

**Examen de Redes de Computadores.
Ingeniería Técnica en Informática de Gestión
Enero de 2010**

Duración: 2 horas. Se permiten libros y apuntes.

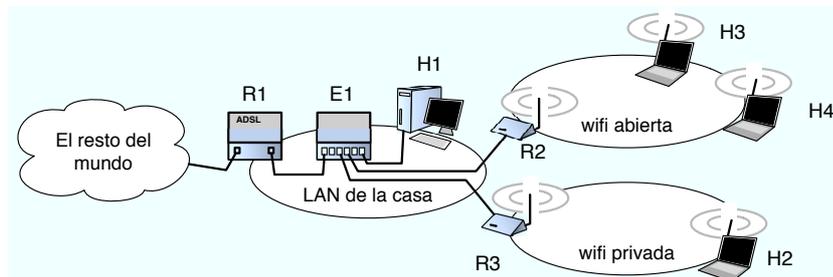


Figura 1: Red del usuario de ADSL

Un usuario de ADSL quiere utilizarlo para configurar una red como la de la figura, con objeto de tener acceso a Internet en su casa que incluya una red inalámbrica privada y otra red inalámbrica con la que pueda dar acceso a Internet a los usuarios de la cafetería que hay debajo de su casa. Ha estimado que se conforma con permitir que 20 ordenadores estén conectados al mismo tiempo en la red inalámbrica abierta. En la privada como es para el y sus invitados vale con que pueda haber 4 ó 5 ordenadores y en la red de cable le vale con poder tener los equipos que aparecen en la figura, quiere tener un ordenador permanentemente conectado en la red de cable (H1 en la figura). Para ello se ha acogido a una oferta de su proveedor que le proporcionaba IPs públicas por poco dinero y ha conseguido el bloque de direcciones IP 43.2.1.128/26 todo para él.

Pregunta 1: (1 pto) Indique como repartiría el bloque disponible de direcciones IP en subredes IP para cada una de las zonas. Tenga en cuenta que como se ve en la pregunta 2 que ya ha decidido que quiere que el router R1 tenga la dirección IP 43.2.1.129.

Zona	subred IP	primera IP válida	ultima IP válida
LAN de casa			
Wifi abierta			
Wifi privada			

Zona	subred IP	primera IP válida	ultima IP válida
LAN de casa	43.2.1.128 / 28	43.2.1.129	43.2.1.142
Wifi abierta	43.2.1.160 / 27	43.2.1.161	43.2.1.190
Wifi privada	43.2.1.144 / 28	43.2.1.145	43.2.1.158

Pregunta 2: (1 pto) Esta es la configuración de los routers así como de algunos PCs antes de elegir las IPs. Se ha decidido que el router R1 queremos que use en el interfaz interno la dirección IP 43.2.1.129. Indique la configuración del resto de direcciones IP, máscaras y router por defecto en los PCs.

R1> conf externo: IP: 43.1.1.121 mask: 255.255.255.252 MAC: 00:33:00:11:00:11 eth1: IP: 43.2.1.129 mask: _____ MAC: 00:33:00:11:00:22 Routing table <table border="1"> <thead> <tr> <th>Net</th> <th>Next hop</th> </tr> </thead> <tbody> <tr> <td>...</td> <td>..</td> </tr> </tbody> </table>	Net	Next hop	R2> conf eth: IP: _____ mask: _____ MAC: 00:33:00:22:00:11 wifi: IP: _____ mask: _____ MAC: 00:33:00:22:00:22 Routing table <table border="1"> <thead> <tr> <th>Net</th> <th>Next hop</th> </tr> </thead> <tbody> <tr> <td>default</td> <td>43.2.1.129</td> </tr> </tbody> </table>	Net	Next hop	default	43.2.1.129	R3> conf eth: IP: _____ mask: _____ MAC: 00:33:00:33:00:11 wifi: IP: _____ mask: _____ MAC: 00:33:00:33:00:22 Routing table <table border="1"> <thead> <tr> <th>Net</th> <th>Next hop</th> </tr> </thead> <tbody> <tr> <td>default</td> <td>43.2.1.129</td> </tr> </tbody> </table>	Net	Next hop	default	43.2.1.129
Net	Next hop													
...	..													
Net	Next hop													
default	43.2.1.129													
Net	Next hop													
default	43.2.1.129													
PC H1 IP: _____ mask: _____ MAC: 00:99:00:00:00:11 default router: _____	PC H2 IP: _____ mask: _____ MAC: 00:99:00:00:00:22 default router: _____	PC H3 IP: _____ mask: _____ MAC: 00:99:00:00:00:33 default router: _____												

Una solución

R1> conf externo: IP: 43.1.1.121 mask: 255.255.255.252 MAC: 00:33:00:11:00:11 eth1: IP: 43.2.1.129 mask: 255.255.255.240 MAC: 00:33:00:11:00:22 Routing table <table border="1"> <thead> <tr> <th>Net</th> <th>Next hop</th> </tr> </thead> <tbody> <tr> <td>...</td> <td>..</td> </tr> </tbody> </table>	Net	Next hop	R2> conf eth: IP: 43.2.1.130 mask: 255.255.255.240 MAC: 00:33:00:22:00:11 wifi: IP: 43.2.1.161 mask: 255.255.255.225 MAC: 00:33:00:22:00:22 Routing table <table border="1"> <thead> <tr> <th>Net</th> <th>Next hop</th> </tr> </thead> <tbody> <tr> <td>default</td> <td>43.2.1.129</td> </tr> </tbody> </table>	Net	Next hop	default	43.2.1.129	R3> conf eth: IP: 43.2.1.131 mask: 255.255.255.240 MAC: 00:33:00:33:00:11 wifi: IP: 43.2.1.145 mask: 255.255.255.240 MAC: 00:33:00:33:00:22 Routing table <table border="1"> <thead> <tr> <th>Net</th> <th>Next hop</th> </tr> </thead> <tbody> <tr> <td>default</td> <td>43.2.1.129</td> </tr> </tbody> </table>	Net	Next hop	default	43.2.1.129
Net	Next hop													
...	..													
Net	Next hop													
default	43.2.1.129													
Net	Next hop													
default	43.2.1.129													
PC H1 IP: 43.2.1.132 mask: 255.255.255.240 MAC: 00:99:00:00:00:11 default router: 43.2.1.129	PC H2 IP: 43.2.1.146 mask: 255.255.255.240 MAC: 00:99:00:00:00:22 default router: 43.2.1.145	PC H3 IP: 43.2.1.162 mask: 255.255.255.224 MAC: 00:99:00:00:00:33 default router: 43.2.1.161												

Pregunta 3: (1 pto) Indique que tabla de rutas debe configurar en el router R1. El proveedor de servicio le ha proporcionado la dirección del router por defecto y es la IP 43.1.1.122

<i>Routing table</i>	
<i>Net</i>	<i>Next hop</i>
43.2.1.128 / 27	-
43.1.1.120 / 30	-
43.2.1.160 / 27	43.2.1.130
43.2.1.144 / 28	43.2.1.131
default	43.1.1.122

Pregunta 4: (1 pto) ¿Qué camino sigue un paquete que va de H2 a H3? Razone la respuesta indicando todas las decisiones en PCs y routers. ¿Cuántos niveles IP diferentes envían el paquete a otro nivel IP?

Respuesta correcta:
H2 tiene un paquete para H3 (43.2.1.162) H2 ve que 43.2.1.162 no pertenece a la red vecina 43.2.1.144/28 así que lo envía al router por defecto R3
R3 tiene un paquete para 43.2.1.162 R3 ve que 43.2.1.162 no pertenece a ninguna de sus redes vecinas así que envía el paquete a su router por defecto 43.2.1.129 (R1)
R1 tiene un paquete para H3 (43.2.1.162) R1 decide que 43.2.1.162 no pertenece a ninguna red vecina así que mira en la tabla de rutas... La dirección 43.2.1.162 verifica la regla 43.2.1.160/27 así que el siguiente salto es 43.2.1.130(R2) le envía el paquete
R2 tiene un paquete para 43.2.1.162 R2 ve que 43.2.1.162 pertenece a la red vecina 43.2.1.160/27 así que envía el paquete a 43.2.1.162
H3 tiene un paquete para 43.2.1.162 H3 ve que el paquete es para él así que lo entrega al nivel superior
En total 4 niveles IP han reenviado el paquete a otro

Pregunta 5: (1 pto) El siguiente paquete enviado desde H2 a H3 ha sido capturado en el PC H1 en LAN de la casa, Algunos campos han sido dejados en blanco porque dependen de las direcciones que haya elegido en la pregunta 2. Indique las direcciones MAC que aparecerán en la cabecera ethernet.

Se muestra la trama a nivel de enlace, excluyendo el preámbulo y el CRC.

```
0x0000: [AAA AAAA AAA] [BBB BBBB BBB] 0800 4510 .....&J.*...E.  
0x0010: 0034 f238 4000 4006 [CS] [ IP H2 ] [ IP .4.8@.@.....  
0x0020: H3 ] d5ed 0017 371d efeb b85f c13b 8010 .....7...._;...  
0x0030: ffff 37b2 0000 0101 080a 1e6d db7b 01d0 ..7.....m.{..  
0x0040: 6789 g.
```

[AAA AAAA AAA] = _____

[BBB BBBB BBB] = _____

Respuesta correcta:

Si el paquete se ha capturado cuando R3 se lo enviaba a R1 tendra estas direcciones

[AAA AAAA AAA] = 00:33:00:11:00:22

[BBB BBBB BBB] = 00:33:00:33:00:11

Si el paquete se ha capturado cuando R1 se lo enviaba a R2 tendra estas direcciones

[AAA AAAA AAA] = 00:33:00:22:00:11

[BBB BBBB BBB] = 00:33:00:11:00:22

Pregunta 6: (1 pto) ¿A qué aplicación pertenece el paquete capturado? ¿Qué acción ha realizado el usuario de H2 para provocar su envío?

Respuesta correcta:

El paquete es un paquete TCP que va del puerto origen 0xd5ed al puerto 0x0017 que en decimal es el 23 que es el puerto normalmente asignado a telnet

Así que el paquete parece un paquete de cliente a servidor de una conexión de telnet H2 es el cliente y H3 el servidor

Así que el usuario de H2 parece que ha lanzado una aplicación de telnet para conectarse a H3

Pregunta 7: (1 pto) ¿Cuales de los siguientes son ciertos sobre el paquete anterior?

- a) Transporta un paquete UDP
- b) Transporta un paquete TCP
- c) Transporta un paquete de ARP
- d) Ha sido enviado por el cliente
- e) Ha sido enviado por el servidor
- f) Transporta datos de nivel de aplicación
- g) Es un paquete de establecimiento de una conexión TCP (SYN)
- h) Es un paquete de cierre de una conexión TCP (FIN)
- i) Es un paquete de confirmación en una conexión TCP (ACK)

Respuesta correcta:

Transporta un paquete TCP

Ha sido enviado por el cliente

Es un paquete de confirmación en una conexión TCP (ACK)

El usuario quiere saber que hacen los usuarios que se conectan a su red inalámbrica abierta y para ello pretende averiguar que servidores TCP están utilizando en las direcciones IP que están en la red activas en la wifi abierta. Comienza a escribir un programa para barrer todos los puertos en todas las direcciones de esa red comprobando si existe un servidor. Lo primero escribe esta función

```
int hay_servidor_en(char *ipstr, int puerto) {
    in_addr_t ip;
    struct sockaddr_in servidor;
    int s;
    int err;

    ip=inet_addr(ipstr);
    s=socket(PF_INET,SOCK_STREAM,0);
    servidor.sin_family=AF_INET;
    servidor.sin_port=htons(puerto);
    servidor.sin_addr.s_addr=ip;
```

```
err=connect(s,(struct sockaddr *)&servidor,sizeof(servidor));
if (err==0) {
    printf("No hay servidor escuchando en el puerto %u de %s\n", puerto, ipstr);
    return 0;
} else {
    close(s);
    printf("Hay servidor escuchando en el puerto %u de %s\n", puerto, ipstr);
    return 1;
}
}
```

Pregunta 8: (1 pto) ¿Qué errores ha cometido al escribir la función?

- a) Ningún error, funciona, e indica correctamente si el servidor indicado está activo.
- b) El `SOCK_STREAM` debería ser `SOCK_DGRAM` ya que IP es una red de datagramas.
- c) La asignación de `ip` en la estructura `servidor` debe pasarse antes por un `htonl()`.
- d) La entrada ya es una dirección IP no hace falta traducir el nombre con `inet_addr()`.
- e) `ip` debe ser en mayúsculas IP.
- f) En el caso de que no haya servidor escuchando, se ha olvidado de cerrar el socket con `close()`.
- g) Los casos del `if` están al revés cuando `err==0` debería decir que si hay servidor escuchando y en caso contrario decir que no.
- h) El puerto debería imprimirse con `%d` en lugar de `%u`.
- i) No se envía nada al servidor una vez establecida la conexión por lo que el programa dejará colgados todos los servidores que encuentre, esperando a que les envíe algo, con lo que dejarán de aceptar peticiones de otros clientes. Aunque esto no hace que el programa no funcione y si que encuentra los servidores, será poco discreto porque los usuarios de los ordenadores de la red wifi abierta se darán cuenta de que pasa algo raro.

Respuesta correcta:

Los casos del if están al revés cuando err==0 debería decir que si hay servidor escuchando y en caso contrario decir que no.

La función `main()` utiliza la función anterior `hay_servidor_en()` para barrer todas las direcciones y puertos de interés buscando posibles servidores. Sólo le interesan los puertos por debajo del 8000. Por lo que hace un bucle de 1 a 8000. Pero el bucle que barre las direcciones IP depende de las que haya elegido antes.

Pregunta 9: (1 pto) Complete la función `main()` para que barra todas las direcciones de la red wifi abierta llamando correctamente a la función anterior.

```
int main(int argc, char *argv[]) {
    int i,j;    char ipstr[50];
    // hacer un bucle que ponga en j todos los valores posibles de IP
    for (j=...    ) {

        // poner en ipstr el valor de la IP que quiero probar

        for (i=1;i<8000;i++) {
            hay_servidor_en(ipstr,i);
        }
    }
}
```

Respuesta correcta:

```
int main(int argc, char *argv[]) {
    int i,j;
    char ipstr[50];

    for (j=161;j<=190;j++ ) {

        sprintf(ipstr, "43.2.1.%u",j);

        for (i=1;i<8000;i++) {
            hay_servidor_en(ipstr,i);
        }
    }
}
```

Pregunta 10: (1 pto) El usuario lanza el programa en H2. En ese momento en la red inalámbrica abierta hay un solo ordenador conectado H3, con la dirección IP que le ha asignado en la pregunta 2. El usuario de H3 está utilizando un programa de correo para leer su correo (utilizando POP3), mientras tiene una ventana de telnet abierta en la que esta ejecutando comandos remotamente en el ordenador de su empresa. Además tiene activado en el portátil H3 que comparta una página web personal alojada en su directorio y tiene funcionando un programa de BitTorrent (Un programa peer-to-peer) con el que esta bajándose la última distribución de su sistema operativo. ¿Que resultados dará el programa de H2?

Respuesta correcta:

De todos los mencionados solo el compartir paginas web y la aplicación peer-to-peer tendran sockets escuchando en TCP Asi que en general aparecera un monton de lineas

No hay servidor escuchando en el puerto 1 de 43.2.1.162

No hay servidor escuchando en el puerto 2 de 43.2.1.162

...

No hay servidor escuchando en el puerto 7999 de 43.2.1.162

Las únicas líneas que si encontrara seran

Hay servidor escuchando en el puerto 80 de 43.2.1.162

por el servidor web y otro en el puerto que escuche la aplicación peer2peer si es menor que el 8000, concretamente en BitTorrent suele ser el 6881 aunque no se pide saberse el número exacto

Hay servidor escuchando en el puerto 6881 de 43.2.1.162