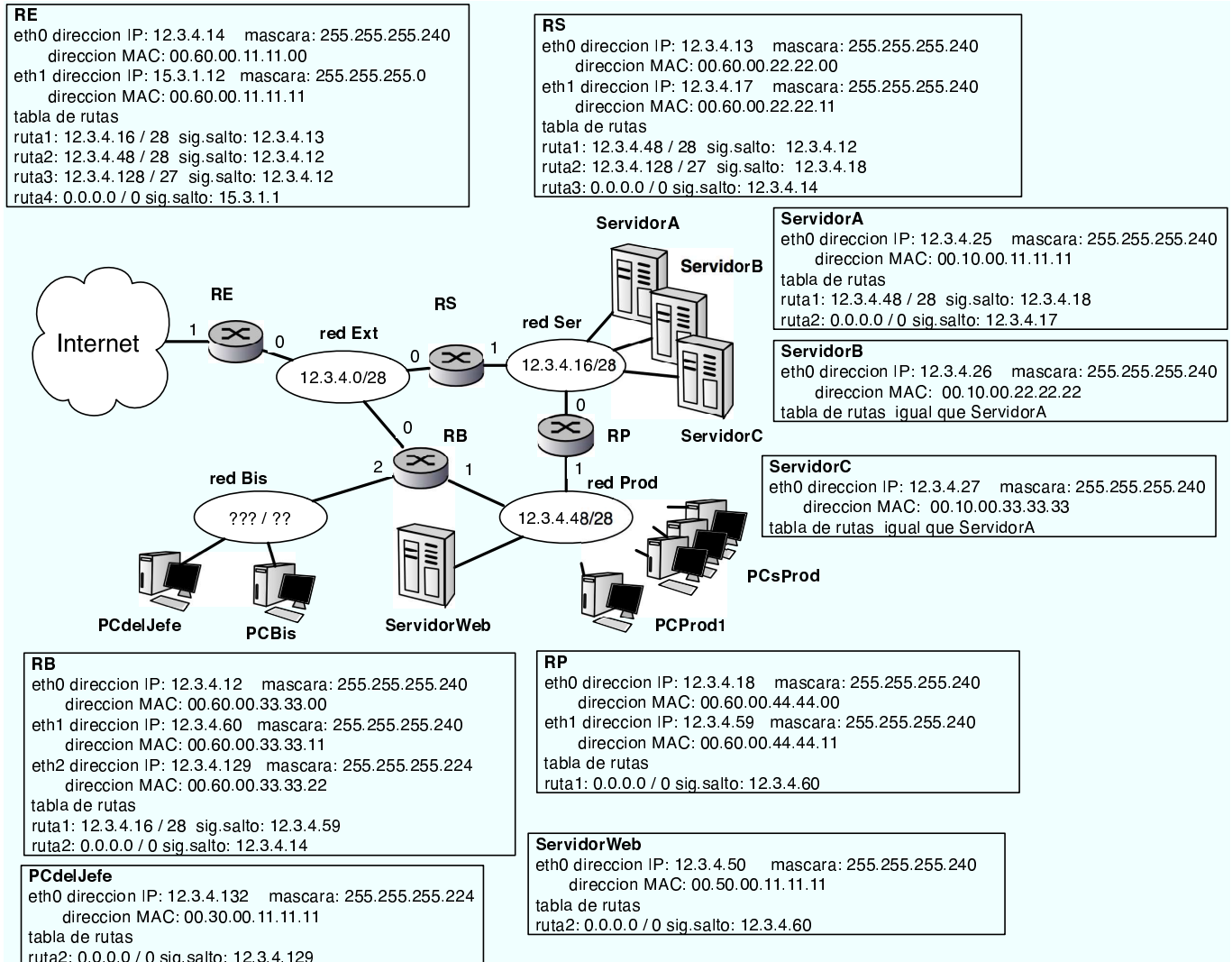


Examen de Redes de Computadores.
Ingeniería Técnica en Informática de Gestión
Septiembre de 2008
Número de identificación del examen: examid
 Duración: 2 horas. Se permiten libros y apuntes.



Le han nombrado administrador de red de una empresa que vende un juego multijugador en red. Su red está construida por varias redes interconectadas como se ve en la figura. La red Ser en la que están los servidores del juego. La red Prod contiene a los ordenadores de producción y desarrollo desde los que controlan los servidores públicos del juego, así como el servidor web. La red Bis contiene los ordenadores de negocios de la empresa y no relacionados directamente con el desarrollo.

Pregunta 1: (0.833 ptos) A la vista de la información de que dispone ¿cual es la subred IP y el rango de direcciones de la red Bis? Razone la respuesta

Pregunta 2: (0.833 ptos) Como administrador se le ha encargado reorganizar la red Prod. Se han desconectado todos los PCs de esa red a excepción del servidor web, con lo que ahora la red esta como se ve en la figura. Indique la configuración que instalará en uno de los PCs de producción (PCProd1). ¿Cual es la dirección IP mas alta y mas baja que puede emplear en un ordenador de producción?

Configuracion de PCProd1

Direccion IP:

Mascara:

Tabla de rutas:

Dirección IP más alta utilizable:

Dirección IP más baja utilizable:

Pregunta 3: (0.833 ptos) Se pretende colocar 25 PCs en la red de producción. ¿Se puede con la configuración de red actual? ¿Cuál es el máximo numero de PCs que podemos añadir a la red Prod de la figura? Si el rango de IPs asignado a la empresa es el 12.3.4.0/24 y no queremos cambiar la direccion IP del servidor web. Asigne una subred a la red Prod de forma que podamos direccionar los 25 PCs pero no tengamos que cambiar ninguna dirección IP de las ya asignadas.

Pregunta 4: (0.833 ptos) Un administrador de los servidores de juego configura incorrectamente la máscara del servidor B a 255.255.255.192. Se excusa diciendo que tras el cambio sigue funcionando y los jugadores externos se conectan y juegan normalmente. Sin embargo parece que ahora el servidor es inaccesible desde el servidor web. ¿Es posible? ¿Por qué falla el ping desde el servidor web al servidor B? ¿Por que funciona el ping a un jugador exetrno? Razone la respuesta

Pregunta 5: (0.833 ptos) ¿Dónde se pierden los paquetes (a consecuencia del fallo de la pregunta 4) si el ServidorB envía un paquete de datos al servidorWeb?

- a) El router RS los envía a RE y RE vuelve a enviarlos a RS y así sucesivamente hasta que uno de los dos los descarta por demasiados saltos
- b) En el router RP, que recibe los paquetes de ServidorB a causa del cambio de máscara pero al ser el camino incorrecto se ve obligado a descartarlos
- c) En el Servidor B, que recibe los paquetes del ping pero nunca llega a enviar las respuestas nunca llega a enviarlos
- d) En el servidor web, que descarta los paquetes que vienen de ServidorB ya que no llevan la máscara correcta

Pregunta 6: (0.833 ptos) ¿Qué camino siguen los paquetes de un ping desde ServidorWeb a un servidor de juego configurado correctamente como ServidorA?

- a) La ida *ServWeb* → *RB* → *RP* → *ServA* y la vuelta *ServA* → *RP* → *ServWeb*
- b) La ida *ServWeb* → *RB* → *RS* → *ServA* y la vuelta *ServA* → *RS* → *RB* → *ServWeb*
- c) La ida *ServWeb* → *RP* → *ServA* y la vuelta *ServA* → *RP* → *ServWeb*
- d) La ida *ServWeb* → *RB* → *RP* → *ServA* y la vuelta *ServA* → *RS* → *RB* → *ServWeb*
- e) La ida *ServWeb* → *RB* → *RS* → *ServA* y la vuelta *ServA* → *RP* → *RB* → *ServWeb*
- f) La ida *ServWeb* → *RB* → *RP* → *ServA* y la vuelta *ServA* → *RS* → *RP* → *ServWeb*

Pregunta 7: (0.833 ptos) ¿Puede jugar el jefe desde PCdelJefe a pesar del problema de la pregunta 4? Razone la respuesta

Pregunta 8: (0.833 ptos) Un jugador externo utiliza un PC con dirección IP 86.21.3.2 máscara 255.255.255.0 y dirección MAC 00:88:33:22:11:00. Indique las direcciones IP y MAC del primer paquete que recibiría el ServidorB desde el ordenador del jugador externo así como las del primer paquete que recibiría el ServidorB desde PCdelJefe

Primer paquete de un jugador externo conectándose al servidorB

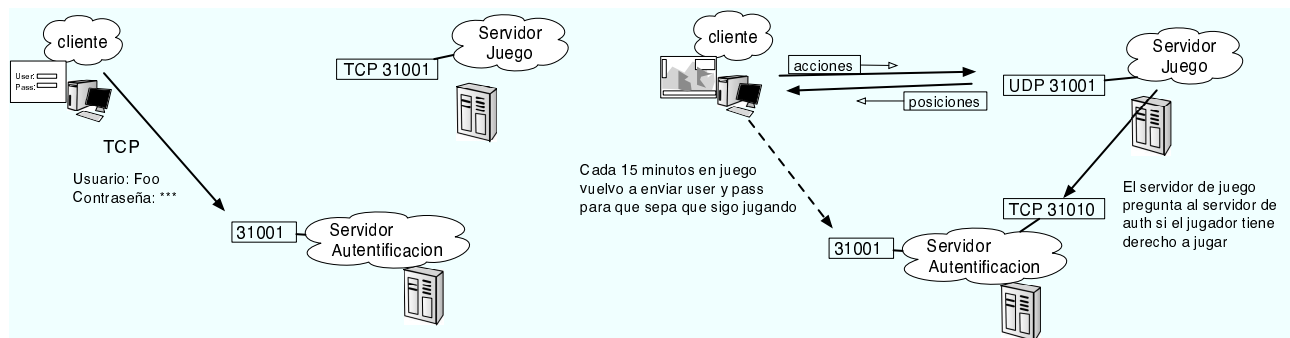
[Dirección MAC origen: _____ Dirección MAC destino: _____]

[Dirección IP origen: _____ Dirección IP destino: _____]

Primer paquete del jefe conectándose al servidorB

[Dirección MAC origen: _____ Dirección MAC destino: _____]

[Dirección IP origen: _____ Dirección IP destino: _____]



Las comunicaciones de tiempo real del servidor de juego con los clientes de los jugadores se utilizan UDP como se ve en la figura. El cliente del juego desde el ordenador del usuario envía mensajes con sus acciones y movimientos y al puerto 31001 del servidor y el servidor desde ese puerto envía el estado del juego y las acciones de los otros jugadores. Para la autenticación del usuario previamente al comienzo del juego el cliente remoto establece una conexión TCP con un servidor en el ordenador ServidorWeb que contiene la base de datos de los usuarios que tienen derecho a jugar y le envía su usuario y contraseña para verificación. Esta autenticación tiene validez por 20 minutos y cada 15 minutos el cliente renueva la autenticación si su usuario no ha salido del programa. Cuando el servidor del juego recibe comandos de un usuario se asegura de que ese usuario se ha autenticado preguntando al servidor de autenticación. Esta pregunta se realiza en una conexión TCP a un servidor de autenticación situado en el puerto 31010. Sobre esta conexión se emplea un protocolo de nivel de aplicación sencillo propio del juego. El cliente establece la conexión y envía un comando al servidor en texto con el siguiente formato.

```
CHECK <nombredejugador> <ipdelservidor> \n
```

Donde se indica el nombre del jugador y la dirección IP en formato a.b.c.d del servidor que pide la autenticación [podemos tener varios servidores con el mismo usuario] El servidor de autenticación debe comprobar si el usuario tiene permiso para jugar e indicar que si o que no con el mensaje

```
AUTH <nombredejugador> FROM <ipdelservidor> OK\n
AUTH <nombredejugador> FROM <ipdelservidor> ERR\n
```

Donde OK o ERR indica si el usuario está autenticado o no para jugar en ese servidor

Se le ha encomendado la realización de un servidor de autenticación que responda a las preguntas del juego según este protocolo.

Para ello se escribe la función main siguiente

```
int main(int argc, char *argv) {
    int s;
    printf("AuthTellGame... running\n");
    s=prepara_socket_tcp(31010);
    escucha_peticiones(s);
}
```

Y la función prepara_socket_tcp() construye el socket

```
int prepara_socket_tcp(int puerto) {
    struct sockaddr_in midireccion;
    int sock , err;

    sock = socket(PF_INET,SOCK_DGRAM,0);
    midireccion.sin_family = AF_INET;
    midireccion.sin_addr.s_addr = htonl(INADDR_ANY);
    midireccion.sin_port = htons(puerto);

    err = bind(sock, (struct sockaddr *)&midireccion, sizeof(midireccion));
    if ( err == -1 ) {
        exit(-1);
    }
    listen(sock,5);
    return sock;
}
```

Pregunta 9: (0.833 pts) ¿Qué errores se han cometido al escribir la función `prepara_socket_tcp()` ?

- a) Ninguno
- b) A la función `socket` hay que pasarle `SOCK_STREAM` en lugar de `SOCK_DGRAM`
- c) No hay que poner `INADDR_ANY` sino la dirección IP del servidor de juego
- d) Otro error (indicar cual) _____
- e) La instrucción `listen(sock,5);` hay que eliminarla dado que al tratarse de un socket UDP dará error

La función `escucha_peticiones` se mantiene a la escucha de peticiones del juego preguntando por la autenticación de diferentes jugadores

```
int escucha_peticiones(int s) {
    int socket_cliente;
    struct sockaddr_in d;
    unsigned int dlen;
    char buf[200], jugador[200], ip[200];
    FILE *f;

    while (1) {
        dlen=sizeof(d);
        socket_cliente=accept(s,(struct sockaddr *)&d,&dlen);
        f=fdopen(socket_cliente,"r+");
        if (fgets(buf,200,f) != NULL) {
            sscanf(buf,"CHECK %s %s",jugador,ip);

            if (esta_autenticado(jugador)) {
                fprintf(f,"AUTH %s FROM %s OK\n",jugador,ip);
            } else {
                fprintf(f,"AUTH %s FROM %s ERR\n",jugador,ip);
            }
        }
        fclose(f);
    }
}
```

Pregunta 10: (0.833 pts) ¿Como modificaría la función para que sólo se acepten preguntas de autenticación sobre una IP de servidor si la conexión proviene de esa dirección IP?

Pregunta 11: (0.833 ptos) Si un cliente esta autentificado y se envía el mensaje AUTH j1 FROM 12.3.4.25 OK hacia el servidor ¿en cual de estos paquetes viaja el mensaje?

- a) Un paquete TCP con puerto origen 31010
- b) Un paquete UDP con puerto origen 31010
- c) Un paquete TCP con puerto destino 31001
- d) Un paquete TCP con puerto origen 31001
- e) No hay suficiente información para saberlo ya que el mensaje se manda por el socket `socket_cliente` y no por el socket `s`

Pregunta 12: (0.833 ptos) Un usuario malicioso realiza una conexión desde su ordenador con direccion IP 133.2.1.12. Utilizando el comando `nc 12.3.4.51 31010` y una vez establecida la conexión la deja abierta sin enviar nada por ella. ¿Como reacciona nuestro programa? ¿Que efectos tiene en el juego?

- a) El servidor se queda esperando la pregunta. Esto bloquea que no se pueden atender posteriores preguntas sobre otros jugadores. Ningun jugador nuevo puede entrar a jugar hasta que se cierre la conexion y los que estan jugando iran cayendo conforme caduquen sus autentificaciones.
- b) El servidor se queda bloqueado esperando la pregunta, cuando pasa un tiempo sin recibir nada la función `fgets()` devuelve el control al programa devolviendo NULL, eso hace que puedan seguir atendiéndose peticiones aunque por unos segundos las entradas de nuevos jugadores están bloqueadas porque el servidor no es concurrente tal y como esta escrito
- c) El proceso que atiende la peticion de ese cliente se queda bloqueado esperando que el `fgets()` devuelva el control. Pero eso no afecta al proceso que escucha el socket del servidor que sigue atendiendo peticiones. La entrada de jugadores no se ve afectada
- d) La peticion no proviene de la dirección IP de ningún servidor así que el `accept()` no acepta esa conexión