



Universidad de Granada
Departamento de Teoría de la Señal,
Telenática y Comunicaciones



FUNDAMENTOS DE REDES

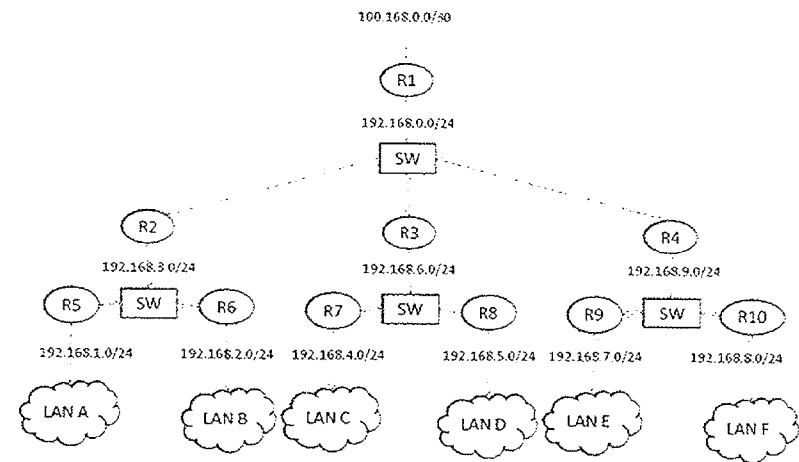
- 3er. curso del Grado de Ingeniería Informática -
Examen de teoría - Septiembre 2016

Apellidos y nombre: PROFESOR Grupo: _____

1. (1 pto.: 10x0,1) Marque como verdaderas (V) o falsas (F) las siguientes afirmaciones:
(Nota: una respuesta errónea anula una correcta)

	V	F
a) Un protocolo requiere, para su definición completa, una sintaxis y una semántica.	<input type="checkbox"/>	<input checked="" type="checkbox"/>
b) HTTP es un protocolo sin estado.	<input checked="" type="checkbox"/>	<input type="checkbox"/>
c) El término "best effort" significa sin garantía de calidad de servicio.	<input checked="" type="checkbox"/>	<input type="checkbox"/>
d) El protocolo de transporte UDP permite la entrega ordenada.	<input type="checkbox"/>	<input checked="" type="checkbox"/>
e) El protocolo de transporte TCP utiliza la técnica piggybacking.	<input checked="" type="checkbox"/>	<input type="checkbox"/>
f) La conmutación de circuitos es siempre más veloz que la de datagramas.	<input type="checkbox"/>	<input checked="" type="checkbox"/>
g) Los algoritmos de encaminamiento aislados producen rutas subóptimas.	<input checked="" type="checkbox"/>	<input type="checkbox"/>
h) En OSI y TCP/IP, la capa de enlace es proveedora de servicio de la capa de red.	<input checked="" type="checkbox"/>	<input type="checkbox"/>
i) La integridad específica que la información no haya sido accedida por un intruso.	<input type="checkbox"/>	<input checked="" type="checkbox"/>
j) El direccionamiento basado en clases no utiliza la máscara de red.	<input checked="" type="checkbox"/>	<input type="checkbox"/>

2. (1 pto.: 2x0,5) 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 $\alpha=0,125$ y $\beta=0,25$.
 - ¿Y si se utiliza el algoritmo de Karn?
3. (1.5 pto.: 2x0,75) Un mensaje de 100 kB se transmite a lo largo de tres saltos de una red. Ésta limita la longitud máxima de los paquetes a 1024 B incluyendo una cabecera de 32 B. Las líneas de transmisión de la red no presentan errores y tienen una capacidad de 100 Mbps. Cada salto corresponde a una distancia de 100 m.
- ¿Qué tiempo se emplea en la transmisión del mensaje mediante datagramas?
 - ¿Y si se realiza un control de flujo con una ventana de 5 datagramas? ¿qué ventaja aporta esta solución?
4. (1.5 pto.: 1+0,5) Imagine la siguiente topología y asignación de direcciones IP a las subredes.



2.-

a) $RTT_0 = 30ms$, $D_0 = 2ms$

$$RTT_1 = (1-\alpha) \cdot RTT_0 + \alpha \cdot 26 = 29,5$$

$$D_1 = (1-\beta) \cdot D_0 + \beta |RTT_1 - 26| = 2,375$$

$$RTT_2 = 29,813$$

$$D_2 = 2,328$$

$$RTT_3 = 30,086$$

$$D_3 = 2,225$$

$$RTT_4 = 29,325$$

$$D_4 = 3$$

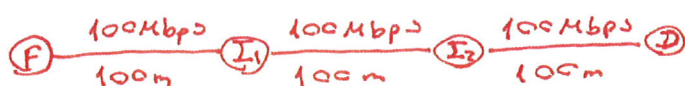
$$T_{out} = RTT_4 + 4 \cdot D_4 = 41,324 ms$$

b) En este caso, los últimos 2 acuses no se consideran por ser ambiguos, por tanto

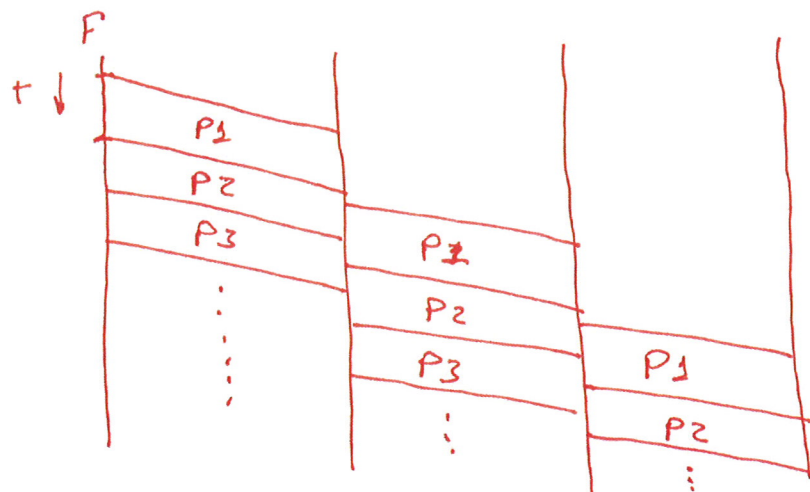
$$T_{out} = RTT_2 + 4 \cdot D_2 = 39,125ms$$

3.-

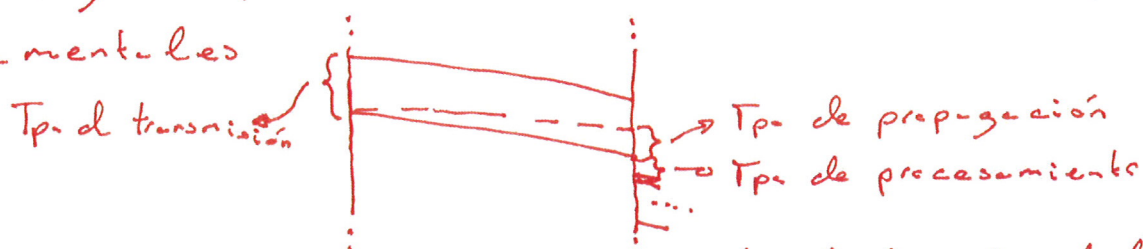
La red especificada sería la siguiente:



a) El diagrama de tiempos sería el siguiente:



Desde la principal característica, al ser envío por datagramas, es que no hay inicio ni cierre de conexión. Para el envío de un datagrama por cada enlace, encontramos tres retardos fundamentales



El tiempo de transmisión depende de la velocidad de la línea y el tamaño del datagrama, por lo que es igual en todos los saltos:

$$t_t = \frac{P}{V_t} = \frac{1024 \times 8 \text{ b}}{100 \text{ Mbps}} = 81,92 \mu\text{s}$$

El tiempo de propagación depende de la longitud de la línea y de la velocidad de propagación ($2 \cdot 10^8 \text{ m/s}$), por lo que también es igual en todos los saltos:

$$t_p = \frac{D}{V_p} = \frac{100 \text{ m}}{2 \cdot 10^8 \text{ m/s}} = 0,5 \mu\text{s}$$

El tiempo de procesamiento se suele despreciar, considerando la velocidad de los procesadores actuales.

Para calcular el tiempo que se nos pide, nos falta el n° de datagramas a enviar:

$$Nd = \left\lceil \frac{100 \cdot 1024}{1024 - 32} \right\rceil = 104$$

Notase que el último datagrama es menor, en total:

$$P_{ul} = 100 \cdot 1024 - 103 \cdot (1024 - 32) + 32 = 256 \text{ B}$$

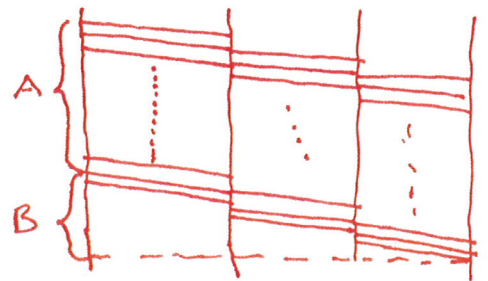
Tenemos dos opciones. La más sencilla es considerar que este datagrama se rellena con 0's hasta 1024 B. La (algo) más complicada es recalcular el tpo de transmisión de este paquete:

$$t_{t,ul} = \frac{P_{ul}}{V_t} = 20,32 \mu s$$

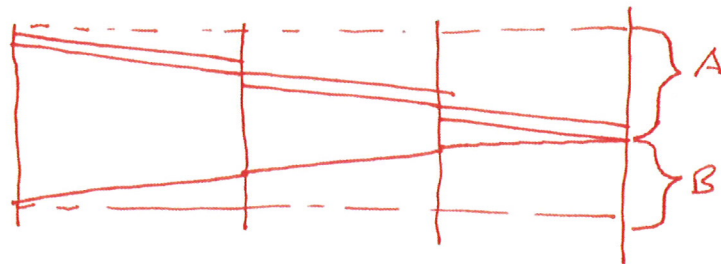
El tiempo total que tarda el mensaje completo en llegar al receptor es igual a:

$$T_{tot} = \underbrace{103 \cdot t_t}_A + \underbrace{(t_{t,ul} + 2t_t + t_p)}_B$$

$$\boxed{T_{tot} = 8,5 \text{ ms}}$$



b) En este caso, hay que establecer si la ventana es lo suficientemente grande para estar en situación de eficiencia unidad, y por tanto no produce ningún retardo adicional. En otro caso, habría que calcular el tiempo muerto incluido en el emisor. Para ello, primero se calcula el tiempo que tarda en llegar el primer ACK



(Nota: se ha considerado que no hay ACK retardado, como en TCP)

$$t_{sack} = \underbrace{3 \cdot (t_t + t_p)}_A + \underbrace{3 \cdot (t_{ack} + t_p)}_B = 256,44 \mu s$$

$$\text{con } t_{ack} = \frac{32 \times 8b}{100 \text{ Mbps}} = 2,56 \mu s$$

Si este tiempo es menor que lo que se tarda en enviar la ventana completa estamos en eficiencia unidad

$$t_{vent} = 5 \cdot t_t = 409,6 \mu s > t_{sack}$$

Por tanto, el tiempo empleado es el mismo pero con la ventaja de realizar control de flujo que permite los reenvíos si se pierden o corrompen datagramas y no saturar al receptor.

4

a)

R1

Subred	Máscara	Sig. N.º
100.168.0.0	/30	—
192.168.0.0	/24	—
192.168.0.0	/22	192.168.0.2 (R2) → permite resumir 1.0, 2.0 y 3.0
192.168.4.0	/22	192.168.0.3 (R3) → permite resumir 4.0 y 5.0 y 6.0
192.168.8.0	/24	192.168.0.4 (R4)
192.168.8.0	/23	192.168.0.4 (R4) → permite resumir 8.0 y 9.0
0.0.0.0	/0	100.168.0.1

b) Una solución sencilla (aunque deje direcciones sin asignar) es sustituir 192.168.0 por 192.168.10.0, quedando:

R1

Subred	Máscara	Sig. N.º
100.168.0.0	/30	—
192.168.0.0	/24	—
192.168.0.0	/22	R2 → 1.0, 2.0 y 3.0
192.168.4.0	/22	R3 → 4.0, 5.0, 6.0
192.168.8.0	/22	R4 → 8.0, 9.0, 10.0
0.0.0.0	/0	100.168.0.1

Otra solución es replantear toda la asignación de direcciones, pero la tabla sería equivalente.