Solución Examen - 1 de agosto de 2008 (ref: sirc0807.doc)

Instrucciones

- **Indique su nombre completo y número de cédula en cada hoja.**
- **Numere todas las hojas e indique la cantidad total de hojas que entrega en la primera.**
- **Escriba las hojas de un solo lado.**
- **Comience cada ejercicio en una hoja nueva.**
- **Sólo se contestarán dudas de letra. No se aceptarán dudas de ningún tipo los últimos 30 minutos del examen.**
- **El examen es individual y sin material.**
- **Es obligatorio responder correctamente al menos 15 puntos en las preguntas teóricas.**
- **El puntaje mínimo de aprobación es de 60 puntos.**
- **Para todos los ejercicios, si es necesario, puede suponer que dispone de los tipos de datos básicos (p.ej. lista, cola, archivo, string, etc.) y sus funciones asociadas (ej: tail(lista), crear(archivo), concatenar(string, string).**
- **Duración: 3 horas.**

Preguntas Teóricas

Pregunta 1 (5 puntos)

Defina el concepto IDS. Reseñe y compare el principio de funcionamiento de los dos grandes tipos de dichos sistemas que existen.

Pregunta 2 (10 puntos)

Describa el mecanismo de control de congestión de TCP partiendo de explicar los principios básicos de operación de al menos dos de los cuatro algoritmos básicos involucrados.

Pregunta 3 (10 puntos)

Explique el protocolo de acceso múltiple con evitado de colisiones (CSMA/CA) utilizado en las redes inalámbricas IEEE 802.11 (Wi-Fi) trabajando en modo infraestructura.

Pregunta 4 (10 puntos)

Explique la diferencia entre plano de datos (o forwarding) y plano de control. Mencione un ejemplo de protocolo de datos con control fuera de banda (con protocolo de control asociado) y otro con control dentro de banda (sin protocolo de control asociado).

Pregunta 5 (5 puntos)

Mencione 3 ventajas y 2 desventajas del protocolo IPv6.

Problemas Prácticos

Problema 1 (30 puntos)

Una aplicación de transmisión de audio por Internet funciona emitiendo datos de forma redundante desde tres fuentes (emisores). Los datos se transmiten utilizando RTP sobre UDP.

Payload type (7 bits)	Sequence number (16 bits)	Timestamp (32 bits)	Synchronization Source Identifier (32 bits)	Miscellaneous Fields (variable)

Cabezal RTP

Todos los paquetes que se envían desde cualquiera de los tres emisores son generados a una tasa constante, son de la misma duración, y están sincronizados. Puede considerarse que exactamente es el mismo flujo de audio replicado desde tres fuentes diferentes.

Se desea implementar un buffer de recepción que haga uso de la característica redundante del streaming. Se cuenta con una estructura de buffer que funciona como una cola circular donde se pueden almacenar hasta N paquetes UDP. Para manejarlo se cuenta con las siguientes primitivas ya implementadas:

- **`lockBuffer()`**
- **`releaseBuffer()`**
- **`escribirPaqueteBuffer(indice: int, PaqueteUDP: paq)`**
- **`leerPaqueteBuffer(indice: int): PaqueteUDP`**
- **`borrarPaqueteBuffer(indice: int)`**

Debido a las características del buffer circular, es posible que en algunos casos un paquete llegue demasiado rápido y su lugar en el buffer esté ocupado por un paquete anterior cuyo instante de reproducción aún no ha llegado. Resulta de interés llevar un registro de cuántos paquetes han llegado en esta situación, distinguiendo la cantidad para cada una de las fuentes.

En el nodo receptor existe un proceso **`reproductor`** (ya implementado) que realiza una iteración con la siguiente lógica:

```
Lee el paquete correspondiente al instante de tiempo del stream que debe reproducir
Elimina este paquete del buffer
Reproduce el paquete
Aumenta en una unidad el instante de tiempo que debe reproducir
```

En caso de que no se encuentre el paquete que debe reproducir, simplemente se mantendrá en silencio durante el tiempo correspondiente y luego continuará con la iteración. Este proceso realiza el correspondiente bloqueo de la región crítica del buffer. Se puede consultar cuál es el instante de tiempo que se está reproduciendo actualmente mediante la función: **`getPlayingTimestamp(): int`**

Se propone implementar un proceso **`escucha`** que reciba los paquetes UDP que lleguen, los agregue al buffer en sus posiciones respectivas, y calcule las estadísticas correspondientes.

Se pide:

a) Indicar qué campos de RTP puede utilizar en los siguientes casos:
   i. Para que el proceso **`escucha`** decida en qué posiciones almacenar los paquetes en el buffer, de forma que queden en el orden en que el proceso reproductor los tomará para reproducirlos.
   ii. Para que el proceso **`escucha`** distinga de cuál de los emisores se está enviando el paquete, y poder así llevar el registro de los que llegaron de cada emisor.

b) Implemente en un lenguaje de alto nivel el proceso **`escucha`**.

Solución

Parte a)

i. Para saber en qué posición almacenar un paquete que llegue, se puede utilizar el *número de secuencia* de RTP, alcanzaría con hacer número de secuencia módulo N. Como los paquetes son de una duración fija también puede utilizarse el *timestamp*, el índice del paquete con *timestamp* T sería (T/duración_de_paquete) módulo N.

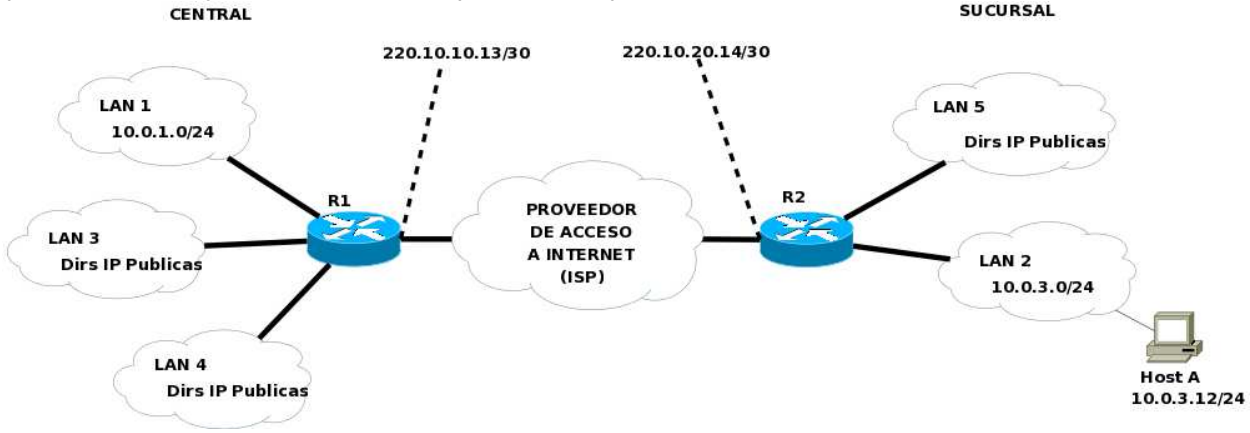
ii. Para distinguir de cuál de los tres emisores proviene el paquete se puede utilizar el campo SSRC (*synchronization source identifier*) que será único para cada una de las réplicas del stream.

Parte b)

```
map<int, int> muy_rapido; // mapping de <id de fuente, cantidad de paquetes>
Socket s = crear_socket_udp();
while (true) {
        DatagramaUDP p = receive(s);
        PaqueteRTP r = p.getPayload();
        int t = getPlayingTimestamp();
        if (t < r.getTimestamp()) {
                // todavía estamos a tiempo para reproducirlo
                int pos = r.getNumSeq() % N;
                lockBuffer();
                PaqueteRTP q = leerPaqueteBuffer(pos);
                if (q == null) {
                        // no estaba, lo agregamos
                        escribirPaqueteBuffer(pos, r);
                } else if (q.getNumSeq() > r.getNumSeq()) {
                        // es un paquete que llegó demasiado pronto
                        muy_rapido[r.getSsrc()]++;
                }
                releaseBuffer();
        }
}
```

Problema 2 (30 puntos)

La figura muestra la red corporativa de una empresa con una sede central y una sucursal. A los interfaces WAN de los routers R1 y R2, el ISP les asigna las direcciones públicas indicadas en la figura (220.10.10.13/30 y 220.10.20.14/30 respectivamente).



Las subredes LAN1 y LAN2 tienen direccionamiento privado y permanecerán sin cambios. Para expandir la red corporativa se ha adquirido el prefijo 147.10.8.0/21, con el cual se numerarán las subredes LAN3, LAN4 y LAN5, con perspectivas de seguir agregando subredes en el futuro. Los dos routers R1 y R2 tienen capacidad para realizar tunneling (IP en IP) y NAT.
Se pide:

Parte a)

i. Realice un plan de numeración de las subredes LAN3, 4 y 5 utilizando el prefijo de dirs. IP públicas mencionado anteriormente, asumiendo que no se conectan más de 120 hosts por subred. ¿Cuántas subredes se pueden numerar de acuerdo a su plan de numeración?

ii. ¿Para qué utilizaría el NAT en esta arquitectura de red? Justifique brevemente su respuesta. Indicar las direcciones IP de origen/destino si se envía una petición de conexión a un servidor Web cuya dir. IP es 220.12.20.156, desde el Host A en LAN2, para los paquetes en LAN2, y para los paquetes en la red del ISP.

Parte b)

i. Los administradores de la red corporativa desean que exista conectividad completa entre todos los *end systems* de la empresa y de éstos con Internet; ¿cómo se podría utilizar *tunneling* con este propósito? Justifique brevemente su respuesta.

ii. El protocolo de enrutamiento que utilizan los routers de la red corporativa (Sede central y sucursal) es RIP. Mostrar una posible tabla de enrutamiento del router R2, que permita conectividad completa entre todos los *end systems* de la empresa y de éstos con Internet. Utilizar el formato que se indica, asumiendo que las interfaces de túnel se llaman *tunel_R1* y *tunel_R2*, y que siempre se utiliza la dirección más baja para el gateway en cada subred.

Adquisición	Red destino / Máscara	Gateway

En la columna Adquisición utilizar la notación que sigue:
C: para las entradas correspondientes a las redes a las que está directamente conectado.
R: para las entradas que son anunciadas vía RIP.
O: para las entradas que son anunciadas por el ISP.

Solución

Parte a)

i. Se dispone del prefijo 147.10.8.0/21; la solución no es única.
Para cada subred se necesitan como máximo 120 + 1 + 2 = 123 direcciones IP por LAN.

Una solución posible:
LAN 3 – 147.10.8.0/25
LAN 4 – 147.10.8.128/25
LAN 5 – 147.10.9.0/25
Ésta elección permite que el ISP y la sucursal de la empresa *sumaricen* las subredes LAN3 y LAN4 (a través de R1) como un solo prefijo /24: 147.10.8.0/24.

Otra solución posible:
LAN 3 – 147.10.8.0/25
LAN 4 – 147.10.9.0/25
LAN 5 – 147.10.10.0/25
Ésta elección implica que la tabla de enrutamiento de R2 tenga una entrada más respecto a la solución anterior (no se puede sumarizar).

Según este plan de numeración se cuenta con 2^4 = 16 subredes /25:
147.10.8.0/25, 147.10.8.128/25
147.10.9.0/25, 147.10.9.128/25
147.10.10.0/25, 147.10.10.128/25
147.10.11.0/25, 147.10.11.128/25
147.10.12.0/25, 147.10.12.128/25
147.10.13.0/25, 147.10.13.128/25,
147.10.14.0/25, 147.10.14.128/25,
147.10.15.0/25, 147.10.15.128/25.

ii. Utilizaría NAT para permitir conectividad desde las LANs con numeración privada hacia la red del ISP. El motivo de ésto es que, el ISP en particular e Internet en general, filtran el forwarding de paquetes que tengan direcciones IP origen y/o destino pertenecientes al rango de redes privadas: 10.0.0.0/8, 172.16.0.0/12 y 192.168.0.0/16.

Para paquetes en LAN2
IPOrigen: 10.0.3.12
IPDestino: 220.12.20.156

Para paquetes en la red del ISP
IPOrigen: 220.10.20.14
IPDestino: 220.12.20.156

Parte b)

i. El tunneling permite interconectar las subredes de direccionamiento privado entre sí y las de direccionamiento público (tanto de Internet como de la empresa) con las de privado para subredes en distintas sucursales.

ii.

Adquisición	Red destino / Máscara	Gateway
C	10.0.3.0 / 255.255.255.0	10.0.3.1
C	147.10.9.0 / 255.255.255.128	147.10.10.1
R	10.0.1.0 / 255.255.255.0	tunel_R2
R	147.10.8.0 / 255.255.255.0	tunel_R2
O	0.0.0.0 / 0	220.10.20.13