

Tema 4: El nivel de transporte

Sesión A8

Clase 8: Introducción al nivel de transporte. Direccionamiento. UDP

- **Lectura previa:**
 - [Kurose2010, secciones 3.1, 3.2 y 3.3](#)
- **Consulta:**
 - <http://www.iana.org/assignments/port-numbers>
 - http://en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers
- **Conceptos:**
 - El nivel de transporte [\[3.1 – intro\]](#), [\[3.1.1\]](#) y [\[3.1.2\]](#)
 - Concepto de protocolo de extremo a extremo
 - Direccionamiento de transporte [\[3.2\]](#)
 - Direccionamiento de red y necesidad de identificar los procesos
 - Puertos
 - Clasificación de los puertos:
 - Puertos bien conocidos, puertos registrados y puertos privados o dinámicos
 - Puertos en UDP
 - Puertos en TCP
 - Transporte sin conexión: UDP [\[3.3\]](#)
 - Funcionamiento
 - Necesidad
 - Estructura de los segmentos UDP [\[3.3.1\]](#)
 - Suma de comprobación [\[3.3.2\]](#)

Sesión A9

Clase 9: Fundamentos de la transferencia fiable de datos

- **Lectura previa:**
 - [Transparencias o apuntes](#)
- **Conceptos:**
 - Protocolo ARQ (*Automatic Repeat reQuest*)
 - Elementos necesarios
 - Capacidad de detección de errores por parte del receptor
 - Uso de códigos redundantes
 - Reconocimientos (ACK)
 - Retransmisión
 - Temporizador de retransmisión y tiempos de retransmisión
 - Detección de errores por el emisor
 - Vencimiento del temporizador de retransmisión
 - Generación de duplicados
 - Necesidad de números de secuencia y ACKs
 - Concepto de reconocimiento acumulativo
 - Protocolo de “parada y espera”
 - Descripción
 - Problemas de rendimiento
 - Protocolo de procesamiento en cadena” (*pipeling* o *sliding window*)
 - Concepto de ventana de transmisión
 - Descripción
 - Envío continuo
 - Comparación de rendimiento
 - Control de errores en el protocolo de “procesamiento en cadena”
 - Vuelta atrás
 - Descripción del funcionamiento
 - Repetición selectiva
 - Concepto de ventana de recepción
 - Descripción del funcionamiento
 - Transmisión bidireccional
 - *Piggybacking*
 - Control de flujo
 - Concepto
 - Aplicación de los protocolos ARQ propuestos

Sesión A10

Clase 10: Estructura de un segmento TCP. Transferencia fiable de datos en TCP

- **Lectura previa:**
 - [Kurose2010, sección 3.5](#)
- **Consulta:**
 - [RFC 793 especialmente la sección 3.1 \(Formato de la cabecera\)](#)
 - [RFC 2988: Computing TCP's Retransmission Timer](#)
- **Conceptos:**
 - Transporte orientado a conexión: TCP [\[3.5.1\]](#)
 - Concepto
 - Estructura de un segmento TCP [\[3.5.2\]](#) [\[RFC 793, sección 3.1\]](#)
 - Números de secuencia y reconocimientos
 - Reconocimientos acumulativos
 - Estimación del tiempo de ida y vuelta (RTT) y fin del temporizador [\[3.5.3\]](#) [\[RFC 2988\]](#)
 - Estimación del tiempo de ida y vuelta
 - Intervalo de fin de temporización
 - Duplicación del intervalo de fin de temporización (algoritmo de *Karn*) [\[parte de 3.5.4\]](#)
 - Transferencia fiable de datos en TCP [\[3.5.4\]](#)
 - Reconocimientos retardados
 - Reconocimientos duplicados
 - Retransmisión rápida
 - Vuelta atrás *N* o repetición selectiva
 - Control de flujo [\[3.5.5\]](#)
 - Ventana de recepción
 - Funcionamiento

Sesión A11

Clase 11: Gestión de la conexión TCP. Opciones TCP

- **Lectura previa:**
 - [Kurose2010 sección 3.5.6 excepto los estados TCP](#)
 - [RFC 879: The TCP Maximum Segment Size and Related Topics, secc. 3](#)
 - [RFC 793: TRANSMISSION CONTROL PROTOCOL, secc.3.1 \(Formato de la cabecera\)](#)
 - [RFC 2018: TCP Selective Acknowledgment Options](#)
 - [RFC 1323: TCP Extensions for High Performance, secc.2 y 3](#)
- **Consulta:**
 - [RFC 793: TRANSMISSION CONTROL PROTOCOL, secc.3.4 \(Establecimiento de una conexión\) y 3.5 \(cierre de una conexión\)](#)
- **Conceptos:**
 - Gestión de la conexión TCP [[3.5.6 excepto los estados TCP](#)] [[RFC 793, secc. 3.4 y 3.5](#)]
 - Establecimiento de la conexión
 - El bit SYN
 - Acuerdo en tres fases
 - Cierre de la conexión
 - El bit FIN
 - Acuerdo en tres o cuatro fases
 - El bit RST
 - Usos
 - Opciones TCP
 - Esquema general del campo de opciones [[RFC 793, secc. 3.1](#)]
 - Tamaño máximo de un segmento [[RFC 793, secc. 3.1](#)] [[RFC 879, secc. 3](#)]
 - Reconocimientos selectivos [[RFC 2018](#)]
 - Reconocimiento
 - Escala de ventana [[RFC 1323, secc. 2](#)]
 - Funcionamiento
 - Marca de tiempo [[RFC 1323, sección 3](#)]
 - Funcionamiento

Sesión A12

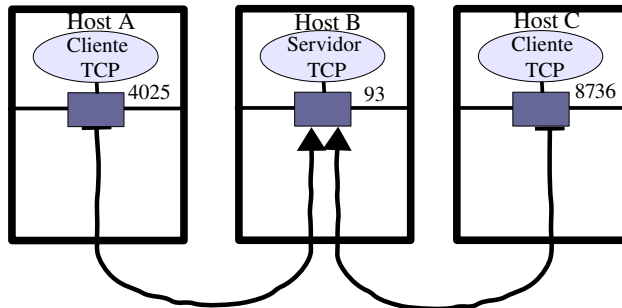
Clase 12: Control de la congestión en TCP

- **Lectura previa:**
 - [Kurose2010, sección 3.7–intro](#), excepto “*Descripción Macroscópica de ...*” y “*El Futuro de TCP*”; tampoco la sección 3.7.1
- **Consulta:**
 - [RFC 5681: TCP Congestion Control](#)
- **Conceptos:**
 - El problema de la congestión en la red
 - Mecanismo de control de la congestión en TCP [\[3.7-intro\]](#)
 - Principios de funcionamiento
 - Ventana de congestión
 - Concepto
 - Funcionamiento
 - Síntomas de congestión
 - Fin de temporización
 - Recepción de tres ACK duplicados
 - Algoritmo de control de la congestión en TCP
 - Arranque lento
 - Funcionamiento
 - Umbral de arranque lento
 - Evitación de la congestión
 - Funcionamiento
 - Recuperación rápida
 - Funcionamiento

Ejercicios

Ejercicio 1:

Supongamos que tenemos 3 hosts: A, B y C. En los hosts A y C tenemos dos clientes que usan los puertos 4025 y 8736 respectivamente. En el host B tenemos un servidor TCP que ofrece su servicio en el puerto 93 (como muestra la figura).



El cliente en A se conecta con el servidor en B y posteriormente, el cliente en C se conecta también a este servidor en B.

- ¿Cómo puede el servidor en B diferenciar las dos conexiones, si ambas son gestionadas a través de un mismo puerto (el 93)?.
- ¿Y si el cliente en C hubiese obtenido como puerto local el 4025?
- ¿Produciría algún problema añadir en el host B un servidor UDP que ofrezca su servicio en el puerto 93? Justifica la respuesta.
- ¿Qué diferencia/s presenta un servidor concurrente frente a uno iterativo desde el punto de vista de los clientes?

Ejercicio 2: Encuentra los cinco errores de la siguiente cabecera TCP, sabiendo que corresponde al primer segmento de establecimiento de una conexión realizada por un cliente estándar de HTTP hacia el correspondiente servidor estándar, y que el campo de opciones no está vacío. Razona la respuesta.

Puerto TCP origen:	120	
Puerto TCP destino:	80	
Número de secuencia:	1400	
Número de reconocimiento:	0	
Longitud de la cabecera:	5	
Reservado:	0	
Código:	URG = 0	RST = 0
	ACK = 1	SYN = 1
	PSH = 0	FIN = 0
Tamaño de la ventana (en decimal)	86535	
Checksum (en hexadecimal)	9FB0	
Opciones (en bytes hexadecimales)	02- 04 - 05 - B4 (MSS = 1460)	
Relleno:	04 - 02 (activar reconocimiento selectivo)	
	(sin relleno)	

Ejercicio 3: El control de flujo TCP, basado en ventana deslizante, dispone de una indicación de ventana (buffer disponible en el otro extremo) que limita la inyección de segmentos en la conexión. El tamaño máximo que se puede indicar es de 64 KB. Esta limitación, ¿podría afectar a las prestaciones de TCP cuando se utilizan redes de alta velocidad (Ej.: Gigabit Ethernet ~1Gbps) con RTTs del orden de 2 ms.?

Ejercicio 4: Supongamos que un proceso P (en el host A1) quiere establecer una conexión TCP con el proceso Q (en el host E1):

a) Suponiendo que no se pierden datagramas ni se producen retransmisiones, indique los segmentos que se intercambiarían ambos TCP's en las siguientes situaciones:

1. Q realiza una apertura pasiva, y P una apertura activa (suponer que $NSI(P)=0$, $WIN(P)=20$, $NSI(Q)=13500$, $WIN(Q)=30$).
2. A continuación P envía 100 octetos a su entidad TCP para que los envíe a Q.
3. Después el proceso P le indica a su TCP el cierre de conexión. Supongamos que Q también está dispuesto a cerrar cuando le llegue el aviso de cierre de P.

b) Lo mismo que el apartado a), pero suponiendo que Internet pierde un segmento de cada 3 que envíe el TCP del proceso P. Los segmentos que envía el TCP del proceso Q no se pierden. El temporizador de retransmisión (RTO) de P tiene un valor igual a dos RTT's.

NOTA: NSI (P) representa el número de secuencia inicial del proceso P, mientras que WIN(P) refleja el tamaño de ventana inicial que escoge el TCP del proceso P. El MSS=100 bytes. La respuesta ha de reflejarse en una tabla con formato:

Origen/ Nº secuencia / Flags / Nº reconocimiento / Datos (indicando byte inicial y final)

Ejercicio 5: Un emisor ha enviado los segmentos 1 al 50. Cada uno de ellos con 512 bytes de datos. El emisor recibe un ACK con valor 15873 ($31 \times 512 = 15872$), y después 3 ACKs duplicados con valor 15873.

a) Basándose en esta información, ¿qué segmento(s) puede suponer el emisor que se han perdido? ¿Y cuáles puede considerar que se han recibido correctamente?

b) El mismo emisor reenvía el segmento supuestamente perdido y recibe como respuesta un ACK con valor 18433 ($36 \times 512 = 18432$). De los 50 segmentos enviados inicialmente, ¿puede suponer el emisor que se ha perdido alguno?. En caso afirmativo, ¿cuál o cuáles?. ¿Qué segmentos puede considerar como recibidos correctamente?

Ejercicio 6: El host B recibe peticiones de conexión TCP desde un host remoto llamado "A". Realiza, para cada uno de los escenarios siguientes, un esquema gráfico detallando qué segmentos se intercambiarían después de la llegada de los segmentos que se indican. No hay que detallar ningún segmento de la posible fase de transferencia de datos.

a) Llega una petición de conexión desde "A", con número de secuencia 34, a un puerto donde no hay ningún servidor escuchando.

b) Llega una petición de conexión desde "A", con número de secuencia 34, a un puerto donde hay un servidor escuchando. Antes de que se haya completado el establecimiento de la conexión en "B" llega, desde "A", una retransmisión de la petición anterior.

c) "B" envía una petición de conexión a "A", con número de secuencia 75, a un puerto donde hay un servidor escuchando. Instantes después, "B" recibe una petición de conexión desde "A", con número de secuencia 34, a un puerto donde también hay un servidor escuchando.

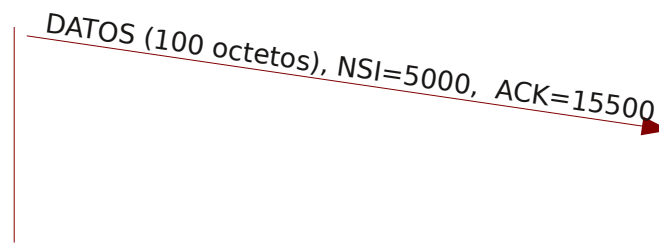
d) Llega una petición de conexión desde "A", con número de secuencia 34. Tras enviar "A" la petición de conexión, el computador A es reiniciado. Cuando "B" contesta esta petición de conexión, el computador A vuelve a estar totalmente operativo, ¿cuál sería su respuesta?

Ejercicio 7: Un usuario quiere conectarse al servidor web de un computador que en realidad no tiene ningún servidor web en ejecución, por lo que la conexión no se podrá establecer. Llamaremos "computador A" al host en el que el usuario está ejecutando el navegador, y "computador B" al host en el que presuntamente está en marcha el servidor web. Completa la siguiente tabla, especificando

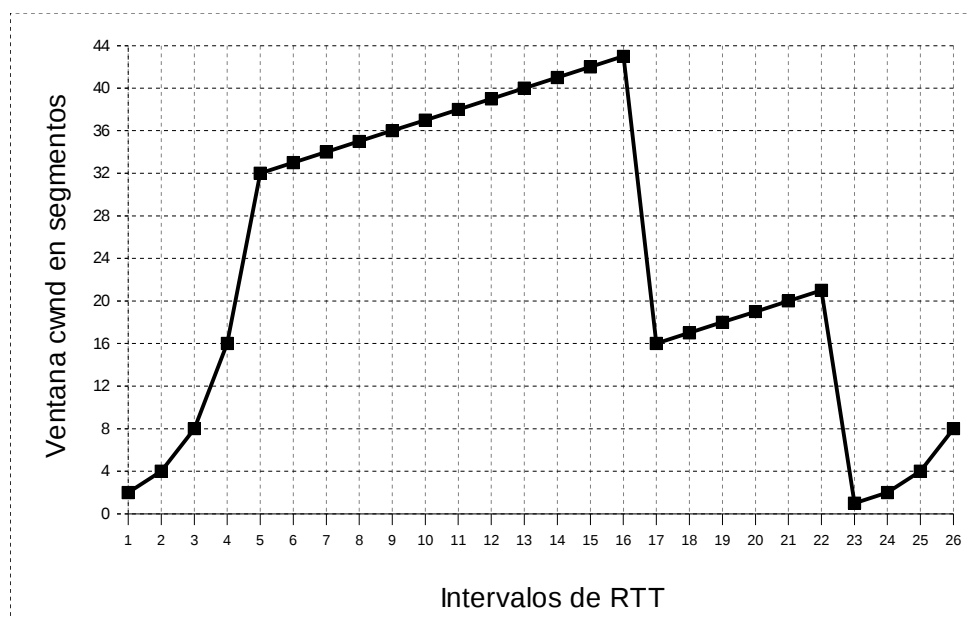
la secuencia de segmentos que tendrá lugar entre los dos computadores en el supuesto anterior.

Origen (computador)	Nº secuencia	Flags	Nº ACK	Datos
A	51	SYN		
B				

Ejercicio 8: Suponiendo que no se producen errores y que ninguno de los dos extremos envía ya más datos, completa la siguiente secuencia de segmentos TCP hasta que la conexión quede cerrada en los dos extremos. Indica para cada segmento: *flags* activados, valor del campo ACK (sólo si el flag ACK está activado) y número de secuencia del segmento.



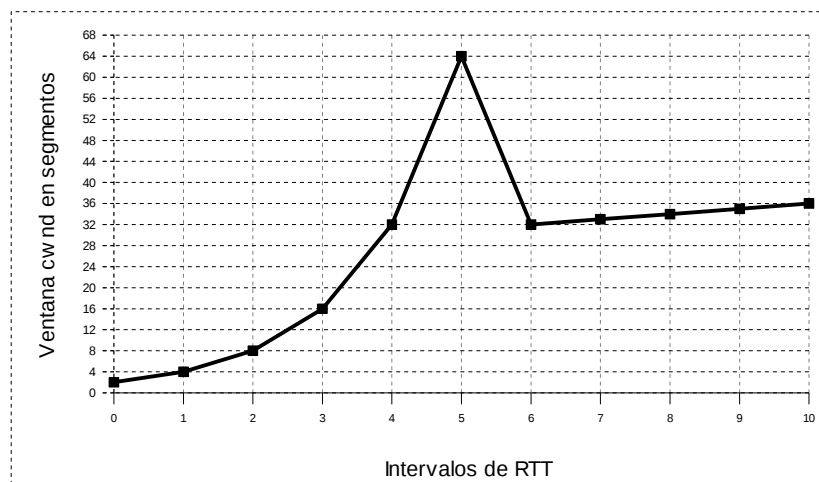
Ejercicio 9: Considera la siguiente representación del tamaño de la ventana de congestión en función del tiempo. Para el control de la congestión de TCP, responde a las siguientes cuestiones: (Cuando se pregunte tamaño de ventanas, indícalo en segmentos).



- Identifica los intervalos en los que actúa el mecanismo arranque lento (slow-start).
- Identifica los intervalos en los que actúa el mecanismo de evitación de la congestión (congestion avoidance).
- A continuación del RTT 16, ¿cómo se detecta la pérdida del segmento? ¿Y a continuación del RTT 22?
- ¿Cuál es el valor del umbral (slow-start threshold) en el RTT 2?
- ¿Cuál es el valor del umbral (slow-start threshold) en el RTT 18?
- ¿Cuál es el valor del umbral (slow-start threshold) en el RTT 24?
- ¿Durante que RTT se envía el segmento 70?
- Suponiendo que tras el RTT 26 se detecta la pérdida de un paquete por la recepción de tres

ACKs duplicados, ¿cuáles serán los valores de la ventana de congestión y del umbral?

Ejercicio 10: La gráfica siguiente representa la evolución de una ventana de congestión TCP en función del tiempo. La ventana de permisos del receptor (WIN) es siempre igual a 100 segmentos (100*MSS bytes). Responde a las siguientes preguntas:



- Indica qué mecanismos están actuando y durante qué intervalos.
- ¿Qué ocurre durante el RTT 5? ¿Cómo se detecta?
- ¿Cuál es el valor del umbral durante el RTT 3? ¿Y durante el RTT 8?
- Si durante el RTT 10 vence un temporizador, ¿cuáles serán los valores de la ventana de congestión y del umbral en el RTT 11?

Ejercicio 11: El proceso pA en el computador A y el proceso pB en el computador B se disponen a establecer una conexión TCP. A lo largo de toda la conexión, los tamaños de la ventana que pA y pB declararán en sus segmentos permanecerán constantes con los valores siguientes: $WIN(pA) = WIN(pB) = 300$ bytes. Suponiendo que no se pierden paquetes ni se producen retransmisiones, describe el intercambio de segmentos entre pA y pB en las situaciones siguientes (véase la nota al final del enunciado con el formato a seguir):

- Establecimiento de conexión entre pA y pB (pA realiza una apertura activa y pB una apertura pasiva). Suponed que durante este proceso se intercambian los siguientes números de secuencia iniciales: $NSI(pA) = 1000$ y $NSI(pB) = 5000$.
- A continuación, pA envía 700 bytes a pB. Para este apartado supondremos, que durante la fase de establecimiento de la conexión, ambos procesos han acordado intercambiar segmentos de tamaño máximo 100 bytes ($MSS = 100$ bytes). Siempre que sea posible pA envía segmentos del tamaño MSS. Se aplica una política de reconocimientos retrasados. La transferencia de información debe seguir el protocolo de arranque lento (slow-start). No se produce la pérdida ni el cambio de orden de ningún segmento.

NOTA: La respuesta ha de reflejarse en una tabla con formato:

Origin/ N° secuencia / Flags / N° reconocimiento / Datos (indicando byte inicial y final)

Ejercicio 12: Un cliente y un servidor se comunican mediante el protocolo TCP. La aplicación cliente envía una petición de 30 bytes al servidor. La respuesta del servidor es un mensaje de 1476 bytes, tras el que iniciará el cierre de la conexión. Sabemos que el MSS que emplean los dos extremos es de 512 bytes, $NSI(C) = 7.000$, $NSI(S) = 15.000$ (NSI es el número de secuencia inicial), $WIN(C) = WIN(S) = 2048$ y que, para este ejercicio, el tamaño inicial de la ventana de congestión es dos segmentos ($2 * MSS$ bytes). Ambos extremos emplean reconocimientos retrasados. Describe la evolución de la conexión TCP, desde el establecimiento hasta el cierre de la conexión. La respuesta ha de reflejarse en una tabla con formato:

Origen/ N° secuencia / Flags / N° reconocimiento / Datos (indicando byte inicial y final)

Ejercicio 13: El computador A desea solicitar un servicio al computador B. Para ello abre una conexión TCP y transmite una petición de 100 bytes. Una vez recibida, B transmite una respuesta de 1000 bytes y cierra la conexión. Describe el intercambio total de segmentos entre A y B, teniendo en cuenta que el tamaño máximo de segmento que ambos establecen es de 200 bytes, las ventanas de recepción de A (WINA) y de B (WINB) son iguales y de 500 bytes. Ambos extremos emplean reconocimientos retardados, y el tiempo de transmisión se considera despreciable frente al RTT. El número de secuencia inicial de A es 1500, y el de B 5000. Los mecanismos de control de congestión actúan de la forma habitual. La respuesta ha de reflejarse en una tabla con formato:

Origen/ N° secuencia / Flags / N° reconocimiento / Datos (indicando byte inicial y final)

Ejercicio 14: Suponiendo que el emisor tiene siempre datos pendientes de transmitirse, escribe en forma de la tabla la evolución durante los 10 primeros RTT's de la **ventana de transmisión** del emisor (en segmentos) en función del tiempo (medido en RTTs), para una conexión TCP que se inicia en RTT=0 con la ventana de congestión igual a 2 segmentos y la ventana de permisos del receptor permanentemente igual a 64 segmentos. Asimismo, indica cómo se modifica esta ventana si en RTT=10 vence un temporizador.