

Tema 3 Capa de transporte

Fundamentos de Redes

Grado en Ingeniería Informática y dobles grados
Curso 3º

Jorge Navarro Ortiz

Departamento de Teoría de la Señal, Telemática y Comunicaciones
E.T.S. Ingenierías Informática y Telecomunicación – Universidad de Granada
C/ Periodista Daniel Saucedo Aranda, s/n - 18071 – Granada (Spain)
Teléfono: +34-958 241000, ext 20042 - Fax: +34-958 243032 - Email: jorgenavarro@ugr.es

© 2022



1

Tema 3. Capa de transporte en Internet



Esquema

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.

Fundamentos de Redes - Grado en Ingeniería Informática y dobles grados
© 2022 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

2



2



Tema 3. Capa de transporte en Internet

Objetivos del tema

- Comprender las funcionalidades y servicios de la capa de transporte :
- Servicio de **multiplexación/demultiplexación**.
 - Servicio **orientado a conexión** frente a **no orientado a conexión**.
 - Cómo conseguir una transferencia de datos **fiable**.
 - Cómo proporcionar **control de flujo**.
 - Cómo proporcionar **control de congestión**.
 - Cómo se han implementado estas funcionalidades en Internet.

3



3



Tema 3. Capa de transporte en Internet

Bibliografía



Capítulo 10, Pedro García Teodoro, Jesús Díaz Verdejo y Juan Manuel López Soler. **TRANSMISIÓN DE DATOS Y REDES DE COMPUTADORES**, Ed. Pearson, 2017, ISBN: 978-0-273-76896-8



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.

4



4



Tema 3. Capa de transporte en Internet

Esquema

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.

5



5



Tema 3. Capa de transporte en Internet

1. Introducción.

- Funciones y servicios de la capa de transporte:
 - Comunicación **extremo a extremo** (end-to-end). → No se realiza nada salto a salto
 - Multiplexación/demultiplexación de aplicaciones → puerto.
- Protocolo UDP:
 - Multiplexación/demultiplexación de aplicaciones.
 - Servicio **no orientado a conexión, no fiable**. → Se pueden perder paquetes, desordenarlos... etc
- Protocolo TCP:
 - Multiplexación/demultiplexación de aplicaciones.
 - Servicio **orientado a conexión, fiable**:
 - Control de errores y de flujo.
 - Control de la conexión.
 - Control de congestión.

■ Extensiones TCP

6



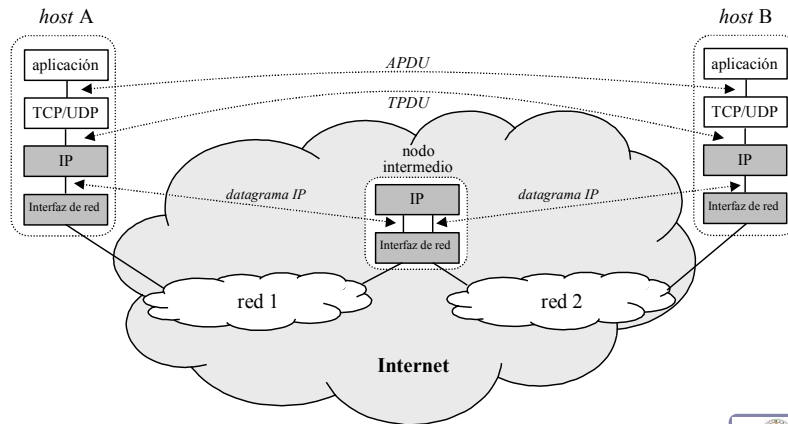
6



Tema 3. Capa de transporte en Internet

1. Introducción.

Comunicación extremo a extremo (end-to-end):



7

7



Tema 3. Capa de transporte en Internet

Esquema

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.

8

8



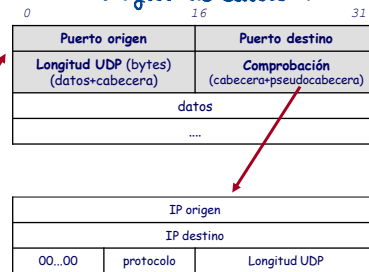
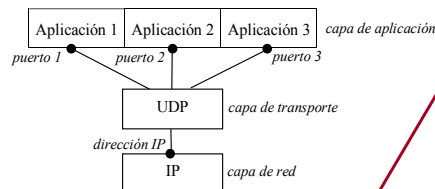
Tema 3. Capa de transporte en Internet

2. Protocolo de datagrama de usuario (UDP).

■ "User Datagram Protocol": RFC 768.

■ Funcionalidad "best-effort":

- Servicio **no orientado a conexión**: no hand-shaking, no hay retardos de establecimiento, cada TPDU es independiente.
- Servicio **no fiable**: puede haber pérdidas.
- 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.



9 ■ Datagrama de usuario UDP.

9



Tema 3. Capa de transporte en Internet

2. Protocolo de datagrama de usuario (UDP).

■ Multiplexación/demultiplexación: transportar las TPDU al proceso correcto.

- Existen **puertos preasignados** con servicios normalizados:

Ejemplos de
puertos UDP
preasignados

Puerto	Aplicación/Servicio	Descripción
7	echo	Eco
13	daytime	Fecha
37	time	Hora
42	nameserver	Servicio de nombres
53	domain DNS	Servicio de nombres de dominio
69	ftp	Transferencia simple de ficheros
123	ntp	Protocolo de tiempo de red

Se consultan en etcservicios.en.unix

Para que la hora del sistema sea segura de internet

- Otros **puertos (>1024)** están a **libre disposición** del desarrollador.

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

■ Cada segmento UDP se **encapsula** en un datagrama IP.

10

10



Tema 3. Capa de transporte en Internet

Esquema

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.

Fundamentos de Redes - Grado en Ingeniería Informática y dobles grados
© 2022 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

11



11



Tema 3. Capa de transporte en Internet

3. Protocolo de control de transmisión (TCP).

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

- Servicio **punto a punto**.
- Servicio **orientado a conexión** (exige un estado común entre el emisor y el receptor: "hand-shaking").
- **Entrega ordenada** de las secuencias de bytes generadas por la aplicación ("stream oriented").
- Transmisión **full-duplex**. → Puedo transmitir en los 2 sentidos
- Mecanismo de **detección y recuperación de errores** (ARQ) con confirmaciones positivas **ACKs** (acumulativas) y "timeouts" adaptables. *automatic repeat request*
- **Servicio fiable** → control de congestión y control de flujo con ventanas deslizantes con tamaño máximo adaptable.
- **Incorporación de confirmaciones** ("piggybacking").

Fundamentos de Redes - Grado en Ingeniería Informática y dobles grados
© 2022 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

12



12



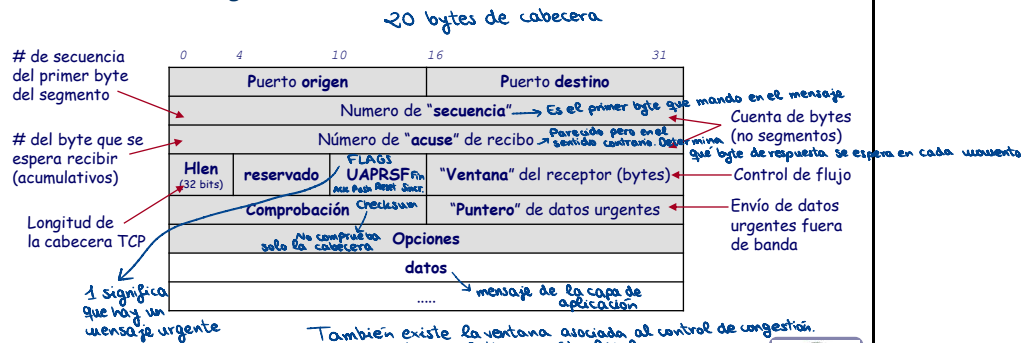
Tema 3. Capa de transporte en Internet

3. Protocolo de control de transmisión (TCP).

Funcionalidades de TCP:

- Multiplexación/demultiplexación de aplicaciones.
- Control de la conexión (establecimiento y cierre).
- Control de errores y de flujo.
- Control de congestión.

TPDU TCP = Segmento TCP:



13

Cada segmento TCP se encapsula en un datagrama IP.



13



Tema 3. Capa de transporte en Internet

3.1. TCP. Multiplexación / demultiplexación.

Multiplexación/demultiplexación de aplicaciones:

- Transportar las TPDU al proceso correcto.
 - Existen puertos preasignados con servicios normalizados:

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

- Otros puertos (>1024) están a libre disposición del desarrollador.
- La "conexión TCP" se identifica por: puerto e IP origen y puerto e IP destino.

14



14



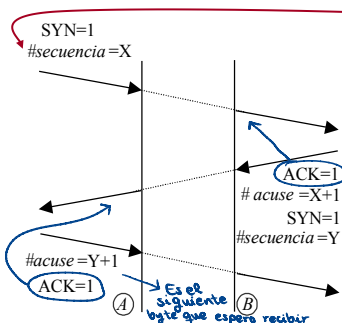
Tema 3. Capa de transporte en Internet

3.2. TCP. Control de conexión.



Control de la conexión:

- TCP ofrece un servicio **orientado a conexión**.
- El intercambio de información tiene **tres fases**:
 - **Establecimiento** de la conexión (sincronizar # de secuencia y reservar recursos).
 - Intercambio de **datos** (full-duplex).
 - **Cierre** de la conexión (liberar recursos).
- ¿Es posible **garantizar** un establecimiento/cierre **fiable** de la conexión sobre un servicio (IP) no fiable? **NO**.
- **Establecimiento** de la conexión: *three-way handshake*.



"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

Fundamentos de Redes - Grado en Ingeniería Informática y dobles grados
© 2022 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

15

15



Tema 3. Capa de transporte en Internet

3.2. TCP. Control de conexión.



Control de la conexión. Números de secuencia.

- El **número de secuencia** es un campo de 32 bits que cuenta bytes en módulo 2^{32} (el contador se da la vuelta cuando llega al valor máximo).
- El número de secuencia no empieza normalmente en 0, sino en un valor denominado **ISN** (Initial Sequence Number) elegido "teóricamente" al azar; para evitar confusiones con solicitudes anteriores.
- El ISN es elegido por el sistema (cliente o servidor). El estándar sugiere utilizar un contador entero incrementado en 1 cada $4 \mu s$ aproximadamente. En este caso el contador se da la vuelta (y el ISN reaparece) al cabo de 4 horas 46 min.
- El mecanismo de selección de los ISN es suficientemente fiable para proteger de coincidencias, pero no es un mecanismo de protección frente a sabotajes. Es muy **fácil averiguar el ISN** de una conexión e interceptarla suplantando a alguno de los dos participantes.
- TCP **incrementa el número de secuencia** de cada segmento según los bytes que tenía el segmento anterior, con una sola excepción:
 - Los flags **SYN** y **FIN**, cuando están puestos, incrementan en 1 el número de secuencia.
- La presencia del flag **ACK** no incrementa el número de secuencia.

Fundamentos de Redes - Grado en Ingeniería Informática y dobles grados
© 2022 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

16

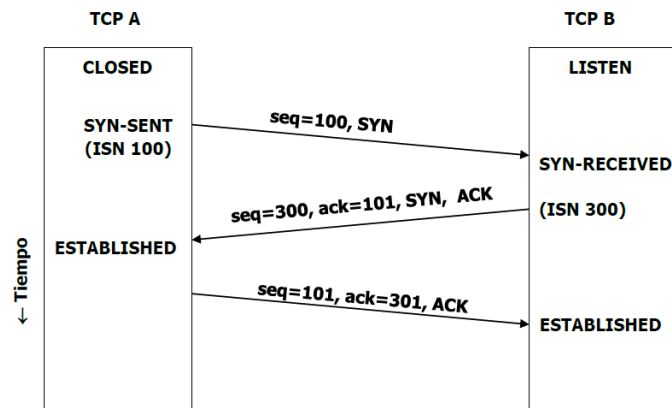
16



Tema 3. Capa de transporte en Internet

3.2. TCP. Control de conexión.

Establecimiento de la conexión. Caso **sin incidencias** (normal):



17

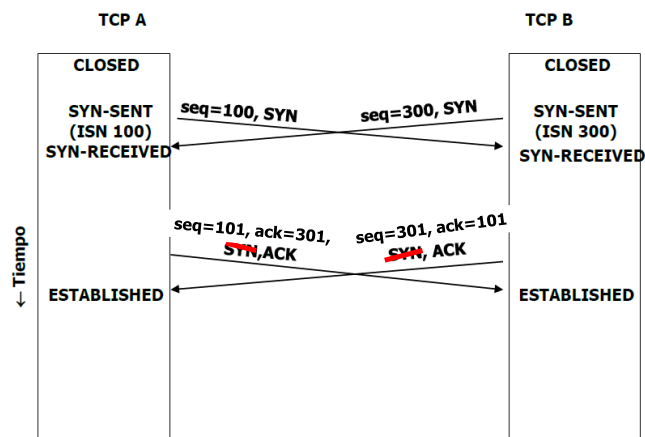
17



Tema 3. Capa de transporte en Internet

3.2. TCP. Control de conexión.

Establecimiento de la conexión. Caso **de conexión simultánea**:



18

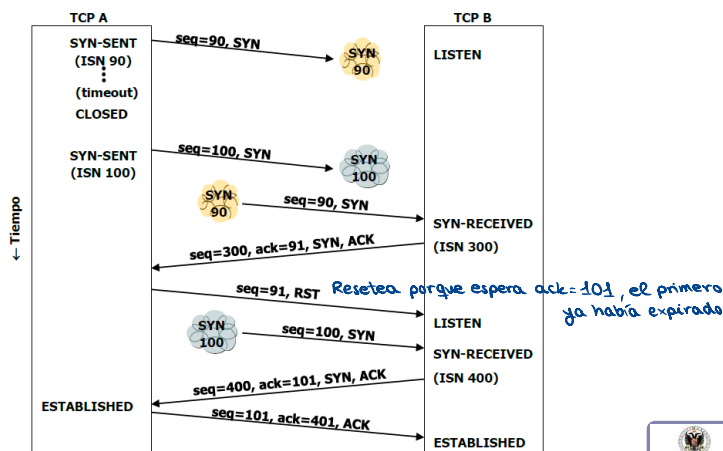
18



Tema 3. Capa de transporte en Internet

3.2. TCP. Control de conexión.

Establecimiento de la conexión. Caso con **SYN retrasados y duplicados**:



19

19

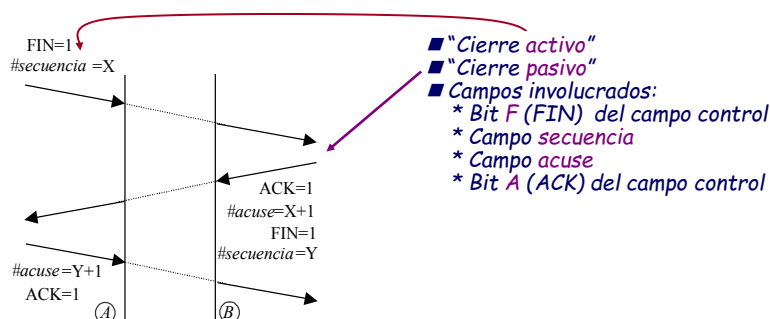


Tema 3. Capa de transporte en Internet

3.2. TCP. Control de conexión.

Control de la conexión:

Cierre de la conexión: liberación de recursos.



Para evitar bloqueos por pérdidas, una vez comenzado el procedimiento de CIERRE se usan timeouts (ver diagrama de estados: *Maximum Segment LifeTime* = 2 min).

20

20



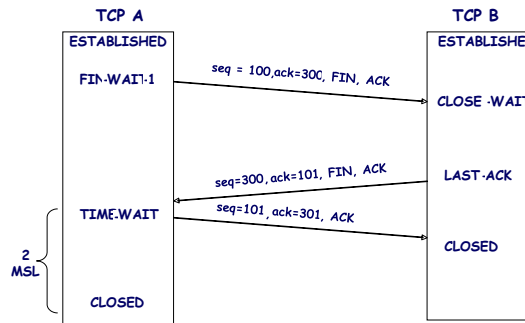
Tema 3. Capa de transporte en Internet

3.2. TCP. Control de conexión.



Control de la **conexión**:

■ **Cierre** de la conexión: caso normal.



MSL: Maximum Segment Lifetime (normalmente 2 minutos)

Hay otras posibilidades de cierre de la conexión (ver el diagrama de estados siguiente).



21

21



Tema 3. Capa de transporte en Internet

3.2. TCP. Control de conexión.

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

- ¿Cuál es el valor máximo de L tal que los números de secuencia de TCP no se agoten?
- 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.



22

22



Tema 3. Capa de transporte en Internet

3.3. TCP. Control de errores y de flujo.

11. 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.

- Especifique los números de secuencia de ambos segmentos.
- 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?
- 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?
- Imagine que los segmentos llegan en orden pero se pierde el primer ACK.

Fundamentos de Redes - Grado en Ingeniería Informática y dobles grados
© 2022 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

27



27



Tema 3. Capa de transporte en Internet

3.3. TCP. Control de errores y de flujo.

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:

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

$$RTT_{nuevo} = \alpha \cdot RTT_{viejo} + (1-\alpha) \cdot RTT_{medido}, \alpha \in [0,1]$$

$$Desviacion_{nueva} = (1-x) \cdot Desviacion_{vieja} + x \cdot |RTT_{medido} - RTT_{nuevo}|$$

$$Timeout = RTT_{nuevo} + 4 \cdot Desviacion$$

- Problema con ACKs repetidos: ambigüedad en la interpretación.
- Solución: Algoritmo de Karn, actualizar el RTT sólo para los no ambiguos, pero si hay que repetir un segmento incrementar el timeout:

$$tout_{nuevo} = \gamma \cdot tout_{viejo}, \gamma = 2.$$

Fundamentos de Redes - Grado en Ingeniería Informática y dobles grados
© 2022 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

28



28



Tema 3. Capa de transporte en Internet

3.3. TCP. Control de errores y de flujo.

15. Si el RTT es 30 ms, la Desviación es 2 ms y se reciben ACKs tras 26, 32 y 24 ms, ¿Cuál será el nuevo RTT, Desviación y timeout? Usar $\alpha=0,125$ y $\beta=0,25$. ¿Y si los dos primeros ACKs tienen el mismo número de acuse y se usa el algoritmo de Karn?

Fundamentos de Redes - Grado en Ingeniería Informática y dobles grados
© 2022 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

29



29



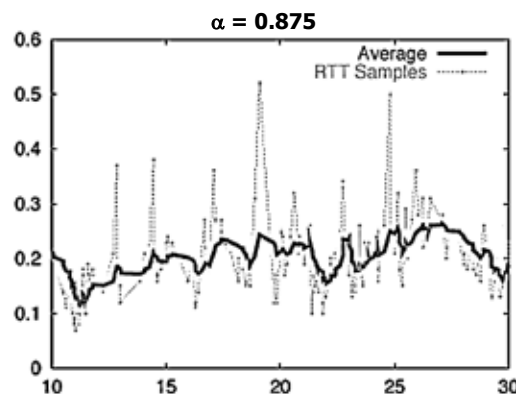
Tema 3. Capa de transporte en Internet

3.3. TCP. Control de errores y de flujo.

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.



Fundamentos de Redes - Grado en Ingeniería Informática y dobles grados
© 2022 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

30



30



Tema 3. Capa de transporte en Internet

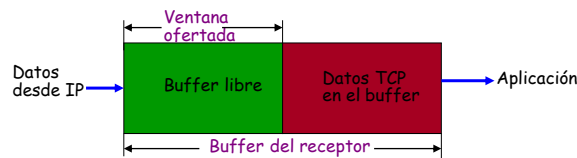
3.3. TCP. Control de errores y de flujo.

Control de errores y de flujo:

Control de flujo:

- Procedimiento para evitar que el emisor **sature** al receptor con el envío de demasiada información y/o demasiado rápido.
- Es un **esquema crediticio**: el receptor informa al emisor sobre los bytes autorizados a emitir sin esperar respuesta.
- Se utiliza el campo **ventana**:

$\text{ventana útil emisor} = \text{ventana ofertada receptor} - \text{bytes en tránsito}$



31



31

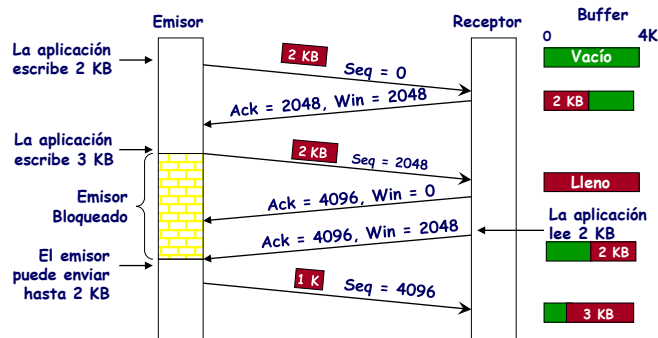


Tema 3. Capa de transporte en Internet

3.3. TCP. Control de errores y de flujo.

Control de errores y de flujo:

Control de flujo:



- ¿Alguna debilidad en el control de flujo?
- ¿Y si pierde el anuncio de $\text{WIN} = 2048$? → ¡Bloqueo! ¿Cómo evitarlo? → temporizador de persistencia

32



32

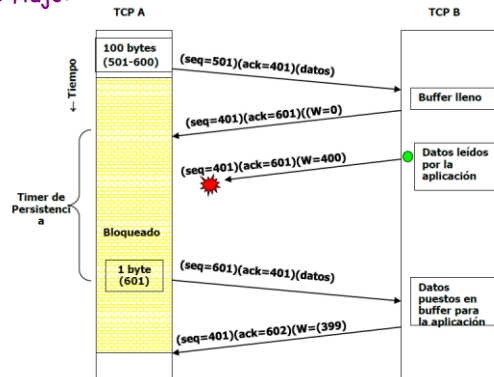


Tema 3. Capa de transporte en Internet

3.3. TCP. Control de errores y de flujo.

Control de errores y de flujo:

- Control de flujo:
Temporizador de persistencia



- Possible problem: *síndrome de la ventana tonta (RFC 813)* if small segments are used.
- Possible improvement: *la ventana optimista (RFC 813)*.
- It is possible to make deliveries "no ordenadas": Bit *U* (URG), field *puntero*.
- Request an immediate delivery to the application: bit *P* (PSH).



Tema 3. Capa de transporte en Internet

3.4. TCP. Control de congestión.

Control de congestión (RFC 2001):

- It is a problem due to the *insuficiencia de recursos* (bandwidth of the lines as buffer in routers and final systems).
- It is a problem *diferente al control del flujo*: involves the network and the final systems.
- It has a nature *adelante-atrás*.
- It manifests in *pérdidas y/o retrasos* in the ACKs.
- Solution: in the source *limitar* in an adaptable way the *tráfico* generated.





Tema 3. Capa de transporte en Internet

3.4. TCP. Control de congestión.



Control de congestión:

En el emisor se utilizan dos **ventanas** y un **umbral**.

Bytes_permitidos_enviar =
 $\min\{\text{VentanaCongestion}, \text{VentanaDelReceptor}\}$

VentanaDelReceptor: utilizada para el control de flujo (de tamaño variable) según el campo "ventana" recibido.

VentanaCongestion:
Inicialmente $\text{VentanaCongestion} = 1$ (num. segmentos)

Inicio lento

Si $\text{VentanaCongestion} < \text{umbral}$, por cada ACK recibido
 $\text{VentanaCongestion}++$ (crecimiento exponencial)

Prevención de la congestión

Si $\text{VentanaCongestion} > \text{umbral}$, cada vez que se recibe todos los ACKs pendientes
 $\text{VentanaCongestion}++$ (crecimiento lineal)

Si hay timeout entonces
 $\text{umbral} = \text{VentanaCongestion} / 2$ y $\text{VentanaCongestion} = 1$

Fundamentos de Redes - Grado en Ingeniería Informática y dobles grados
© 2022 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

35



35

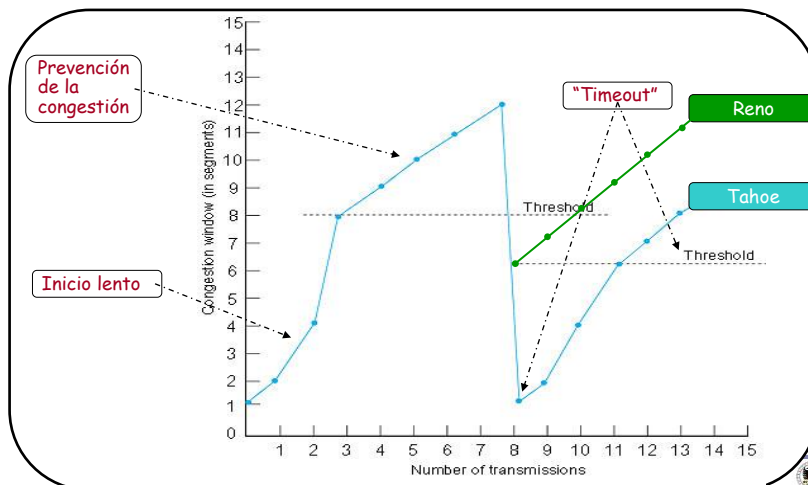


Tema 3. Capa de transporte en Internet

3.4. TCP. Control de congestión.



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



Fundamentos de Redes - Grado en Ingeniería Informática y dobles grados
© 2022 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

36



36



Tema 3. Capa de transporte en Internet

Esquema

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.

Fundamentos de Redes - Grado en Ingeniería Informática y dobles grados
© 2022 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

37



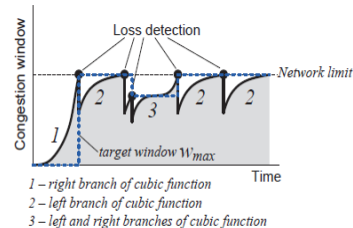
37



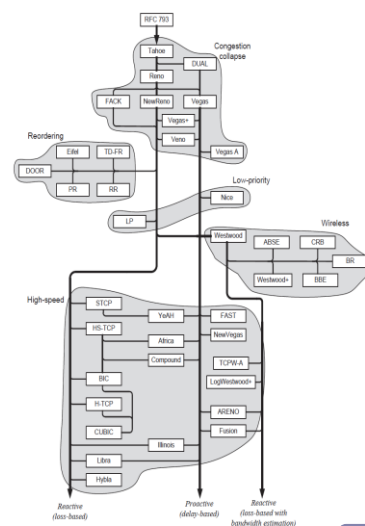
Tema 3. Capa de transporte en Internet

4. Extensiones TCP.

- TCP se define con múltiples "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**



38 Fig. 50. Congestion window dynamics in CUBIC



very graph of variants of TCP congestion control.



38



Tema 3. Capa de transporte en Internet

4. Extensiones TCP.

Adaptación de TCP a redes actuales (RFC 1323).

Ventana escalada:

- Opción TCP en segmentos SYN:
- Hasta $2^{14} \times 2^{16}$ 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.

Referencias:

- RFCs
- `/usr/src/linux-2.../net/ipV4/tcp.c`
- `/usr/include/netinet/tcp.h`
- <http://www.cs.columbia.edu/~hgs/internet/tcp.html>
- Herramientas de análisis: `ethereal/wireshark` y `tcpdump`.

39



39



Tema 3. Capa de transporte en Internet

Esquema

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.

40



40



Tema 3. Capa de transporte en Internet

5. Ejercicios

10. Considere un enlace con una velocidad de transmisión de 1 Mbps, un tiempo de ida y vuelta (RTT) de 30 ms y segmentos fijos de 1500 bytes, incluyendo cabeceras y datos, ¿cuál tiene que ser el tamaño de la ventana para que la eficiencia en la transmisión (ratio entre el tiempo de transmisión y el tiempo total en el emisor) sea de al menos un 95%?

Fundamentos de Redes - Grado en Ingeniería Informática y dobles grados
© 2022 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

41



41



Tema 3. Capa de transporte en Internet

5. Ejercicios

17. Suponiendo que la ventana de congestión es 18 KB y que se dispara un *timeout* ¿Cuánto será la ventana de congestión si las 4 siguientes ráfagas de transmisiones, donde se envía la ventana completa, son exitosas? Suponed que el MSS es 1 KB.

Fundamentos de Redes - Grado en Ingeniería Informática y dobles grados
© 2022 v1.0 - Juan M. López Soler y Jorge Navarro Ortiz

42



42