UNIVERSIDAD AUTÓNOMA DE MADRID	Escuela Politécnica Superior Redes de comunicaciones 1	**Examen 2º parte** 8 enero 2013 Modelo A/B

APELLIDOS (MAYÚSCULAS)_____

NOMBRE (MAYÚSCULAS):_____

DNI:_____ **Nº PAGs:**_____ **POSICIÓN:**_____

GRUPO(mañana(m)/tarde(t)/doble(d)):__ **FIRMA:**_____

Tiempo: Dos horas y media. Sin libros ni apuntes, pueden usarse calculadoras.
El examen se compone de 2 preguntas de desarrollo (PD), 6 cuestiones (C) cortas y 3 problemas (P). La puntación de cada una de ellas se muestra en el enunciado.
Para que el examen haga media con la primera parte de la asignatura es necesario sacar más de cuatro puntos de calificación, la entrega de este examen supone la aceptación expresa de estas normas de evaluación. El alumno al entregar el examen debe firmar la hoja de asistencia para que el examen sea corregido.
La fecha estimada de la publicación de notas es el 14 enero 2013 (se publicaran al menos en el tablón del curso y probablemente en Moodle).
La revisión será el 17 enero 2013. Aula y hora por confirmar.

PD1 (1,25 p) Accesibilidad desde Internet a equipos tras NAT.
Transparencias 4-78 y siguientes junto a páginas 341 y 342 del libro:

- Estáticamente
- Tablas de traducciones dinámicas
- Retransmisores/skype
- NAT transversal
- Explicación

PD2 (1,25 p) UDP. La confiabilidad y la ordenación en la transmisión de datos así como los controles de flujo y congestión son servicios típicamente facilitados por el nivel de transporte. Sin embargo, un desarrollador de *software* de redes puede trasladar estas funcionalidades a un nivel superior y utilizar […]
[com: No se pide un resumen de UDP]
Trasparencias 3-15 y 3-72.
Trasladar funcionalidades como la confiabilidad y ordenación no tiene una utilidad evidente pues su implementación de nivel 4 (por ejemplo TCP) está optimizada en los sistemas operativos actuales y su re-implementación (además a nivel más alto, el en caso de Linux en la zona de usuario) sería presumiblemente peores. La reimplantación del control de flujo tampoco conlleva ventajas, un nivel de comunicación con confiabilidad pero sin control de flujo, básicamente consistiría en enviar paquetes que el receptor nunca capturaría y que mediante algún mecanismo (como los números de secuencia TCP) serían detectados como perdidas y constantemente reenviados. Esto es, malgastar el ancho de banda sin contrapartida positiva.
En cualquier caso las cabeceras no serían de tamaño significativamente menor pues se facilitan servicios equivalentes.
Por el contrario un desarrollar puede implementar un algoritmo de control de congestión que le beneficie con respecto al resto de usuarios ante un canal compartido de comunicación asumiendo que la mayoría de ellos usan un protocolo que busca la equidad (TCP, por ejemplo, mediante sus algoritmos de control de congestión favorece la equidad en el uso del ancho de banda). Un usuario de esta aplicación puede evitar reducir su tasa de transmisión cuando se detecten pérdidas o retardos grandes, esperando que sean las demás conexiones quienes lo hagan y acaparando más ancho de banda del que le correspondería. Este protocolo se ha

denominado a veces como UDP caníbal, como el nivel de aplicación y nivel 4 se ejecutan en los extremos y se basa en UDP no hay problemas de incompatibilidad o de violación de estándares.
Sin embargo, el inconveniente más evidente, que si conlleva este planteamiento, es que si todos los usuarios de la red implementaran este protocolo la red se saturaría y podría llegar a reducir la tasa efectiva de los usuarios de la aplicación modificada. Problemas menores serían la optimización de esta implementación, y el filtrado que UDP a menudo sufre.

C1 (0,5 p) Explique a que hace referencia el término patata caliente (*hot potato*) cuando se usa como política de enrutado. Dispone de un máximo de 5 líneas.
Transparencia 4-130. El enrutamiento por patata caliente consiste en dar prioridad a rutas inter-AS según minimicen el coste intra-AS.

C2 (0,5 p) Explique por qué tanto los algoritmos de enrutado intra-AS e inter-AS se ejecutan dentro de cada router de cada sistema autónomo. Dispone de un máximo de 5 líneas.
Sección 4.6 del libro y transparencias 4-126,-127,-128,-129. La clave está en decir que los router interiores de un AS deben saber llegar a direcciones de otros AS así como a direcciones dentro del propio AS.

Conteste verdadero o falso, y explique el porqué en un máximo de 3 líneas:
C3b/a Cuando una aplicación recibe un bloque de datos de UDP, la aplicación sabe que los datos fueron enviados como un *mensaje* por el emisor, puede asumir baja carga del sistema.
Verdadero. *[com: Aunque puede aceptarse Falso como respuesta en caso de estar bien justificada como se explica a continuación]*.
UDP transmite los datos en un segmento/datagrama UDP cada vez que una aplicación escribe en el socket correspondiente. Al no haber control de congestión o flujo no se espera a que el receptor esté listo o a las limitaciones de tasa de transmisión debido a problemas de congestión.
Puede darse el caso que un sistema muy cargado no pueda atender a las peticiones del módulo UDP por cada bloque de datos UDP que se envía/recibe, sin embargo se asume baja carga en el sistema.
Sin embargo, es cierto que un socket UDP no demultiplexa comprobando la IP origen o el número de puerto origen, de modo que dos transmisiones de distintos equipos podría llegar escribir en memoria de recepción antes de una lectura, de modo que se agregaran mensajes. Dada esta explicación, o alguna limitación equivalente, se ha aceptado como respuesta válida Falso.

C3a/b Cuando una aplicación recibe un bloque de datos de TCP, la aplicación sabe que los datos fueron enviados como un *mensaje* por el emisor.
Falso. TCP sigue distintas políticas que le permiten agregar más de un mensaje en un segmento o segmentar un mensaje en varios segmentos. Por ejemplo, el control de flujo puede hacer que (debido a saturación en el receptor) un mensaje no se transmita inmediatamente después de que ser generado por una aplicación, sino que TCP lo envía cuando el extremo receptor tenga recursos para recibirlo.

C4 (0,75 p)
Paquete 1/3: bytes 20 y 21 (20 00 HEX), valor 0.
Paquete 2/4: bytes 20 y 21 (00 B9 HEX), valor 1480 (185*8).

C5 (0,5 p)
Paquete 1 y 3: bytes 20 (20 HEX) →001: Esto indica que se permite la fragmentación y que hay más fragmentos, lo cual es coherente en cuanto el siguiente paquete comparte el campo identificador y es un fragmento.
Paquete 2 y 4: bytes 21 (00 HEX) →000: Esto indica que se permite la fragmentación, lo cual es coherente pues el campo desplazamiento no fue cero, y se nos indica que no hay más lo cual es posible dado que, simplemente, no se muestran más paquetes en la captura.

C6 (0,75 p)

[com: esta pregunta tenía como dificultad darse cuenta de que la cabecera UDP está en los paquetes 1 y 3, ya que hubo fragmentación IP, lo cual parecía razonable dado que las cuestiones C4 y C5 preguntaban justo sobre ello]
Paquete 1 (modelo A) → 829A HEX equivalente a 33434 (bytes 36 y 37 del paquete 1)
Paquete 3 (modelo B) → 829B HEX equivalente a 33435 (bytes 36 y 37 del paquete 3)

P1 Rangos direcciones IP y tablas de reenvío.
[*com: mismo que ejercicio p13 del libro desplazando bits a la derecha para que sea más fácil (rellenando aleatoriamente por la izquierda)]*
Puede haber otras soluciones pero esta es en mi opinión la más intuitiva (Modelo A):

Rango direcciones destino nominales	Interface
128.129.224.0/26	0
128.129.224.0/23	1
64.64.15.1/32	2
128.129.224.64/32 128.129.225.128/25 0.0.0.0/0	3

(Modelo B):

Rango direcciones destino nominales	Interface
129.128.224.0/26	0
129.128.224.0/23	1
65.64.15.1/32	2
129.128.224.64/32 129.128.225.128/25 0.0.0.0/0	3

P2 Direccionamiento IP/ETH. Ejercicio considerado de mínimos.
[com: ejercicio ya realizado en junio pero en el otro sentido y remarcado en clase como de conocimiento mínimo]

Puede haber otras soluciones válidas en cuanto a los puertos:

Sonda	Dir. MAC Origen	Dir. MAC Destino	Dirección IP origen	Dirección IP destino	Puerto origen	Puerto destino
4	ETH8	ETH7	W	150.8.8.126	80	10001
3	ETH6	ETH5	W	192.168.1.18	80	10000
2	ETH4	ETH3	W	192.168.1.18	80	10000
1	ETH2	ETH1	W	192.168.1.18	80	10000

Modelo B:

Sonda	Dir. MAC Origen	Dir. MAC Destino	Dirección IP origen	Dirección IP destino	Puerto origen	Puerto destino
4	ETH8	ETH7	W	150.8.8.126	80	10001
3	ETH6	ETH5	W	192.168.1.20	80	10000
2	ETH4	ETH3	W	192.168.1.20	80	10000
1	ETH2	ETH1	W	192.168.1.20	80	10000

P3 TCP y ventana de congestión.

P3.a

$TH_{medio} \approx W_{medio} * TS / RTT$

$TH_{medio} \approx 45 * 10000 * 8 / 75 / 10^{-3}$

$TH_{medio} \approx 48$ Mb/s

P3.b

$Tiempo_{30000 \rightarrow 60000} = 30000*75*10^{-3} = 2250$ s > 37 minutos

P3.c

[com: Por descontado que TCP Tahoe funcionaría aún peor, pero se pregunta por posibles ideas para mitigar el problema.]

TCP Reno no es de ningún modo adecuado en la red b. De forma ideal deberíamos poder transmitir en media cerca de la tasa máxima, pero, en la red b, cuando se detecta 3 ACKs duplicados y se divide la ventana, esta queda demasiado pequeña del tamaño máximo como se puede ver empíricamente en la figura, lo que causa una cantidad de tiempo excesiva para recuperar una tasa de envio razonable.

Si calculamos el tiempo para alcanzar el límite físico tras 3 ACKs en a tenemos que:

$Tiempo_{30 \rightarrow 60} = 30*75*10^{-3} = 2'250$s De modo que el mismo razonamiento no aplica tan claramente en a, pudiendo ser en este caso TCP Reno aceptable.

Para mitigar este problema se pueden aplicar básicamente dos ideas: reducir de manera menos significativa la ventana tras 3 ACKs duplicados, o hacer que la ventana crezca más rápido. Cualquiera de ellas o ambas son razonables.

Como ejemplos, se podría proponer reducir la ventana en un ¼ o decrementar de forma lineal el tamaño de ventana. Similarmente, se podría usar arranque lento entre un valor asignado a la mitad de la ventana tras 3 ACKs duplicados y un nuevo valor umbral más grande.