

**TELEMATICA. 18 de Junio de 2003.**

**Dpto. de Comunicaciones. E.T.S.I. de Telecomunicación de Valencia.**

- Duración del examen: 2 horas.
- Utilice sólo el espacio destinado a respuestas.
- Las respuestas deben realizarse con bolígrafo o pluma.
- Se invalidarán las respuestas que no cumplan los requisitos indicados.

**APELLIDOS:**

**NOMBRE:**

1.- Razone la siguiente afirmación: “El control de la congestión es más sencillo en las redes con estructura de circuito virtual que con datagrama”

Con la estructura de cto. virtual, los nodos de la red saben de antemano que va a haber un flujo de paquetes por cada cto. virtual, se realizan reservas de recursos (memoria) adecuadas, con lo que resultará más difícil que los picos de tráfico puedan desbordar y congestionar los nodos.

Además, y en general, pueden complementariamente implementarse otras medidas de control de congestión de tipo preventivo:

- se puede organizar los ctos. virtuales en función de la disponibilidad de los recursos que se necesiten o soliciten,
- se puede rechazar nuevas conexiones si no hay recursos suficientes o si con ellas se puede perturbar la calidad de servicio de las ya existentes,
- se puede implementar mecanismos de control de flujo salto a salto (como en X.25),
- se puede implementar mecanismos de adecuación o conformación de tráfico (como sucede con los parámetros de servicio en Frame Relay),

(1 punto)

2.- ¿Existe algún algoritmo de encaminamiento que garantice que en una red con estructura interna tipo datagrama, todos los paquetes correspondientes a una misma comunicación entre un host origen y un host destino van a atravesar la misma ruta y que, por lo tanto, van a llegar en orden? Razone la respuesta.

El encaminamiento estático (por esa característica estática) hace que todos los paquetes intercambiados entre dos host atraviesen la red por la misma ruta aunque se trate de una estructura datagrama, por lo que salvo por mal funcionamiento, en lo que respecta al encaminamiento la entrega de paquetes será ordenada.

El encaminamiento por inundación también da un resultado similar, lo que sucede es que con este mecanismo los paquetes (y sus copias) van a llegar al destino atravesando todas las rutas posibles y no por una sola, pero entre todas las posibles estará la más corta, por lo que todos los paquetes irán por esa ruta más corta y llegaran en orden.

(1 punto)

3.- Razone si la siguiente afirmación es verdadera o falsa:

Supongamos que en un momento dado el RTT medido en una conexión TCP es igual a T, entonces el timeout calculado para esa conexión tiene que ser necesariamente mayor o igual que T.

Es falsa.

El cálculo del Timeout depende del RTT medido, pero también depende de valores anteriores y de parámetros tal y como se expresa a continuación:

$RTT_{nuevo} = \alpha RTT_{antiguo} + (1 - \alpha) RTT_{muestreado}$

$RTT_{antiguo} = RTT_{nuevo}$

$Timeout = RTT_{nuevo} + 4 * Desviación_{RTT}$

$Desviación_{RTT} = \beta Desviación_{RTT} + (1 - \beta) (RTT_{nuevo} - RTT_{muestreado})$

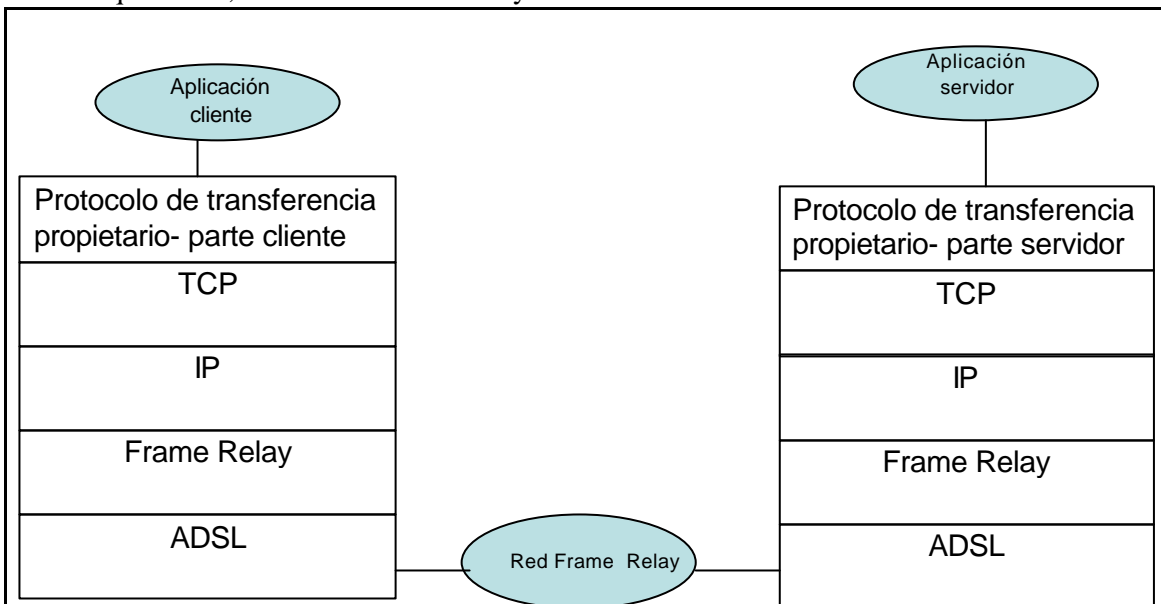
Sería suficiente con que el parámetro  $\alpha$  fuese alto (próximo a la unidad) y que la desviación hasta el momento dado hubiese sido pequeña, para que un RTT medido tuviese poca incidencia en las expresiones, y por tanto el Timeout calculado fuese inferior a ese RTT medido.

(1,5 puntos)

4.- Por encargo de una entidad bancaria, se quiere diseñar un sistema de transferencia de ficheros, incluyendo un protocolo de aplicación cliente/servidor no estándar, en el que sea requisito prioritario la capacidad para transferir grandes volúmenes de datos en poco tiempo, con total fiabilidad.

Defina la arquitectura completa de protocolos extremo a extremo (de aplicación a aplicación), pudiendo utilizar algunos de los siguientes protocolos u otros que usted considere convenientes utilizar o diseñar: X.25, Frame Relay, LAP-B, PPP, LLC, X.21, RDSI, ADSL, HTTP, FTP, IP, UDP y TCP.

En cualquier caso, razones sus elecciones y su diseño.



Razonamiento:

- aplicación no estándar cliente/servidor (no vale FTP ni HTTP)
- total seguridad  $\approx$  TCP
- gran capacidad  $\approx$  IP encapsulado sobre Frame Relay + ADSL

(2 puntos)

5.- Un datagrama (IPv4 sin opciones) de 3000 bytes tiene que atravesar una red cuya MTU (Maximun Transmission Unit o tamaño máximo del paquete en transmisión) es 500 bytes.

- a) ¿Cuántos fragmentos se generan?
- b) ¿Qué información, relativa a la fragmentación, contiene la cabecera de los paquetes IPv4?
- c) ¿Cuál es el valor de estos campos en cada uno de los fragmentos?
- d) Indica las diferencias, ventajas y desventajas de las dos modalidades de fragmentación estudiadas.

a) Datagrama = Paquete completo, supone que un datagrama de 3000 bytes contiene 20 bytes de cabecera (4 palabras de 32 bits x 5 palabras). Por tanto 2980 bytes de información.

Al atravesar la nueva red, la MTU es de 500 bytes, de los cuáles 20 corresponden a la cabecera, esto supone que cada datagrama únicamente puede transportar 480 bytes de información.

Si dividimos 2980 entre 480 obtenemos un valor de 6'21, lo que supone 7 fragmentos: seis con una longitud de 500 bytes (480 de información) y uno de 120 (100 de información).

b) La información relativa a la fragmentación se almacena en diferentes campos de la cabecera:

IDENT: Identifica a los fragmentos que corresponden a un mismo paquete original

FRAGMENT OFFSET: Indica la posición del primer octeto del campo de datos del paquete original. El primer fragmento tiene un valor cero para este campo.

Flag DF: Indica que el paquete original no debe fragmentarse.

Frag MF: Indica si hay más fragmentos (1) o no los hay (0).

c) En el caso que nos ocupa, todos los fragmentos tendrán valor IDENT = i

FRAGMENT OFFSET= 0, 480, 960, 1440, 1920, 2400, 2880

El bit MF = 1 para los seis primeros paquetes y 0 para el séptimo

d) La principal diferencia es que elemento realiza el reensamblado de los fragmentos una vez se ha realizado la misma. En la fragmentación transparente el router de salida de la red en la que se ha tenido que realizar la fragmentación y en el caso de fragmentación no transparente el encargado de realizar el reensamblado es el host destino.

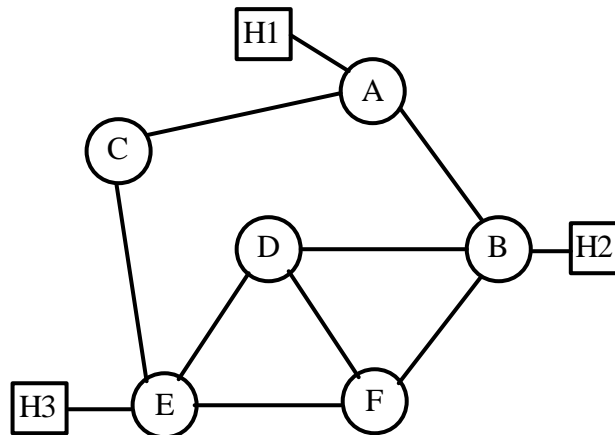
Ventajas fragmentación transparente: el host destino no se da cuenta de que se ha realizado una (o más) fragmentaciones en el trayecto del datagrama, mientras que en la fragmentación no transparente es el host destino es el que debe realizar el reensamblado

Ventajas fragmentación no transparente: una vez realizada la fragmentación, cada fragmento es tratado de forma independiente, lo que supone que no deben salir todos los fragmentos por el mismo router de salida, como sucede en la fragmentación transparente, para que se pueda realizar el reensamblado.

Existe una desventaja común a los dos tipos de fragmentación, consistente en que cuando se realiza el proceso de fragmentación y se transmiten los paquetes, si se pierde un fragmento, todo el paquete es desechado.

(2 puntos)

6.- Sea la red tipo datagrama de la figura, donde trabaja un protocolo de encaminamiento adaptativo distribuido, cuyo criterio de decisión de encaminamiento se basa en la ruta con menor número de saltos (igual al estudiado en clase). A este mecanismo se le ha añadido un segundo criterio basado en la carga de los nodos (tráfico cursado en paquetes/segundo), de forma que en caso de igualdad en 2 o más rutas (tras aplicar el criterio de camino más corto) se considerará como mejor ruta la del nodo menos cargado. Para implementar este algoritmo, las tablas de encaminamiento necesitan de una columna adicional, donde se contabiliza el número de paquetes/segundo que encamina hacia cada destino. La carga de un nodo será la suma de todos los valores de esa columna. Así cuando un nodo informa a sus vecinos de las distancias estimadas a los distintos destinos también les comunica su carga.



Para este ejercicio sólo vamos a considerar como destinos posibles los hosts del dibujo (olvidándonos de otros). Así mismo sólo vamos a considerar el tráfico generado en cada uno de los hosts como el único presente en la red y el único que se contabilizará; tráfico que supondremos constante y que los nodos son capaces de detectar en cuanto por encaminamiento tengan que cursarlo.

Llamaremos  $\tau_{ij}$  al número de paquetes por segundo generado en el host  $i$  con destino el host  $j$ . Consideraremos  $\tau_{ij}=?$  para todos los nodos.

a) Complete las tablas de encaminamiento de los nodos. Los datos a completar son las distancias y los tráficos cursados en cada caso, y para ello utilice la información que ya está consignada en las tablas.

Nodo A

Dest.	Salida	Dista.	Tráf.
H1	H1	1	2?
H2	B	2	?
H3	C	3	?
Carga			4?

Nodo B

Dest.	Salida	Dista.	Tráf.
H1	A	2	?
H2	H2	1	2?
H3	D	3	?
Carga			4?

Nodo C

Dest.	Salida	Dista.	Tráf.
H1	A	2	?
H2	A	3	0
H3	E	2	?
Carga			2?

Nodo D

Dest.	Salida	Dista.	Tráf.
H1	E	4	0
H2	B	2	?
H3	E	2	?
Carga			2?

Nodo E

Dest.	Salida	Dista.	Tráf.
H1	C	3	?
H2	D	3	?
H3	H3	1	2?
Carga			4?

Nodo F

Dest.	Salida	Dista.	Tráf.
H1	B	3	0
H2	B	2	0
H3	E	2	0
Carga			0

- b) Teniendo en cuenta los criterios de encaminamiento señalados en el enunciado, indique que decisiones de encaminamiento de las tablas son incorrectas, por qué y cuál debería ser en cada caso la decisión de encaminamiento correcta.

Hay errores en:

- la tabla del nodo D con respecto al camino más corto: para encaminar al H1 considera la ruta a través de E, con lo que el número de saltos son 4, que es más larga que a través de B que daría un número de saltos de 3.
- la tabla del nodo E o el nodo B con respecto al criterio de ruta menos cargada: en ambos nodos se encamina todo el tráfico entre los H2 y H3 a través del nodo D, cuando hay una ruta entre ambos nodos a través de F con igual número de saltos estando este nodo totalmente descargado. Lo correcto sería que bien el nodo E encaminase hacia el H2 a través de F y B quedase igual, o bien que B encaminase hacia H3 a través de F y E quedase igual. De esta forma la carga de D quedaría igual a la de F(=?).

(2,5 puntos)