa de transporte:

Comunicación extremo a extremo (*end-to-end)*.

Multiplexación/demultiplexación de aplicaciones → *puerto.*

Protocolo UDP:

Multiplexación/demultiplexación de aplicaciones**.**

Servicio no orientado a conexión, no fiable**.**

Protocolo TCP:

Multiplexación/demultiplexación de aplicaciones**.**

Servicio orientado a conexión, fiable: Control de errores y de flujo. Control de la conexión.

Control de congestión.

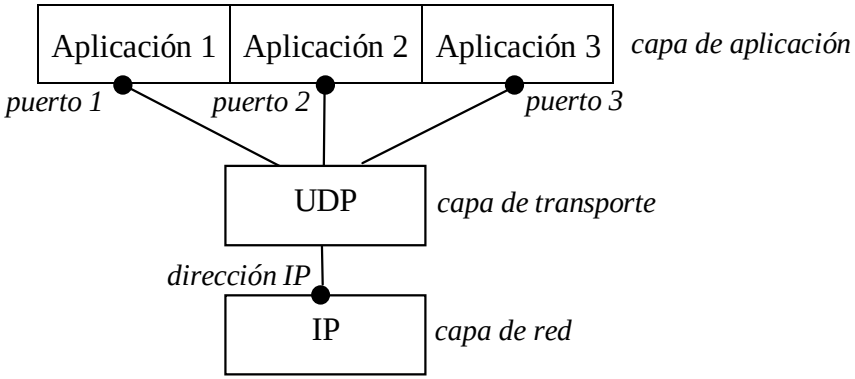
1. Introducción.
2. **Protocolo de datagrama de usuario (UDP).**
3. Protocolo de control de transmisión (TCP).
    1. Multiplexación/demultiplexación.
    2. Control de conexión.
    3. Control de errores y de flujo.
    4. Control de congestión.
4. Extensiones TCP.
5. Ejercicios.

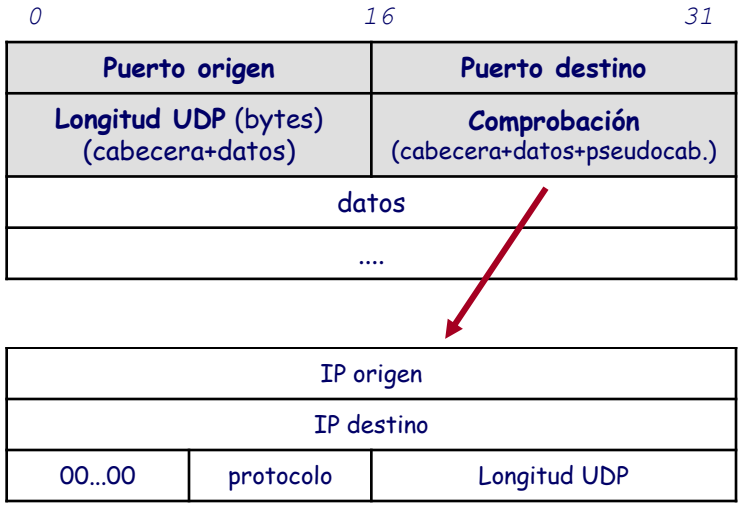
USER DATAGRAM PROTOCOL (UDP)

"User Datagram Protocol": RFC 768.

Funcionalidad "*best-effort*":

- Servicio no orientado a conexión
- Servicio no fiable
- No hay garantías de entrega ordenada.
- No hay control de congestión: entrega tan rápida como se pueda.
- Multiplexación/demultiplexación: transportar las TPDU al proceso correcto.





5. Calcule la suma de comprobación en UDP y TCP de las siguientes palabras de 8 bits (observe que aunque UDP y TCP utilicen palabras de 16 bits, en este ejercicio se pide el mismo cálculo sobre palabras de 8 bits): 01010011, 01010100, 01110100.

a) **¿Por qué UDP/TCP utilizan el complemento a uno de la suma complemento a uno, en lugar de directamente la suma en complemento a uno?**

b) **¿cómo detecta el receptor los errores?**

c) **¿se detectan todos los errores de 1 bit?**

d) **¿se detectan todos los errores que afectan simultáneamente a 2 bits?**

USER DATAGRAM PROTOCOL (UDP)

Multiplexación/demultiplexación: transportar las TPDU al proceso correcto.

Ejemplos de puertos UDP preasignados

Puerto	Aplicación/Servicio	Descripción
53	DNS	Servicio de nombres de domino
69	TFTP	Transferencia simple de ficheros
123	NTP	Protocolo de tiempo de red
161	SNMP	Protocolo simple de administración de red
520	RIP	Protocolo de información de encaminamiento

UDP se usa frecuentemente para aplicaciones multimedia: tolerantes a fallos y sensibles a retardos.

Cada segmento UDP se encapsula en un datagrama IP.

1. Introducción.
2. Protocolo de datagrama de usuario (UDP).
3. **Protocolo de control de transmisión (TCP).**
    1. Multiplexación/demultiplexación.
    2. Control de conexión.
    3. Control de errores y de flujo.
    4. Control de congestión.
4. Extensiones TCP.
5. Ejercicios.

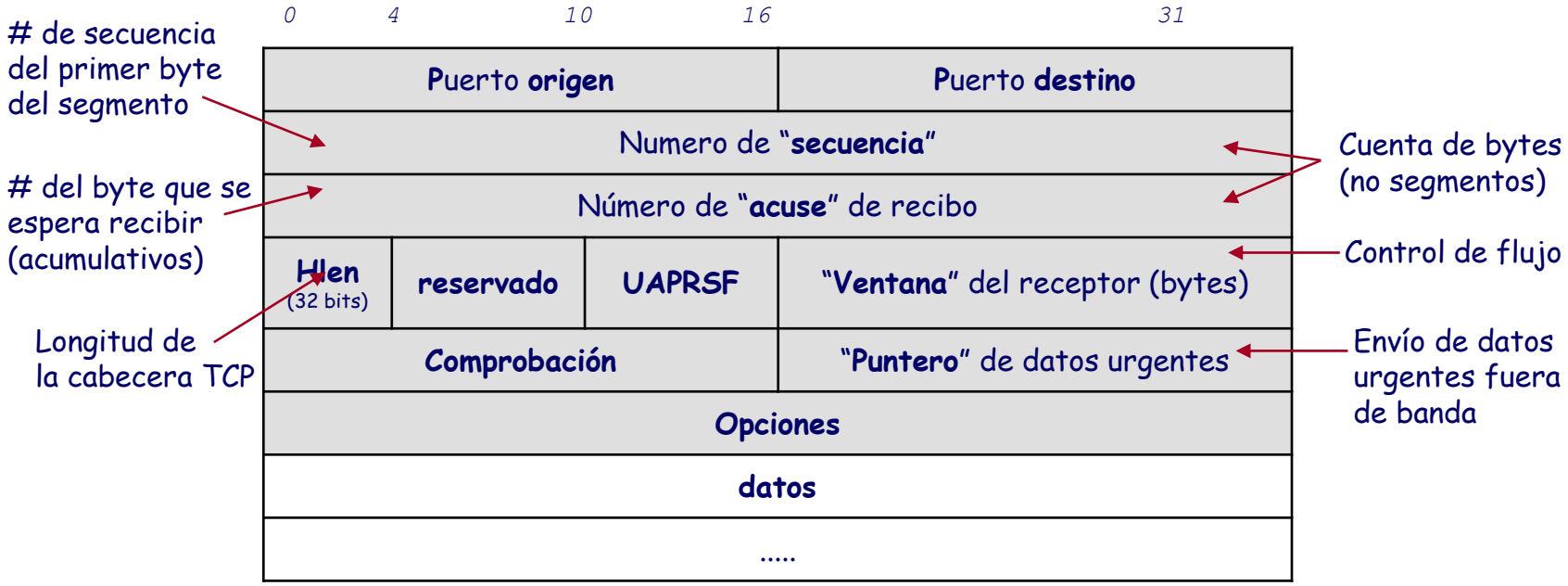
TRANSMISSION CONTROL PROTOCOL (TCP)

Características del "Transmission Control Protocol": RFC 793 (1122, 1323, 2018, 2581).

- Servicio orientado a conexión ("*hand-shaking*")
- Entrega ordenada
- *full-duplex*
- Mecanismo de control de flujo de detección y recuperación de errores (ARQ – Automatic Repeat reQuest)
  - confirmaciones positivas (ACKs) y acumulativas
  - Incorporación de confirmaciones ("piggybacking")
  - "timeouts" adaptables
  - Ventanas adaptables
- Mecanismo de control de congestión
- Servicio fiable → control de congestión y control de flujo

TRANSMISSION CONTROL PROTOCOL (TCP)

TPDU TCP = **Segmento TCP**:



Cada segmento TCP se encapsula en un datagrama IP.

TRANSMISSION CONTROL PROTOCOL (TCP)

Multiplexación/demultiplexación de aplicaciones:

Puerto	Aplicación/Servicio	Descripción
20	FTP-DATA	Transferencia de ficheros: datos
21	FTP	Transferencia de ficheros: control
22	SSH	Terminal Seguro
23	TELNET	Acceso remoto
25	SMTP	Correo electrónico
53	DNS	Servicio de nombres de domino
80	HTTP	Acceso hipertexto (web)
110	POP3	Descarga de correo

La "conexión TCP" se identifica por: puerto e IP origen y puerto e IP destino (y opcionalmente protocolo)

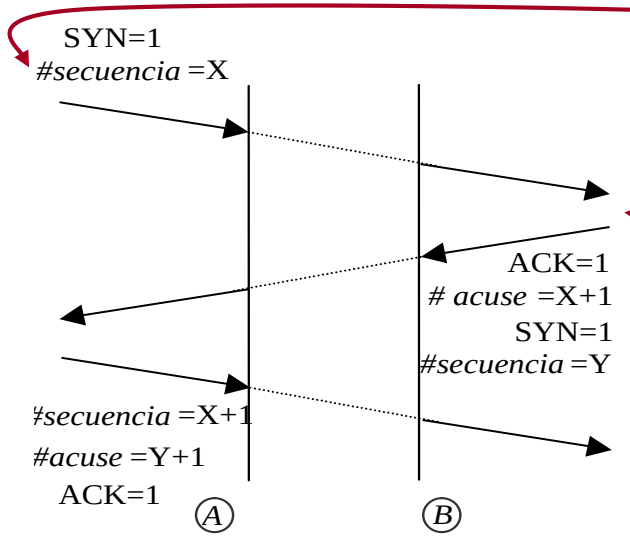
Control de la conexión:

El intercambio de información tiene tres fases: *three-way handshake.*

Establecimiento de la conexión (sincronizar # de secuencia y reservar recursos).

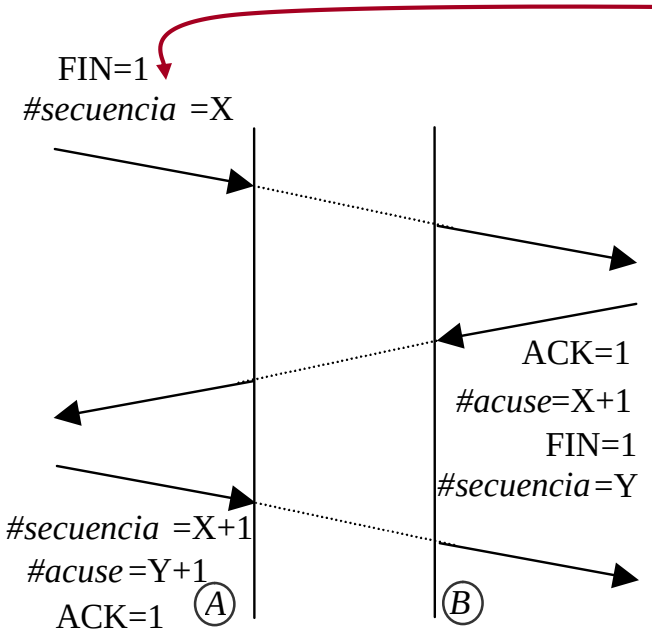
Intercambio de datos (full-duplex).

Cierre de la conexión (liberar recursos).

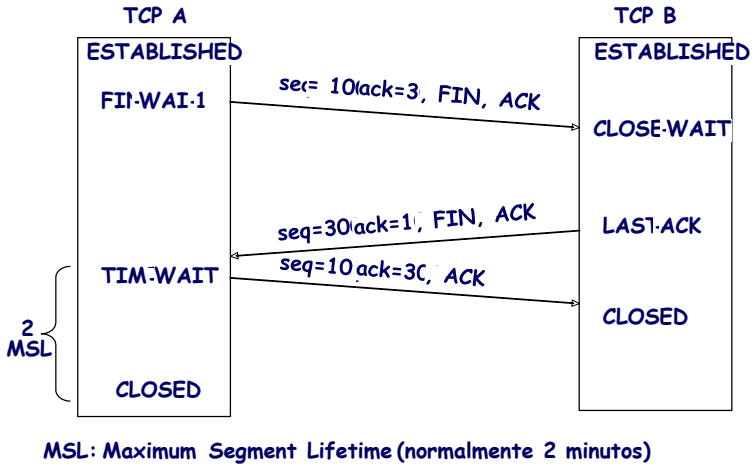


"Apertura activa": cliente
"Apertura pasiva": servidor
Campos involucrados:
* Bit S (SYN) del campo control
* Campo secuencia (valor aleatorio)
* Campo acuse
* Bit A (ACK) del campo control

Control de la conexión: Cierre de la conexión, liberación de recursos.



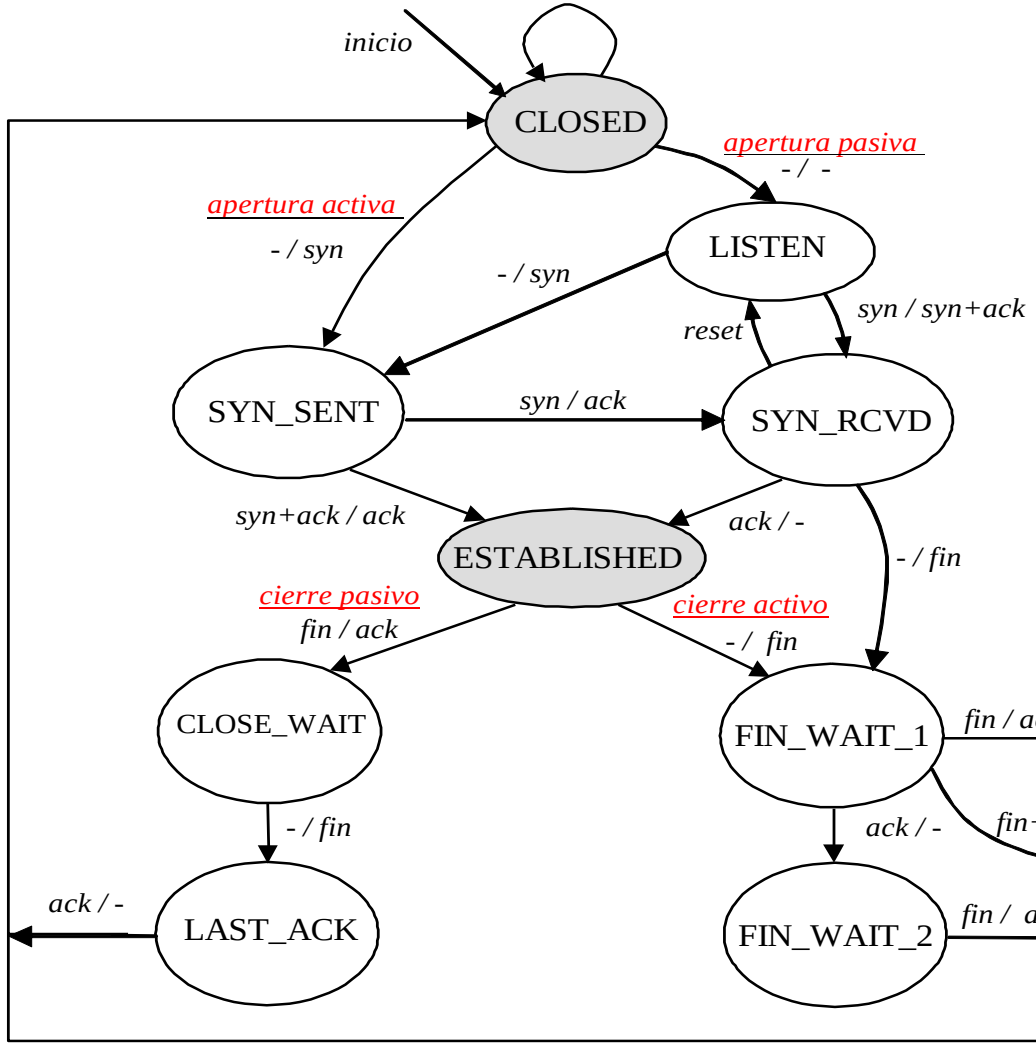
- "Cierre activo"
- "Cierre pasivo"
- Campos involucrados:
  - * Bit F (FIN) del campo control
  - * Campo secuencia
  - * Campo acuse
  - * Bit A (ACK) del campo control



MSL: Maximum Segment Lifetime (normalmente 2 minutos)

Control de la conexión. Números de secuencia.

- Campo de 32 bits ➔ 2^{32} valores
- Inicialización (#secuencia = ISN)
  - Empieza en el ISN (Initial Sequence Number), elegido por el sistema.
  - Para el ISN, el estándar sugiere utilizar un contador entero incrementado en 1 cada 4 µs aproximadamente ➔ Ciclo al cabo de 4 horas 46 min.
  - Protege de coincidencias, pero no de sabotajes.
- Incremento (#secuencia = #secuencia + Nbytes)
  - Se incrementa según los bytes de carga útil (payload)
  - Los flags SYN y FIN incrementan en 1 el número de secuencia.



Autómata de estados finitos TCP

Leyenda: *a/b*
Segmento a recibido, segmento b transmitido.

9. Se desea transferir con protocolo TCP un archivo de L bytes usando un MSS de 536.

a) **¿Cuál es el valor máximo de L tal que los números de secuencia de TCP no se agoten?**

b) **Considerando una velocidad de transmisión de 155 Mbps y un total de 66 bytes para las cabeceras de las capas de transporte, red y enlace de datos, e ignorando limitaciones debidas al control de flujo y congestión, calcule el tiempo que se tarda en transmitir el archivo en A.**

Control de errores y de flujo:

Mejorar rendimiento ⇒ ventana deslizante.

Control de errores: esquema ARQ con confirmaciones positivas y acumulativas.

Campos involucrados:

Campo *secuencia*: *offset* (en bytes) dentro del mensaje.

Campo *acuse*: número de byte esperado en el receptor.

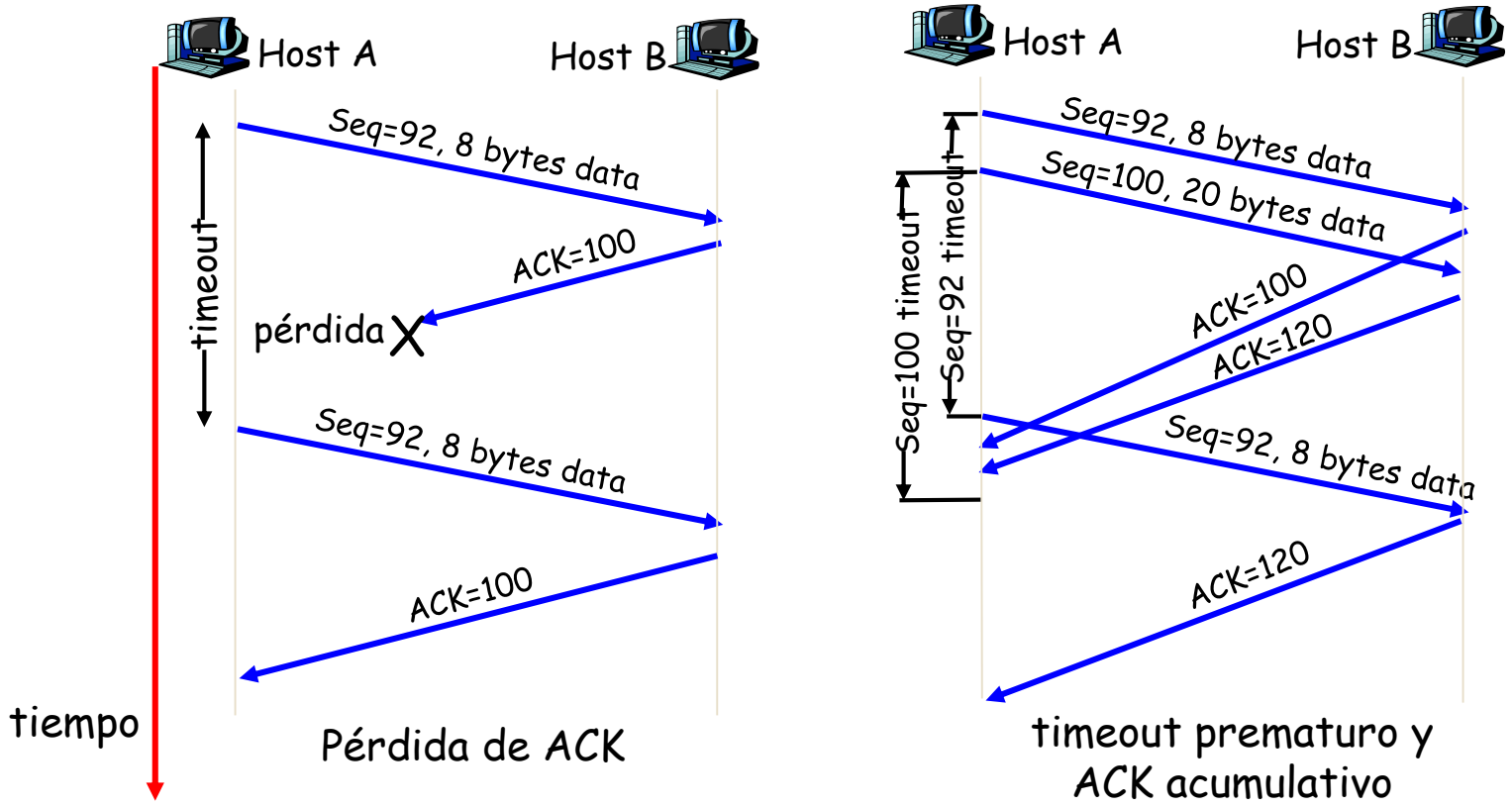
Bit *A* (ACK) del campo de *control.*

Campo *comprobación*: *checksum* de todo el segmento y uso de pseudo-cabecera

Iporigen		
IPdestino		
00...00	protocolo	longitudTCP

Control de errores y de flujo:

Control de errores: escenarios de retransmisión (gráficas © James F. Kurose).



Control de errores y de flujo:

Control de errores: generación de ACKs (RFC 1122, 2581).

Evento	Acción del TCP receptor
Llegada ordenada de segmento, sin discontinuidad, todo lo anterior ya confirmado.	Retrasar ACK. Esperar recibir al siguiente segmento hasta 500 mseg. Si no llega, enviar ACK.
Llegada ordenada de segmento, sin discontinuidad, hay pendiente un ACK retrasado.	Inmediatamente enviar un único ACK acumulativo.
Llegada desordenada de segmento con # de sec. mayor que el esperado, discontinuidad detectada.	Enviar un ACK duplicado, indicando el # de sec. del siguiente byte esperado.
Llegada de un segmento que completa una discontinuidad parcial o totalmente.	Confirmar ACK inmediatamente si el segmento comienza en el extremo inferior de la discontinuidad.

10. Los hosts A y B se están comunicando a través de una conexión TCP y B ya ha recibido y confirmado todos los bytes hasta el byte 126. Suponga que a continuación el host A envía dos segmentos seguidos a B que contienen, respectivamente, 70 y 50 bytes de datos. El envío de A es ordenado, el número de puerto origen en dichos segmentos es 302 y el de destino el 80. El host B envía una confirmación inmediata a la recepción de cada segmento de A, sin esperar el retardo de 500 ms del estándar.

a) **Especifique los números de secuencia de ambos segmentos.**

b) **Si el primer segmento llega antes que el segundo ¿cuál es el número de acuse y los puertos origen y destino en el primer ACK que se envía?**

c) **Si el segundo segmento llega antes que el primero ¿cuál es el número de acuse y los puertos origen y destino en el primer ACK que envía?**

d) **Imagine que los segmentos llegan en orden pero se pierde el primer ACK.**

Control de errores y de flujo:

Control de errores: ¿cómo estimar los "**timeouts**"?

Mayor que el tiempo de ida y vuelta (RTT).

Si es demasiado **pequeño**: **timeouts prematuros**.

Si es demasiado **grande**: **reacción lenta** a pérdida de segmentos.

Para situaciones cambiantes la mejor solución es la adaptable.

Control de errores y de flujo:

Control de errores: ¿cómo estimar los "**timeouts**"?

Kurose & Ross

***RTTmedido**: tiempo desde la emisión de un segmento hasta la recepción del ACK.*

$$RTTnuevo = (1-\alpha) \times RTTviejo + \alpha \times RTTmedido \;,\; \alpha \;\&\; \beta \in [0,1]$$

$$Desviacion_{nueva} = (1-\beta) \times Desviacion_{vieja} + \beta \times |\, RTTmedido - RTTnuevo \,|$$

$$Timeout = RTTnuevo + 4 * Desviacion$$

Problema con ACKs repetidos: ambigüedad en la interpretación.

Solución: Algoritmo de Karn:

- *actualizar el RTT sólo para los no repetidos*
- si hay que repetir un segmento incrementar el timeout: $tout_{nuevo} = \gamma \cdot tout_{viejo}$, $\gamma = 2$.

Examen sept 2013, ejercicio 2

Si el RTT es 30 ms, la Desviación es 2 ms y se reciben 4 ACKs con un valor de acuse de 202, 402, 604 y 604 tras 26, 32, 32 y 24 ms, respectivamente

¿Cuál será el nuevo RTT, Desviación y timeout?

Usar α=0,125 y β=0,25.

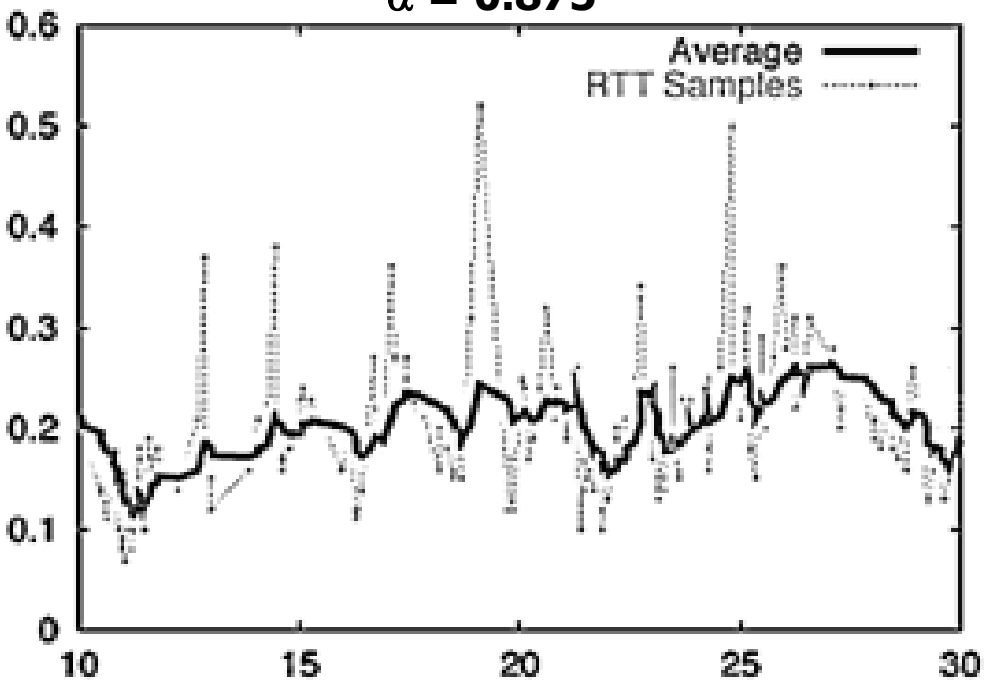
¿Y si se utiliza el algoritmo de Karn?

Control de errores y de flujo:

Control de errores: ¿cómo estimar los "timeouts"?

Ejemplo de RTT medidos y estimados entre Amherst, Massachusetts y St. Louis, Missouri.

α = 0.875



Control de errores y de flujo:

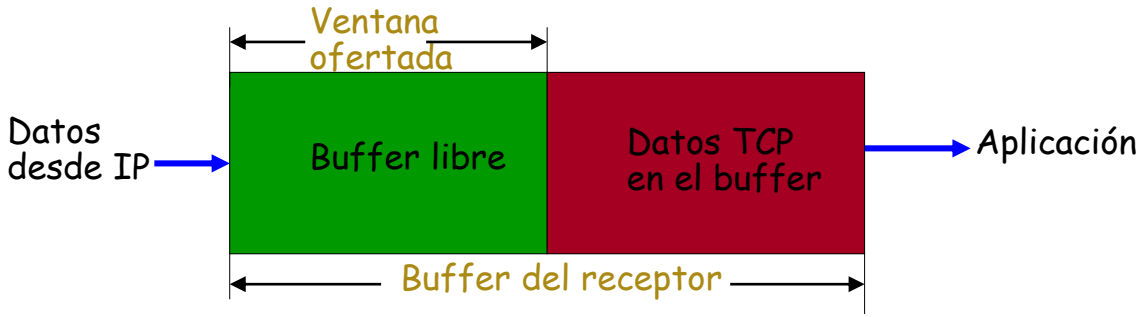
Control de flujo:

Procedimiento para evitar que el emisor sature al receptor

Es un esquema crediticio: el receptor avisa al emisor de lo que puede aceptar

Se utiliza el campo *ventana (WINDOW)* en el segmento TCP para establecer la ventana ofertada

En el receptor (paso 1):

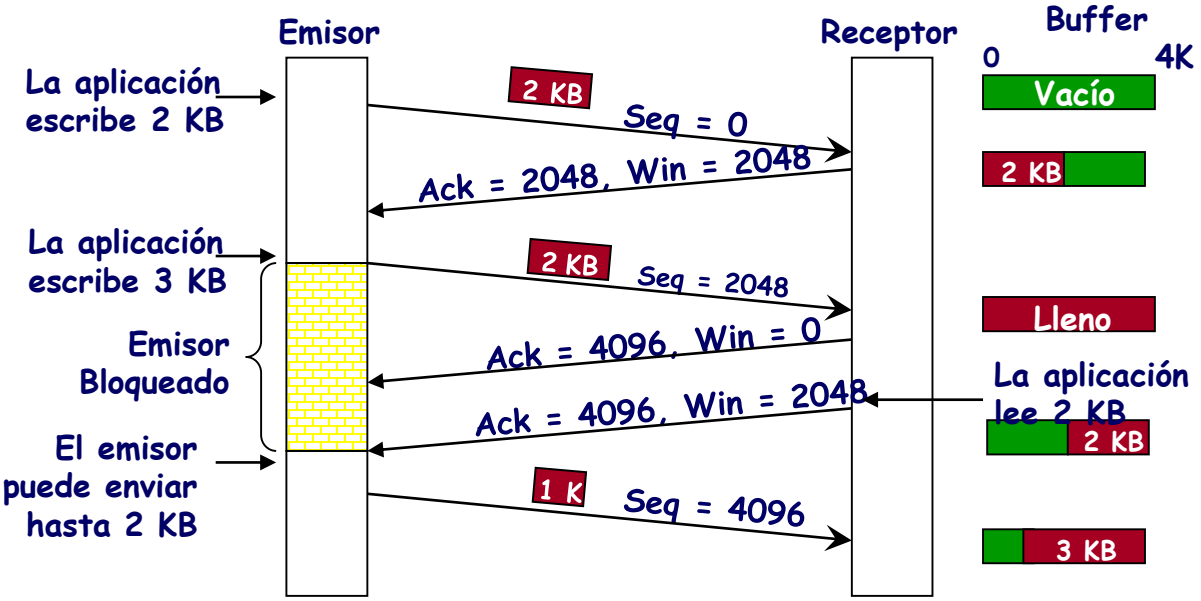


En el emisor (paso 2):

ventana útil emisor = ventana ofertada receptor – bytes en tránsito

Control de errores y de flujo:

Control de flujo:



Posible problema: *síndrome de la ventana tonta (RFC 813) si se utilizan segmentos muy pequeños.*

Posible mejora: *la ventana optimista* (*RFC 813*)

ventana útil emisor = ventana ofertada receptor

Es posible hacer entregas "no ordenadas": Bit *U* (URG), campo *puntero.*

Solicitar una entrega inmediata a la aplicación: bit *P* (PSH).

Control de congestión (RFC 2001):

Procedimiento para evitar que el emisor sature la red:

- ancho de banda de las líneas
- buffer en dispositivos de interconexión

Solución: limitar el tráfico generado (= ctrl de flujo)

Problema: No hay "ente" que avise de lo que se puede enviar (≠ ctrl de flujo)

¿Entonces? Usar una medida indirecta ➔ pérdidas y/o retrasos en los ACKs.

Control de congestión:

En el emisor se utilizan una **ventana** y un **umbral**.

TCP Tahoe

Inicialmente

VCongestion = MSS (maximum segment size)

Umbral a un cierto valor

Si *VCongestion* < *Umbral*, por cada ACK recibido

VCongestion += MSS **(crecimiento exponencial)**

Inicio lento

Si *VCongestion* > *Umbral*, por cada ventana completada (todos ACKs recibidos)

VCongestion += MSS (**crecimiento lineal**)

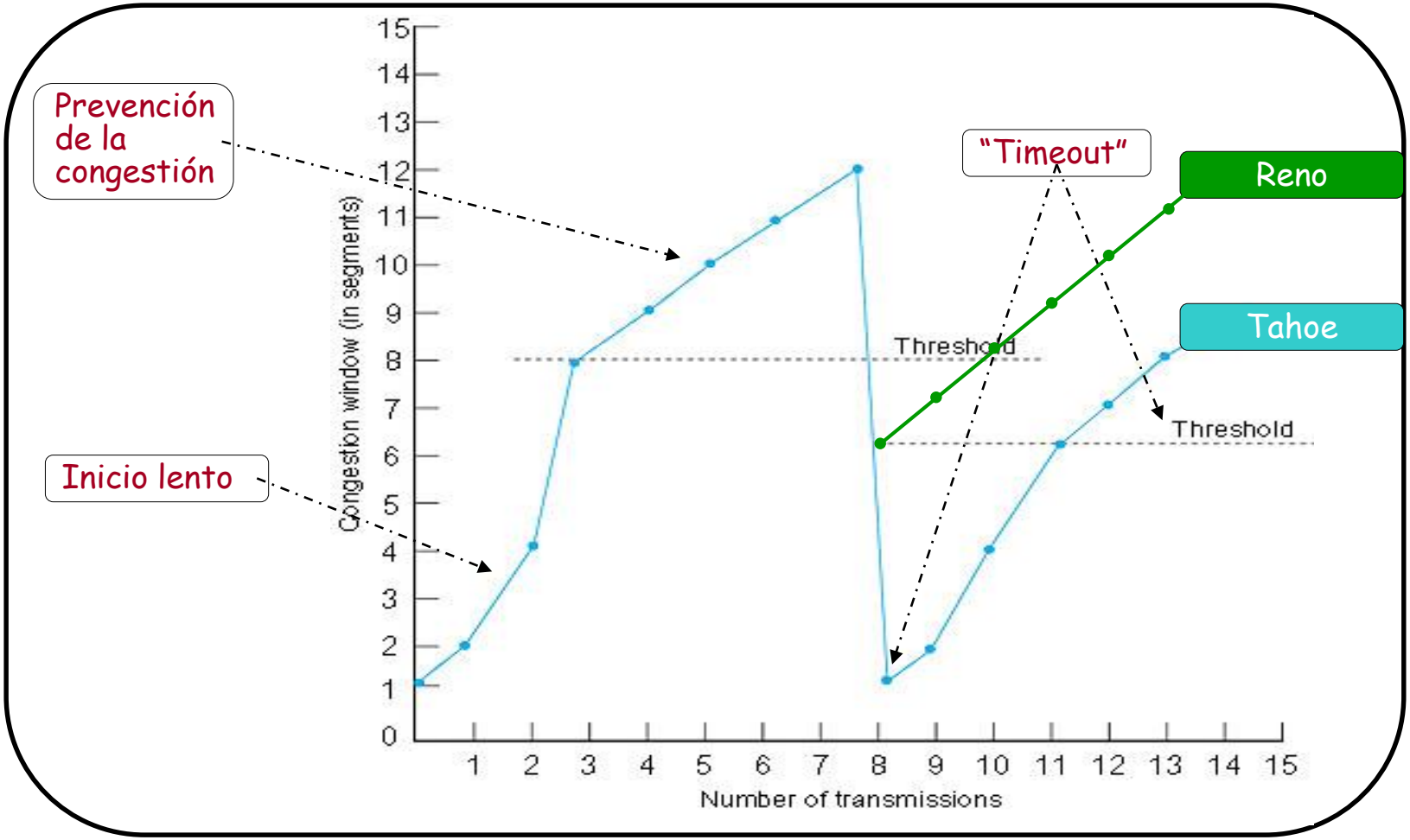
Prevención de la congestión

Si hay timeout entonces

Umbral = *VCongestion*/2

VCongestion = MSS

Control de congestión (gráfica © James F. Kurose):



15. Teniendo en cuenta el efecto del inicio lento, en una línea sin congestión con 10 ms de tiempo de propagación, 1 Mbps de velocidad de transmisión y un MSS de 2KB, ¿cuánto tiempo se emplea en enviar 24 KB?

Combinación del Ctrl de flujo y congestión:

El emisor elige el mínimo de las ventanas correspondientes:

```
Bytes_permitidos_enviar =
            min{VCongestion,VentanaOfertadaReceptor}
```

```
ventana útil emisor = Bytes_permitidos_enviar – bytes en tránsito
```

1. Introducción.
2. Protocolo de datagrama de usuario (UDP).
3. Protocolo de control de transmisión (TCP).
    1. Multiplexación/demultiplexación.
    2. Control de conexión.
    3. Control de errores y de flujo.
    4. Control de congestión.
4. **Extensiones TCP.**
5. Ejercicios.

TRANSMISSION CONTROL PROTOCOL (TCP)

- TCP se define con mútiples "Sabores"
- Los diferentes sabores no afectan a la interoperabilidad entre los extremos
- Desde cualquier versión de Linux con kernel mayor que la 2.6.19 se usa por defecto TCP CuBIC

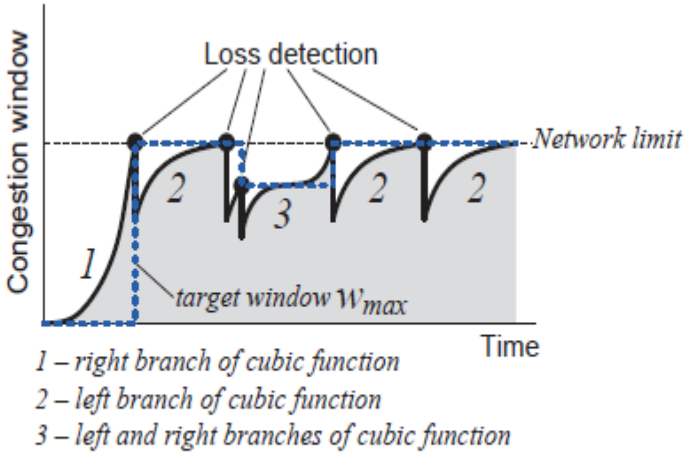
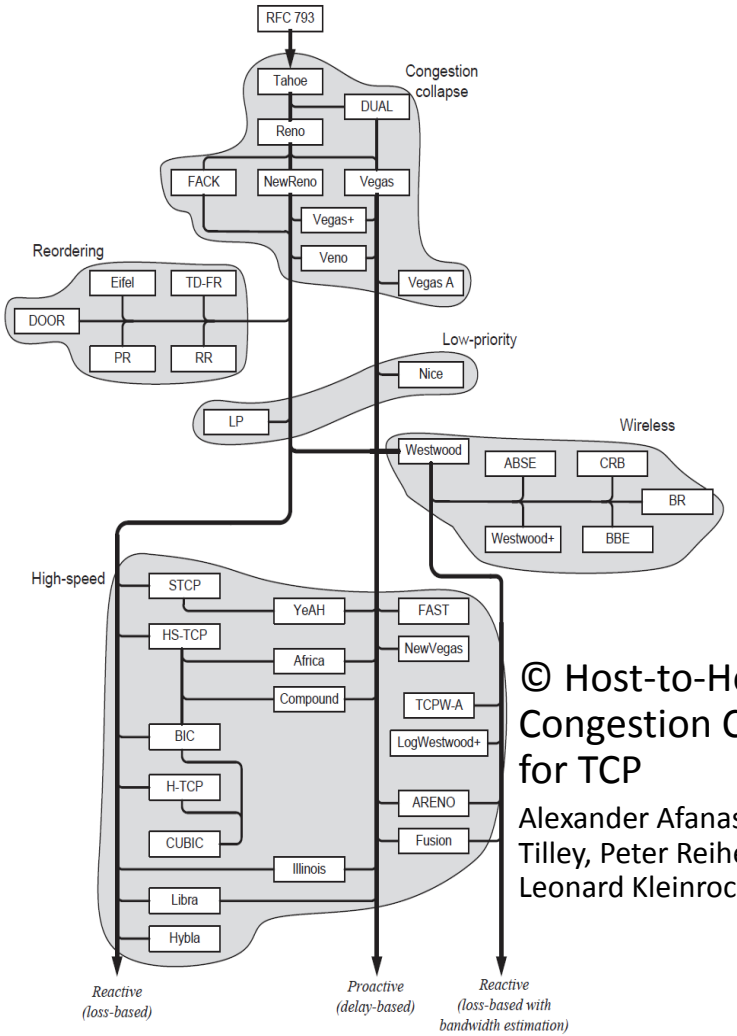


Fig. 50. Congestion window dynamics in CUBIC



57. Evolutionary graph of variants of TCP congestion control.

© Host-to-Host Congestion Control for TCP

Alexander Afanasyev, Neil Tilley, Peter Reiher, and Leonard Kleinrock

EXTENSIONES TCP

Adaptación de TCP a redes actuales (RFC 1323, 2018).

Ventana escalada:

Opción TCP en segmentos SYN:

Hasta $2^{14}x2^{16}$ bytes (=2^{30} bytes=1GB) autorizados.

Estimación RTT:

Opción TCP de *sello de tiempo*, en todos los segmentos.

PAWS ("Protect Against Wrapped Sequence numbers"):

Sello de tiempo y rechazo de segmentos duplicados.

SACK:

Confirmaciones selectivas.

1. Introducción.
2. Protocolo de datagrama de usuario (UDP).
3. Protocolo de control de transmisión (TCP).
    1. Multiplexación/demultiplexación.
    2. Control de conexión.
    3. Control de errores y de flujo.
    4. Control de congestión.
4. Extensiones TCP.
5. **Ejercicios.**

Examen ejemplo, ejercicio 2

Al inicio de una conexión TCP, en una línea sin congestión con 10 ms de tiempo de propagación y 10 Mbps de velocidad de transmisión, ¿cuánto tiempo se emplea en enviar y recibir confirmación de 30 KB con las siguientes asunciones (añada cualquier asunción adicional que crea conveniente)? Realice el diagrama de tiempos de la transmisión.

Ventana ofertada de control de flujo de 10 KB

Todos los segmentos se ajustan a un MSS de 2KB

Umbral de congestión de 8 KB

Respuesta ACK retardada en el receptor de acuerdo a la teoría.

Examen feb 2014, ejercicio 3

(Ver en examen)

Examen sep 2015, ejercicio 3

(Ver en examen)

TEMA 3
CAPA DE TRANSPORTE EN INTERNET

Fundamentos de Redes

2015/2016