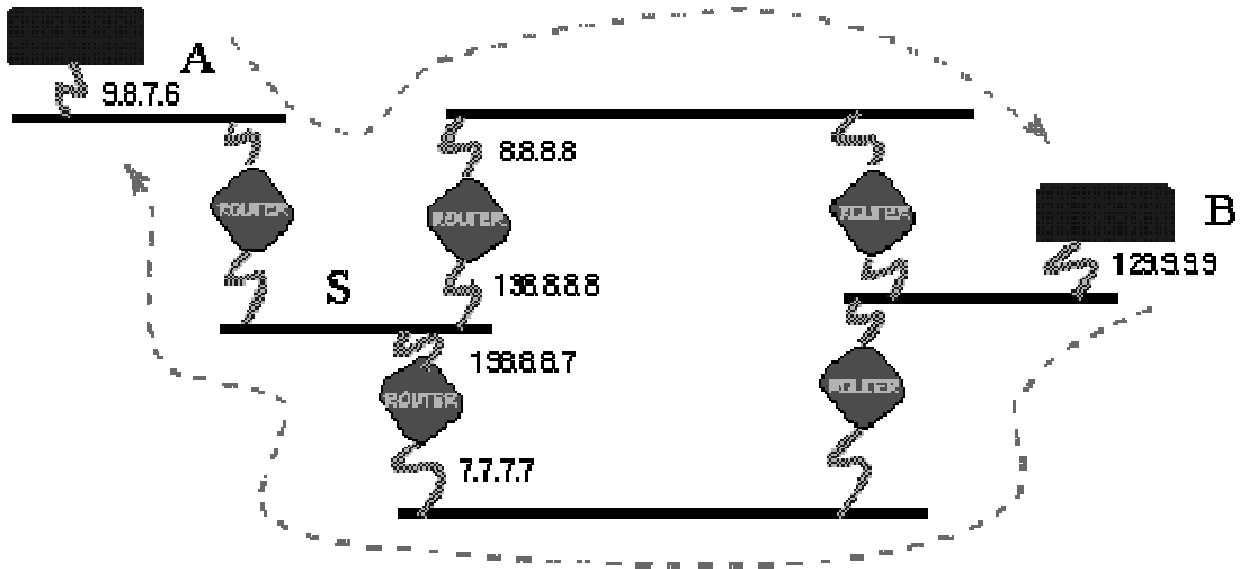


EJERCICIOS DE REDES

Ejercicio 1 (3 puntos)



Entre las máquinas A y B de la figura, se produce la siguiente comunicación.

1. A → B: GET_/_HTTP
2. B → A: Not_Found

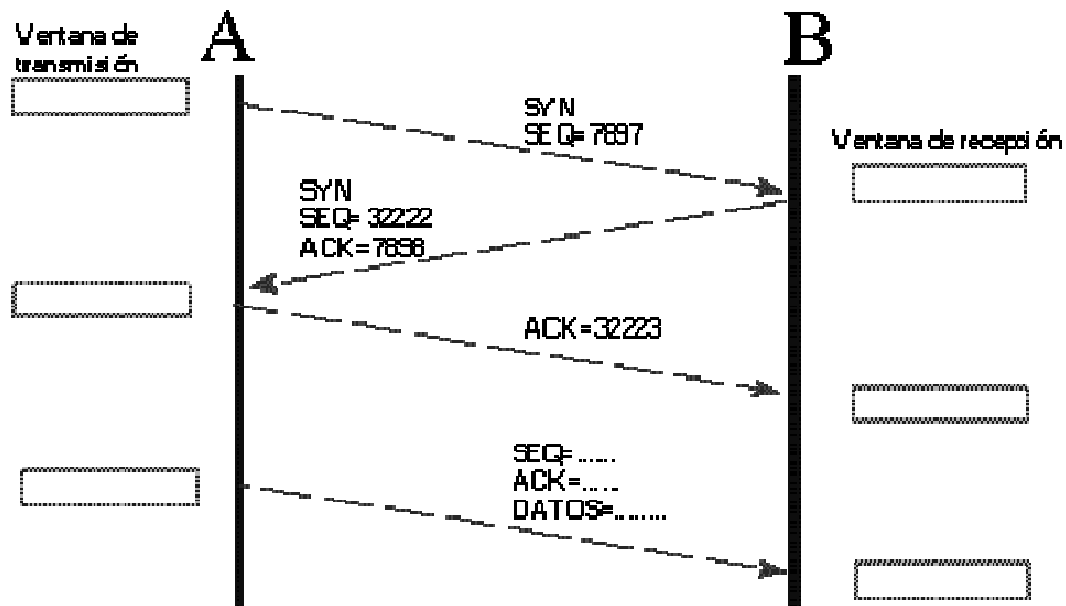
Escribir la trama Ethernet que lleva el primer datagrama IP de A hacia B (tal y como viaja por la subred S). Indíquense los valores de las cabeceras Ethernet, IP y TCP más significativas, y explíquese cómo se obtienen (inventarse valores posibles cuando no haya datos específicos). El servidor WWW de B está en el puerto 80/TCP.

Ejercicio 2 (2.5 puntos)

Escribir las tablas de encaminamiento de A, B y de los routers de la figura para que los datagramas que van de A a B vayan por la parte superior de la figura y los que vayan de B hacia A por la parte inferior, como indican las líneas discontinuas.

Nota: Dar valores IP a los interfaces de los routers a los que no se ha adjudicado valor, teniendo en cuenta que no se emplea subnetting (esto es, respetando la clase A, B ó C de la subred correspondiente). Emplear direcciones IP y no nombres de máquina para definir las tablas.

Ejercicio 3 (2.5 puntos)



En la figura se representa el establecimiento de la conexión entre las máquinas A y B del problema 1. El tamaño de la ventana de transmisión de A es de 8 caracteres. El tamaño máximo del campo de datos de los segmentos TCP utilizados es de 4 caracteres.

Con los datos que le pasa la aplicación, el nivel TCP de A quiere enviar tres segmentos, cuyos datos serán (para cada una):

- GET_
- /_HT
- TP

La secuencia de eventos es la siguiente (se supone que no hay errores, pérdidas ni duplicados en la transmisión):

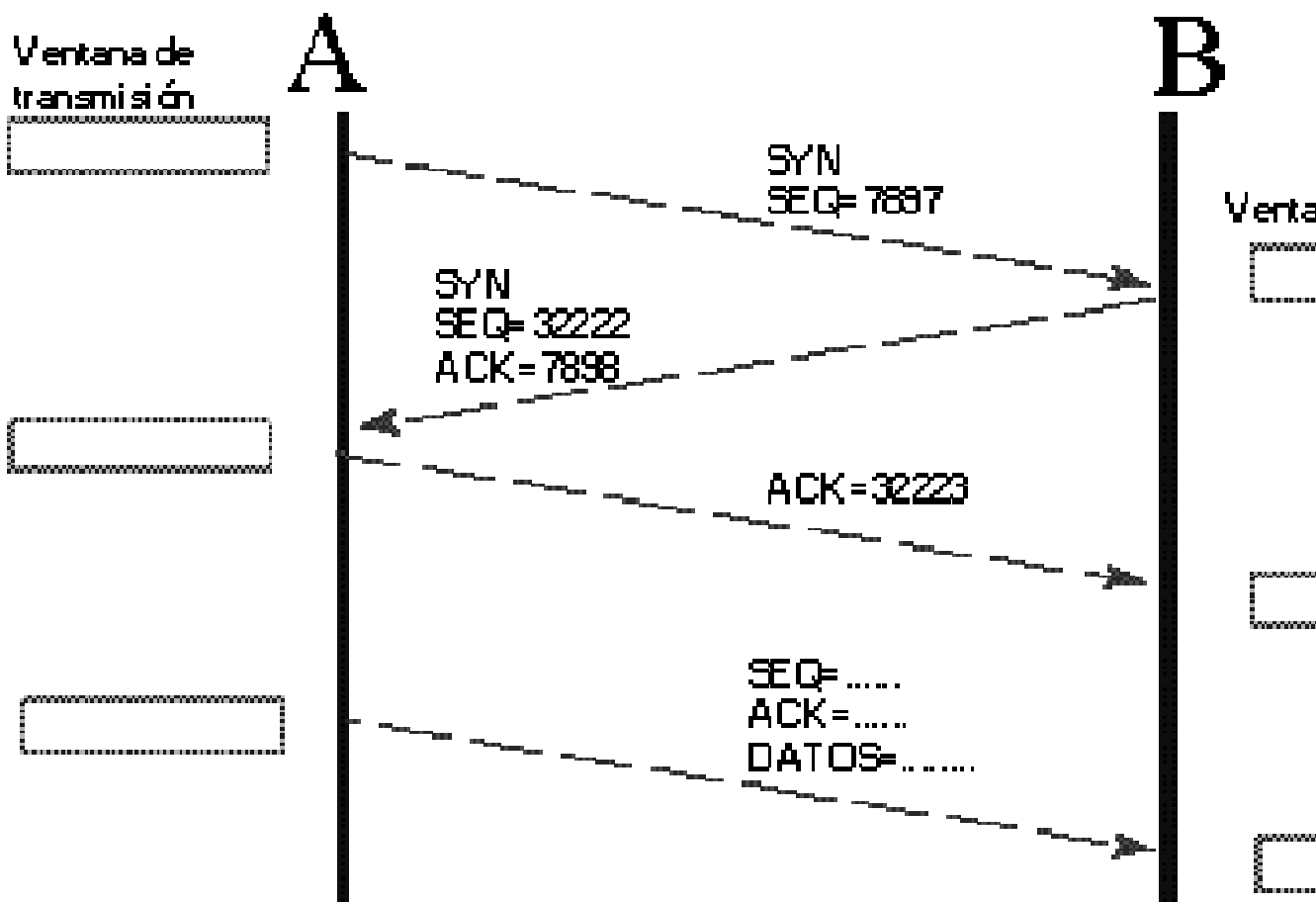
1. El nivel TCP de A envía todos los segmentos que puede.
2. Llegan todos los segmentos al nivel TCP de B, y éste asiente todo lo recibido (de una sola vez).
3. El nivel TCP de B pasa a la aplicación ("saca" de la ventana) todos los datos recibidos hasta el momento.
4. El nivel TCP de A recibe el asentimiento, y envía los datos que aún tiene pendientes.
5. El problema termina cuando el nivel TCP de B asiente estos últimos datos, y A se entera de este asentimiento.

NOTA: La comunicación de B hacia A se limita a los ACKs. El envío de datos con la respuesta se producirá en un momento posterior y no se pregunta en este examen.

Se pide pintar sobre un gráfico como el indicado todos los segmentos descritos en la secuencia anterior, así como los contenidos de las ventanas, indicando los datos y principales campos de las cabeceras TCP (asentimientos y números de secuencia).

Ejercicio 4 (2 puntos)

1. Indíquese el nombre de cada nivel del modelo OSI.
2. ¿Cuál es la diferencia fundamental entre ALOHA y CSMA?
3. Indíquese qué diferencias hay en cuanto a eficiencia entre los códigos de detección de errores y los códigos de corrección de errores.
4. ¿Cuál es la principal diferencia entre los algoritmos de Floyd y Dijkstra?



Ejercicio 1 (Solución)

- Cabecera Ethernet:
 - Destino: Dirección Ethernet de R2 (obtenida por ARP).
 - Origen: Dirección Ethernet de R1 (obtenida de la propia tarjeta).
 - Tipo de protocolo superior: Número correspondiente a IP.
- Cabecera IP:
 - Versión (normalmente 4).
 - Tipo de protocolo superior: Número correspondiente a TCP.
 - Longitud: 5 x 4 (cabecera IP) + 5 x 4 (cabecera TCP) + 0 (no hay datos TCP, paquete de conexión).
 - IP de destino: 129.9.9.9 (B).
 - IP de origen: 9.8.7.6 (A).
- Cabecera TCP:
 - Puerto destino: 80.
 - Puerto origen: uno cualquiera, dado por el S.O.
 - Números A y S (ver problema 3).
 - CRC.
- CRC Ethernet (CRC de todo el paquete Ethernet).

Ejercicio 2 (Solución)

En cuanto a los números IP a asignar a las direcciones que faltan, simplemente hay que tener en cuenta a qué clase pertenece cada una. Una vez se sabe en qué clase están, basta recordar que en la clase A sólo el primer byte es de red, en la clase B los dos primeros y en la clase C los tres primeros bytes son de red. Los que no son de red, son bytes utilizados para numerar máquinas dentro de esa red. Por lo tanto, basta con mirar a qué red está conectado cada interfaz, y elegir un número IP que tenga los mismos bytes de red (teniendo en cuenta la clase de la red), y como bytes de máquina cualquier combinación que no esté ya asignada en esa subred. Por ejemplo, una solución válida sería:

- R1, interfaz "superior": 9.8.7.2 (red clase A: 9.0.0.0).
- R1, interfaz "inferior": 198.8.8.2 (red clase C: 198.8.8.0).
- R2, interfaz "superior": 8.8.8.2 (red clase A: 8.0.0.0).
- R3, interfaz "inferior": 129.9.9.2 (red clase B: 129.9.0.0).
- R4, interfaz "superior": 129.9.9.3 (red clase B: 129.9.0.0).
- R5, interfaz "inferior": 7.7.7.2 (red clase A: 7.0.0.0).

NOTA: Se han numerado los routers empezando por el de arriba a la izquierda (R1), siguiendo hacia su derecha (R2 y R3), y luego por abajo también de izquierda a derecha (R4 y R5).

Para escribir las tablas de encaminamiento, basta tener en cuenta los siguientes puntos:

- A y B sólo tienen que saber cómo alcanzar la red (o máquina) destino (B en el caso de A, A en el caso de B) y su propia red. También se puede poner una entrada para la propia red y otra por defecto, para el router por donde han de salir los paquetes (R1 en el caso de A y R5 en el caso de B).
- Los routers sólo tienen que saber cómo alcanzar las redes a las que están conectados, si han de reenviar los paquetes a través de ellas, y por supuesto, el siguiente router en el camino hacia la máquina final (o la red final).

Con estas consideraciones, unas tablas posibles son las siguientes:

A	default	9.8.7.2 (R1)
	9.0.0.0	9.8.7.6 (A)

B	default	129.9.9.3 (R5)
	129.9.0.0	129.9.9.9 (B)

R1	129.9.0.0	198.8.8.8 (R2)
	9.0.0.0	9.8.7.2 (R1)
	198.8.8.0	198.8.8.2 (R1)

R2	default	8.8.8.2 (R1)
	8.0.0.0	8.8.8.8 (A)

R3	129.9.0.0	129.9.9.2 (R3)
----	-----------	----------------

R4	default	198.8.8.2 (R1)
	198.8.8.0	198.8.8.7 (R4)

R5	default	7.7.7.7 (R4)
	7.0.0.0	7.7.7.2 (R5)

Ejercicio 3 (Solución)

Como la ventana es de 8 caracteres, y los datos que se quieren enviar en los segmentos TCP son como mucho 4 caracteres, se troceará el mensaje en cuatro segmentos (GET_, /_HT, TP). Teniendo en cuenta la secuencia de eventos descrita, primero se enviarán dos segmentos (con GET_ y /_HT), luego el receptor enviará un segmento de ack, luego el emisor enviará el tercer segmento (TP), y por último el receptor enviará el último ack.

Los campos principales de estos mensajes serán los siguientes:

- Emisor → receptor (datos: GET_). S: 7898, A: 32223.
- Emisor → receptor (datos: /_HT). S: 7902, A: 32223, PUSH.
- Receptor → emisor (ack). S: 32222, A: 7906, ACK.
- Emisor → receptor (datos: TP). S: 7906, A: 32223, PUSH.
- Receptor → emisor (ack). S: 32222, A: 7908, ACK.

El contenido de las ventanas será:

- Emisor: ventana completa, con GET_/_HT.
- Al recibir el primer mensaje, el receptor pone GET_ en su ventana.
- Al recibir el segundo mensaje, el receptor pone /_HT en su ventana. Inmediatamente pasa los datos a la aplicación, con lo que la vacía.
- Al recibir el ack, el emisor vacía su ventana. Pone en ella TP antes de enviar su último mensaje.
- Al recibir el último mensaje del emisor, el receptor pone TP en su ventana. Inmediatamente la vacía para pasar los datos a la aplicación, y envía el ack.
- Al recibir el último ack, el emisor vacía su ventana.

Ejercicio 4 (Solución)

1. Físico, enlace, red, transporte, sesión, presentación, aplicación.
2. Fundamentalmente, la detección de portadora (carrier sense).
3. Los códigos de detección necesitan menos redundancia, pero precisan de reenvío en caso de error. Los de corrección precisan de más redundancia, pero en general no precisan de reenvío en caso de error.
4. El algoritmo de Floyd es centralizado, obtiene todas las tablas ``a la vez". El de Dijkstra es distribuido, hay que ejecutarlo para cada nodo, si queremos obtener las tablas de todos los nodos.

Ejercicio 1 (2,5 puntos)

En la red representada en la figura 1 (ver página [□](#)), durante una comunicación TCP/IP entre A y B hemos obtenido una trama que pasa por la Ethernet que une los Routers R1 y R2, tal como indica la figura.

Rellenar en la tabla que se adjunta (y que es donde hay que entregar el resultado):

1. Un valor adecuado para cada uno de los campos indicados .
2. Indicar qué dispositivo ha de poner el valor indicado.
3. Describir brevemente cómo ha obtenido ese valor (qué protocolo ha utilizado, que comunicaciones han tenido que preceder a su obtención, cómo lo ha calculado...).

Ejercicio 2 (3 puntos)

Sea la red representada en la figura 2 (ver página [□](#)). Escribir, cada una en un recuadro indicando claramente a qué máquina corresponde, las tablas de encaminamiento de 1, 2, 3, 4, A y B, con el menor número de entradas posible, para que:

- 1 hable con 2 y 3, pero no con B.

- 2 hable con 1 y 4, pero no con A.
- 3 hable con 1 y 4, pero no con B.
- 4 hable con 2 y 3, pero no con A.

Las direcciones de 1 y 2 son de tipo A, las de 3 y 4 de tipo B, y las de la línea PPP entre A y B de tipo C, y no se utilizan máscaras de red (subnetting).

Ejercicio 3 (2,5 puntos)

Minimalistic Protocols Inc. encarga a su Departamento de Diseño de Protocolos la elaboración de un protocolo de nivel de transporte que ha de funcionar sobre el conocido protocolo IP de nivel de red.

El Director del Proyecto (que sabe lo justito de Transmisión de Datos y Redes) tiene claro que las unidades de datos de dicho protocolo han de tener dos partes: cabecera y datos. Para la parte de datos decide que se va a usar simplemente un byte. Sin embargo no tiene claro cómo diseñar la cabecera, ni los algoritmos a usar en el emisor y el receptor por lo que decide consultarte a ti (como Ingeniero Diseñador de Protocolos recién contratado), y te realiza las siguientes preguntas, que debes contestar razonadamente:

1. ¿Qué deberíamos incluir en la cabecera si quisiéramos ofrecer un servicio de nivel de transporte no orientado a conexión y no fiable? ¿Por qué?
2. ¿Y si queremos poder descartar en el destino de una transmisión los paquetes duplicados que lleguen? Describe no sólo lo que hace falta en la cabecera, sino también el algoritmo de recepción y de envío.
3. ¿Y si queremos, además, poder reordenar en el destino los paquetes que lleguen desordenados? Igualmente, describe la cabecera, y los algoritmos de recepción y de envío.
4. ¿Y si queremos, además, poder recuperar los paquetes que se pierdan? Igualmente, describe la cabecera, y los algoritmos de recepción y de envío.
5. ¿En qué se diferencia el protocolo que has diseñado de TCP?

Nótese que Minimalistic Protocolos Inc. es muy cuidadosa del tamaño que tienen las unidades de datos de sus protocolos, por lo que es conveniente dar, en cada respuesta, el mínimo número de contenidos posibles para la cabecera. También tiene como lema "lo simple es bello", así que los algoritmos utilizados para implementar el protocolo deberían ser lo más simples posible.

Ejercicio 4 (2 puntos)

1. Describe la diferencia fundamental entre la conectividad proporcionada por el nivel 2 y por el nivel 3. Explícala con un ejemplo.
2. Los protocolos de control de acceso al medio con detección de portadora... ¿son mejores o peores que los que no usan detección de portadora? ¿En qué condiciones son mejores o peores? ¿Por qué?
3. En Internet, cuando una máquina envía un paquete IP con destino en alguna otra máquina, conectada a otra subred de Internet... ¿qué máquinas conocen el

camino (secuencia de routers) por el que ese paquete debe pasar? En particular, ¿qué saben la máquina origen y la máquina destino de ese camino?. Explica un poco tu respuesta.

4. ¿Qué garantías ofrece el protocolo TCP con respecto a los datos que transmite? ¿Y UDP?

Figura 1 (ejercicio 1)

Pregunta 1 (5 puntos)

1. (1,25 puntos) Dadas la red de la figura 1, y considerando las tablas de encaminamiento que aparecen en la figura, completa la tabla indicando en la casillas (M,N) en blanco:
 - o NO cuando no hay camino desde M hasta N
 - o La secuencia de encaminadores intermedios cuando sí hay camino desde M hasta N.

Téngase en cuenta:

- o La máscara de subred es siempre 255.255.255.0.
 - o Las entradas ``desvaídas" en las tablas de encaminamiento permiten la conectividad de una máquina con la(s) de su(s) propia(s) subred(es), y están presentes TODAS LAS POSIBLES.
 - o Los trazos gruesos corresponden a redes Ethernet o a líneas punto-a-punto
 - o Los trazos finos corresponden a conexiones desde una tarjeta Ethernet hasta la propia Ethernet.
2. (1,5 puntos) Supongamos que se cae el encaminador R3, quedando fuera de servicio. Reconstruir las tablas de R1, R2, R4, R5, R6 y R7 para que se mantengan las mismas posibilidades de comunicación (obviamente, a través de nuevos caminos).

Nótese que las tablas de A, B, C, X, Y y Z no deben cambiarse, que no deben tocarse las entradas ``desvaídas", y que no es necesario poner en las tablas de la solución dichas entradas ``desvaídas".

3. (0,75 puntos) Indica si la siguiente trama Ethernet puede ser una trama correcta generada en la red de la figura 1 (explicando cuándo y dónde se genera) o no (justificando por qué):

Eth. Destino	Eth. Origen	Protocolo	IP Origen	IP Destino	...
b:b:b:b:b	a:a:a:a:a	IP	10.1.28.9	10.1.74.9	...

4. (0,75 puntos) Indica qué tipo de trama es la siguiente, para qué se generaría en la red de la figura 1 y cuáles serían los campos relevantes que le faltan:

Eth. Destino	Eth. Origen	Protocolo	...
1:2:3:4:5:6	a:a:a:a:a	ARP	...

5. (0,75 puntos) Indica la salida en pantalla que genera el comando `tracert` que hay que invocar para que aparezca en la red de la figura 1 la siguiente trama:

Eth. Destino	Eth. Origen	Protocolo	IP Origen	IP Destino	Protocolo	...
b:b:b:b:b	7:8:9:a:b:c	IP	10.1.28.9	10.1.74.9	UDP	...

Pregunta 2 (1,25 puntos)

Se desea transmitir un partido de fútbol en directo con una señal digital sobre una línea de comunicaciones de fibra óptica, de una gran longitud y de un gran ancho de banda. ¿Será más aconsejable usar un método de detección de errores o uno de corrección de errores en dicha línea? Razona tu respuesta.

Datos a tener en cuenta:

- Vídeo: A 25 imágenes por segundo, cada imagen codificada en 50 Kbytes.
- Retardo de propagación en la línea: 1 segundo
- Ancho de banda: 50 Gigabits por segundo

Pregunta 3 (1,25 puntos)

Se tiene una red Ethernet en la que hay unas 100 máquinas que consultan intensivamente 2 servidores de WWW de la misma red. 50 máquinas consultan siempre al mismo servidor de WWW, y las otras 50 al otro servidor. Se ha observado que la red se encuentra muy cargada de tráfico y en ella hay muchas colisiones. Para paliar el problema se decide partir en dos trozos la Ethernet, pero manteniendo a todas las máquinas en la misma subred (es decir, con una dirección IP con la misma parte de red y subred). ¿Debe colocarse entre ambas partes de la Ethernet un repetidor (repeater) o un puente (bridge) "inteligente" (es decir, que "aprende" direcciones Ethernet). Indica también cómo quedarían conectadas las máquinas y los servidores de red a los dos trozos de Ethernet. Razona tu respuesta.

Pregunta 4 (2,5 puntos)

En la secuencia de envío de segmentos TCP reflejada en la figura 2, se sabe que ni A ni B quieren ya enviar más datos. Escribe los segmentos que faltan para que la conexión quede cerrada adecuadamente, suponiendo:

- Las líneas discontinuas horizontales indican tics del reloj.
- Los segmentos tardan en llegar al receptor medio tic del reloj (si no se pierden).
- El plazo en que las máquinas esperan a que llegue un ACK es de 5 tics del reloj.
- A partir de lo último dibujado en la figura, ya no se perderán más segmentos.
- Cuando a "B" le llegue otro segmento de A responderá inmediatamente (en el siguiente tic de reloj) con un ACK.

Figura 1

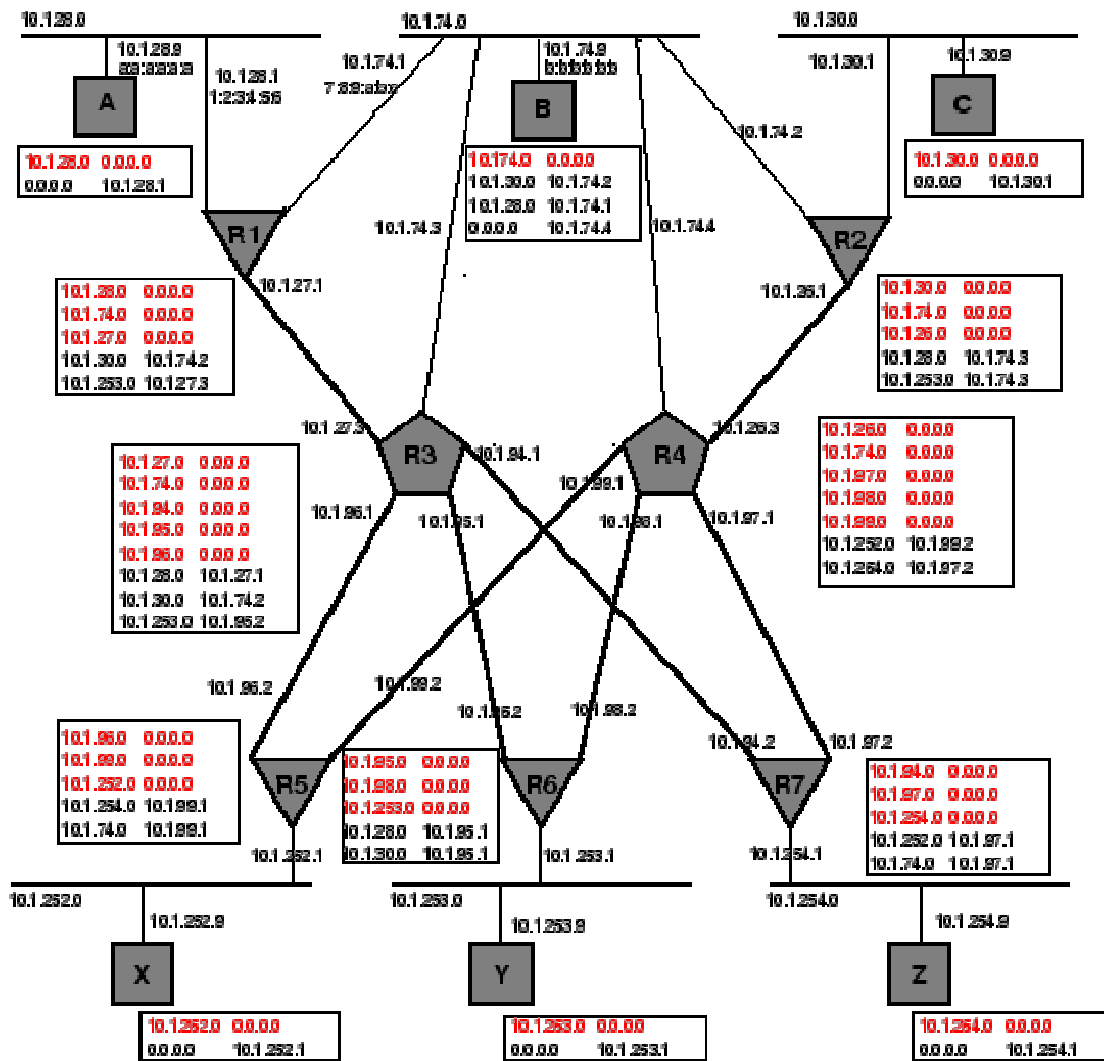
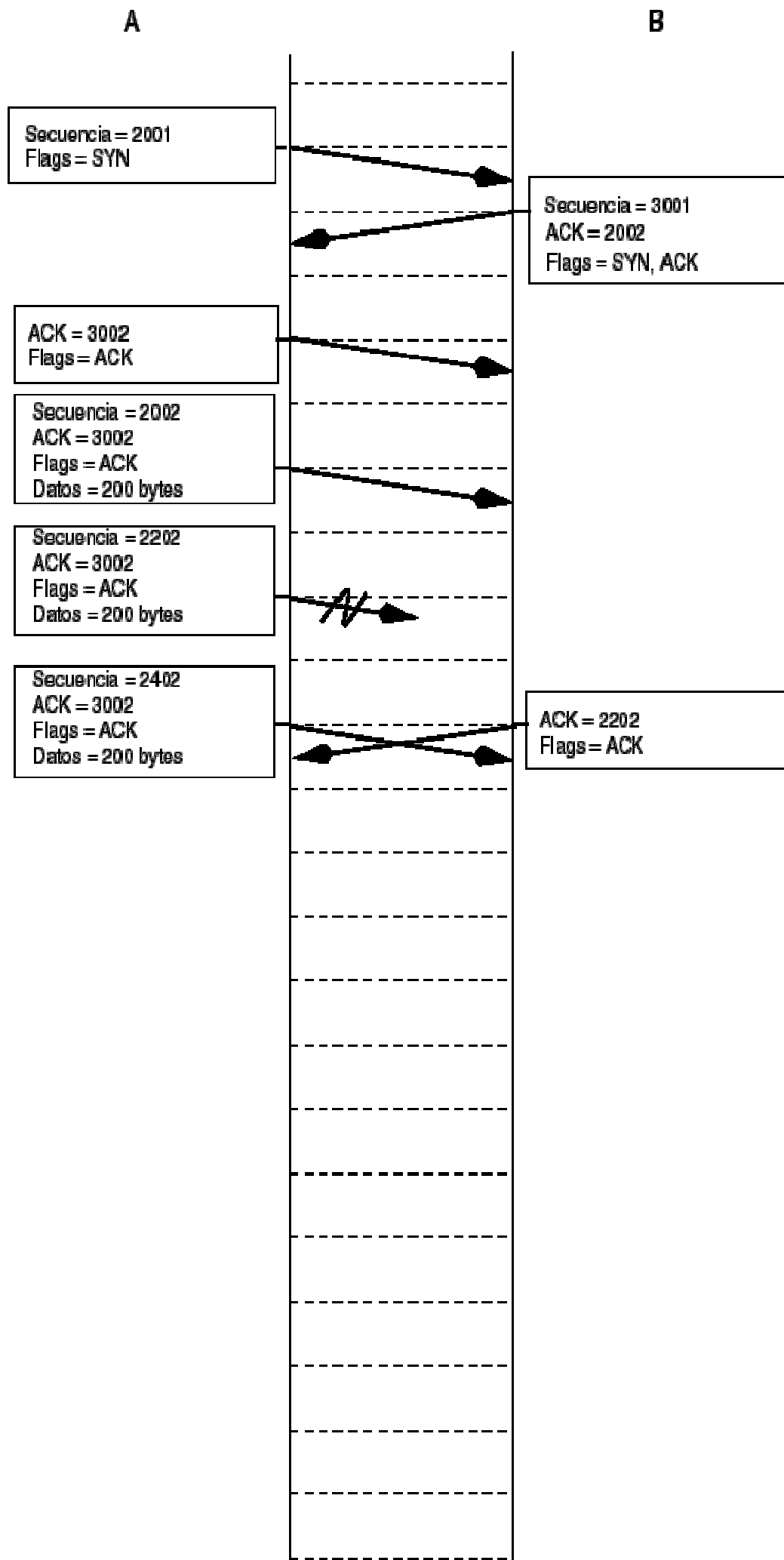


Figura 2



Nombre y Apellidos:	
Grupo:	
Asignatura:	Transmisión de Datos y Redes
Titulación:	Ingeniería Técnica de Informática de Gestión
Fecha:	26 de Junio de 1998

Pregunta 1 (solución)

1.

	A	B	C	X	Y	Z
A	-	R1	R1, R2	NO	R1, R3, R6	NO
B	R1	-	R2	R4, R5	NO	R4, R7
C	R2, R3, R1	R2	-	NO	R2, R3, R6	NO
X	NO	R5, R4	NO	-	NO	R5, R4, R7
Y	R6, R3, R1	NO	R6, R3, R2	NO	-	NO
Z	NO	R7, R4	NO	R7, R4, R5	NO	-

2.

Tabla de R1	
10.1.30.0	10.1.74.2
10.1.253.0	10.1.74.2
Tabla de R2	
10.1.28.0	10.1.74.1
10.1.253.0	10.1.26.3
Tabla de R4	
10.1.28.0	10.1.74.1
10.1.30.0	10.1.26.1
10.1.252.0	10.1.99.2

- campo indicando si se trata de solicitud o respuesta (en este caso indicaría ``respuesta")
- dirección IP de la máquina por cuya dirección Ethernet se preguntaba (en este caso, la de A)
- dirección Ethernet preguntada (al ser una respuesta este campo va relleno con la dirección Ethernet de A)
- dirección IP de la máquina que hizo la pregunta (en este caso, la de R1)
- dirección Ethernet de la máquina que hizo la pregunta (en este caso, la de R1)

10. Con las direcciones IP de origen y destino se ve que es A quien hace un traceroute a B, es decir se ejecuta en A:

11. `traceroute 10.1.74.9`

Se mostrará en la salida (y por triplicado) el tiempo que tarda en llegar la respuesta de:

- cada encaminador intermedio (en este caso sólo R1), que envía un ICMP indicando que llegó a 0 el TTL del datagrama enviado por A.
- el destino (en este caso B), que envía un ICMP indicando que el datagrama iba dirigido a un puerto inexistente.

Es decir:

```
1  R1 (10.1.28.1)  1 ms 2 ms 2 ms
2  B  (10.1.74.9)  5 ms 6 ms 4 ms
```

Nombre y Apellidos:	
Grupo:	
Asignatura:	Transmisión de Datos y Redes
Titulación:	Ingeniería Técnica de Informática de Gestión
Fecha:	26 de Junio de 1998

Pregunta 2 (solución)

Lo más aconsejable sería usar un método de corrección de errores.

En 1 segundo hay que mandar 25 imágenes de 50 kilobytes cada una, luego hay que mandar en total $25 * 50 * 8 = 10.000$ kilobits, algo menos de 10 megabits. El ancho de banda es unas 5000 veces eso.

Si se utilizara detección + retransmisión, una imagen perdida tardaría al menos 3 segundos en recuperarse desde que se transmitió por primera vez, retardo inaceptable en una transmisión en tiempo real.

El ancho de banda me permitiría enviar cada bit repetido 5000 veces, lo que me permitiría fácilmente corregir los errores de transmisión en destino sin perder ningún tiempo en retransmisiones.

Es cierto que si hay pocos errores en la línea podría simplemente olvidarme de los errores, y volcar a la pantalla las imágenes que llegan tal cual, ya que el ojo no lo va a detectar. Pero el enunciado no da ninguna indicación de lo ruidosa que puede ser la línea, y por otro lado no aprovecho el enorme ancho de banda que me queda libre para incluir información redundante que me permita eliminar los errores.

Pregunta 3 (solución)

La situación descrita sólo mejora colocando un puente inteligente.

Un repetidor trabaja al nivel físico, lo que quiere decir que repite la señal eléctrica que hay en un trozo de la Ethernet en el otro. Eso implica que, a los efectos de tráfico y número de colisiones, es exactamente lo mismo tener una sola Ethernet o dos unidas por un repetidor.

En cambio un puente trabaja al nivel de enlace, lo que le permite reconocer las tramas Ethernet que hay en el cable. Si es inteligente "aprende" con el tiempo las direcciones Ethernet que hay en cada lado, con lo que al ver por uno de sus lados una cierta trama sabe si es necesario o no copiarla en el otro, con lo que se aísla tráfico. Para sacar el máximo partido a esta ventaja hay que conectar:

- a un lado, un servidor WWW con las 50 máquinas que lo consultan con frecuencia
- a otro, el otro servidor WWW con las otras 50 máquinas

Pregunta 4 (solución)

