

**Examen de Introducción a las Redes de Computadoras  
y Comunicación de Datos  
(ref: sirc0707.doc)  
31 de julio de 2007**

**Atención:**

- La duración del examen de 3 horas.
- El examen debe realizarse sin material.
- Se debe responder al menos el equivalente a 15 puntos en las preguntas teóricas.
- El puntaje mínimo de aprobación es de 60 puntos.
- para todos los ejercicios, suponga 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)).

**Preguntas Teóricas**

**Pregunta 1 (10 puntos)**

- a) Si se quiere calcular la máxima capacidad de transmisión de datos sobre una línea telefónica, que teorema visto en el curso se debe aplicar, justifique y escriba la fórmula para calcularla, explicando el significado de las variables en la fórmula y que unidades se suelen utilizar.
- b) Suponga que se tiene un modem para enviar datos a través de la línea telefónica. ¿Que tipos de modulación conoce?. Explique como se representan los diferentes tipos de modulación.

**Pregunta 2 (5 puntos)**

Nombre los mecanismos de detección y corrección de errores más comunes en la capa enlace, y describa uno.

**Pregunta 3 (10 puntos)**

- a) Indique cual es la diferencia de performance entre ALOHA y Slotted-ALOHA en el caso de carga normal. Justifique.
- b) Que tipos de protocolos CSMA hay. Describa cual es la diferencia entre ellos.

**Pregunta 4 (10 puntos)**

Explique que función cumple el DNS (Domain Name System), y presente el protocolo involucrado brevemente.

**Pregunta 5 (5 puntos)**

Especifique de manera concisa la arquitectura y el funcionamiento de IP Móvil.

## Problemas Prácticos

### **Problema 1 (35 puntos)**

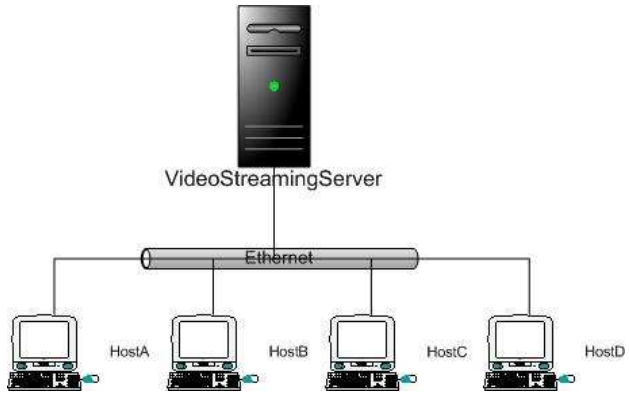
Sea una red en la que los equipos Host buscan recibir diferentes servicios de tipo *video streaming* (señales de TV), difundidas a través de un servidor de la red (VideoStreamingServer).

Todos los equipos se encuentran conectados a través de un segmento ethernet, al cual se encuentra conectado también el servidor.

Actualmente el problema planteado se resuelve de la siguiente forma. Dado un Host, que desea recibir una señal de TV específica, realiza los siguientes pasos:

- El equipo Host, envía por UDP/puerto VS\_SERVER, una solicitud *request* al VideoStreamingServer (equipo conocido), solicitando recibir la lista de señales disponibles.
- El equipo VideoStreamingServer, contesta enviando un *response* por UDP/ puerto VS\_CLIENT, un mensaje que contiene las señales de TV disponibles.
- El equipo Host, envía por UDP/puerto VS\_SERVER, una solicitud *join* al VideoStreamingServer, indicando la señal que desea recibir, y en que puerto espera escucharlo SIGNAL\_PORT.
- El equipo Host, comienza a escuchar por UDP/puerto SIGNAL\_PROT, la señal deseada.
- El equipo Host, cada KEEP\_ALIVE segundos, envía un *confirm*, por UDP/puerto VS\_SERVER, incluyendo las señales que se encuentra escuchando.
- El equipo Host, envía por UDP/puerto VS\_SERVER, una solicitud *leave* al VideoStreamingServer, cuando desee dejar de recibir alguna señal, indicando su identificación.

El VideoStreamingServer, en caso de no recibir 5 *confirm* consecutivos, interrumpirá la transmisión de las señales solicitadas por éste Host.



Se pide:

- a) La solución existente, ¿que ineficiencias tiene, en referencia al uso del canal?. Estudiar en particular el caso en que más de un equipo solicitan la misma señal.
- b) ¿Que ventajas se logran implementando un servicio similar sobre multicast?
- c) Modificar el protocolo para que ejecute sobre multicast. Especificar, lo ejecutado por el servidor (VideoStreamingServer), en un lenguaje de alto nivel. Especificar en especial los tipos de paquetes requeridos. Considerar que el servidor solamente transmite al medio las señales de TV, siempre que algún Host se encuentre escuchando la misma.

Se cuenta además con:

- una dirección multicast 226.0.0.1 para ser utilizada por el servicio y direcciones adicionales multicast pre-definidas, por señal de TV disponible.
- y las primitivas:

```
transmitir(videoID)
suspender(videoID),
```

que permiten iniciar o finalizar la transmisión del videoID especificado como parámetro, por la dirección multicast pre-definida para él mismo.

#### **Nota:**

Se pide exclusivamente resolver el protocolo que permite el registro de los diferentes Host, para ver señales de TV.

## Solución:

- a) La solución existente, ¿que ineficiencias tiene, en referencia al uso del canal?. Estudiar en particular el caso en que más de un equipo solicitan la misma señal.

Principalmente la ineficiencia radica en el uso de transmisiones *unicast*. Dado el caso que en una LAN, más de un equipo solicite la misma señal, para cada equipo que ha solicitado la misma se generará un flujo de paquetes para satisfacer los requerimientos de cada host. Estos flujos transmiten exactamente la misma señal, son paquetes idénticos, cuya diferencia se encuentra en el campo destino del cabezal.

Resumiendo, si muchos usuarios en la misma LAN solicitaran la misma señal, esto podría saturar fácilmente la red.

- b) ¿Que ventajas se logran implementando un servicio similar sobre multicast?

Realizando una implementación del protocolo sobre *multicast*, se enviará un solo flujo por señal que desee publicarse en la red. Todos los equipos que deseen recibirla, solamente tomarán los paquetes del flujo, identificados con una dirección *múlticast* destino, de deberá resolverse su intercambio con el host.

- c) Modificar el protocolo para que ejecute sobre multicast. Especificar, lo ejecutado por el servidor (VideoStreamingServer), en un lenguaje de alto nivel. Especificar en especial los tipos de paquetes requeridos. Considerar que el servidor solamente transmite al medio las señales de TV, siempre que algún Host se encuentre escuchando la misma.

Se cuenta además con:

- una dirección multicast 226.0.0.1 para ser utilizada por el servicio y direcciones adicionales multicast pre-definidas, por señal de TV disponible.
- y las primitivas:

```
transmitir(videoID)
suspender(videoID),
```

que permiten iniciar o finalizar la transmisión del videoID especificado como parámetro, por la dirección multicast pre-definida para él mismo.

Tipos de paquete:

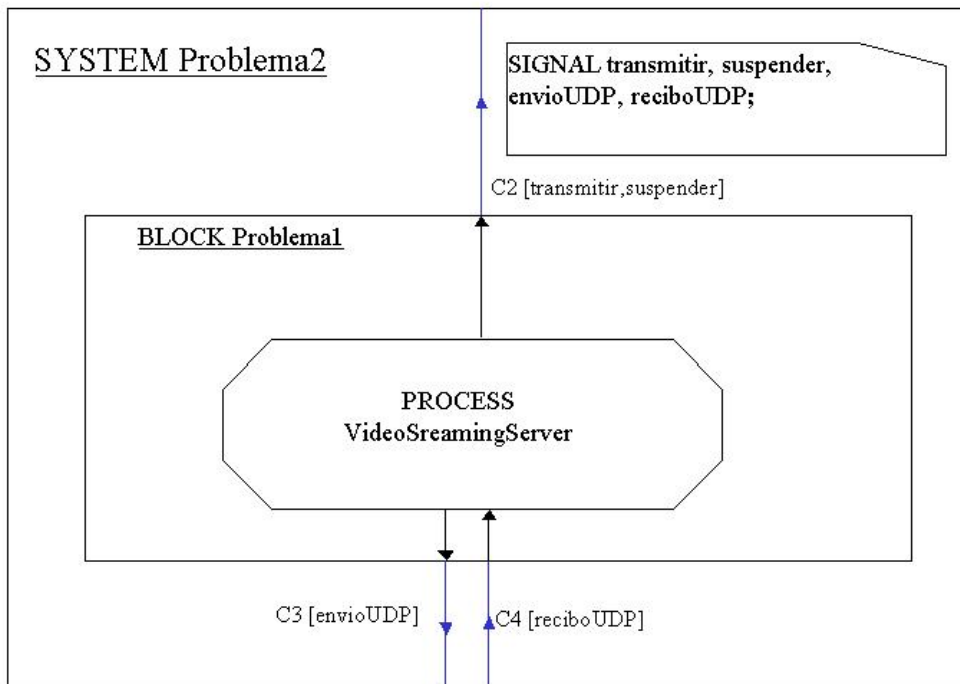
REQUEST\_SOL – para solicitar la lista de señales disponibles  
REQUEST\_RESP – para responder la lista de señales disponibles  
JOIN – para solicitar la recepción una señal  
CONFIRM – indicando señales que se están recibiendo  
LEAVE – indicando señal que no desea seguir recibiendo

Solución propuesta:

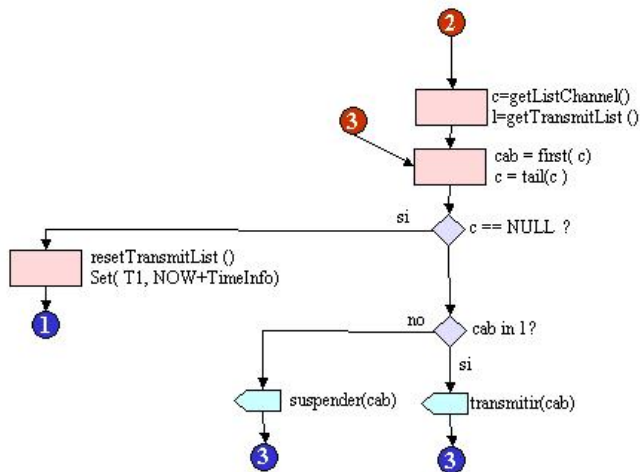
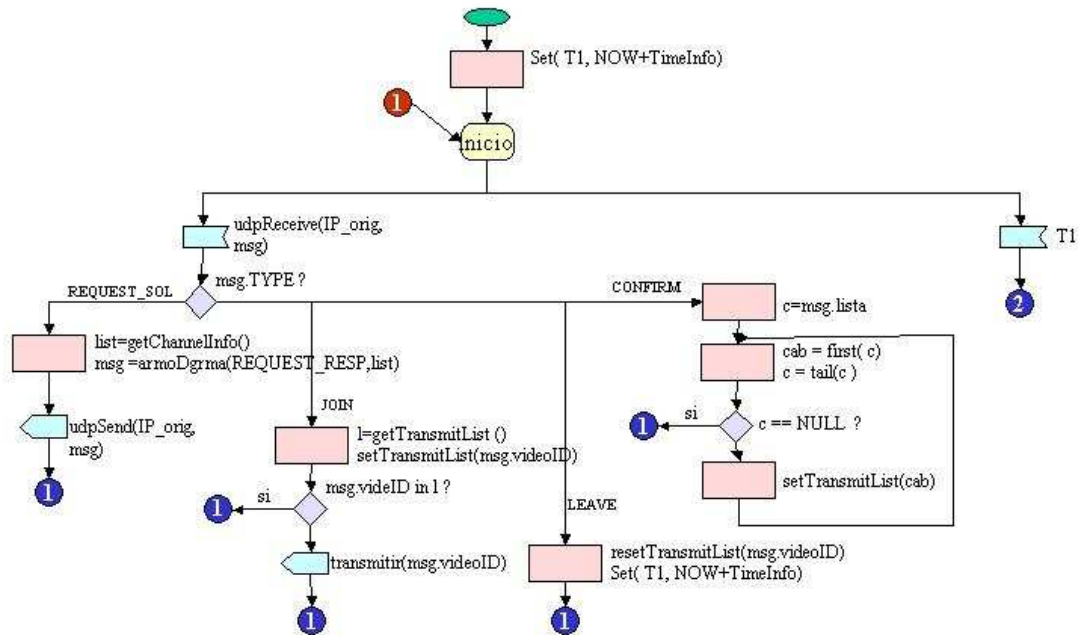
- Los equipos solicitarán la lista de señales a través de los paquetes REQUEST\_SOL (dir. 226.0.0.1/port VS\_SERVER) y REQUEST\_RESP (dir IP\_CLIENTE/ port VS/CLIENT)
- Cuando un equipo desee recibir una señal, realizará un JOIN (dir. 226.0.0.1/port VS\_SERVER), indicando el videoID que desea recibir.
- Cada un tiempo menor que T1, se enviará un CONFIRM con la lista de señales que se están recibiendo (dir. 226.0.0.1/port VS\_SERVER).
- En caso de no desear la recepción de alguna señal, puede hacerse enviando un CONFIRM sin la señal en la lista, o enviando un LEAVE (dir. 226.0.0.1/port VS\_SERVER) indicando que no se desea recibir la señal.

**Formato de los paquetes:**

```
paquete{
  TYPE = [REQUEST_SOL, REQUEST_RESP, JOIN, CONFIRM, LEAVE]
  info_adicional {
    case TYPE of
      REQUEST_SOL: {};
      REQUEST_RESP: {
        list of {
          videoID
          MCAST_IP
          MCAST_PORT
        }
      };
      JOIN: {
        videoID
      };
      LEAVE {
        videoID
      };
      CONFIRM: {
        list of {
          videoID
        }
      };
    endcase
  }
}
```



# Process VideoStreamingServer



### **Funciones auxiliares:**

`list getChanelInfo()` – devuelve una lista conteniendo, los campos `videoID`, `MCAST_IP` dirección IP multicas utilizada para la transmisión, `MCAST_PORT` puerto destino de la transmisión.

`dgram armoDgrama(tipo,payload)` – devuelve el payload de un datagrama UDP, con tipo pasado por parámetro y payload según el tipo.

`list getTransmitList()` –devuelve la lista de `videoID` de señales que se están transmitiendo por la red.

`void resetTransmitList()` – setea todas las señales como no solicitadas por ningún equipo de la red. Esto permite que en el próximo poleo, si recibió un CONFIRM volverá a estar como transmitida.

`list getTransmitList()` – devuelve lista de `videoID` de señales solicitadas por equipos en la red, que se encuentran en transmisión.

`void setTransmitList(videoID)` – setea la señal `videoID` como solicitada por un equipo.

`void resetTransmitList(videoID)` – setea la señal `videoID` como no solicitada por un equipo.

`list getListChanel()` – devuelve una lista conteniendo, los `videoID` de las señales disponibles.

## Problema 2 (25 puntos)

Una empresa desea conectar los PC y servidores existentes en las oficinas de Colonia y Maldonado con la oficina Central en Montevideo (Figura 1). Las oficinas de Colonia y Maldonado tienen 20 nodos (PC y servidores) cada una y la oficina de Montevideo tiene 63 nodos (PC y servidores) a interconectar.

Se pide:

- Asignar las direcciones IP a cada subred (dirección de red y máscara para las subredes A,B,C,D,E) utilizando la menor cantidad de direcciones IP posibles de la red 192.168.1.0/24 (192.168.1.0/255.255.255.0)

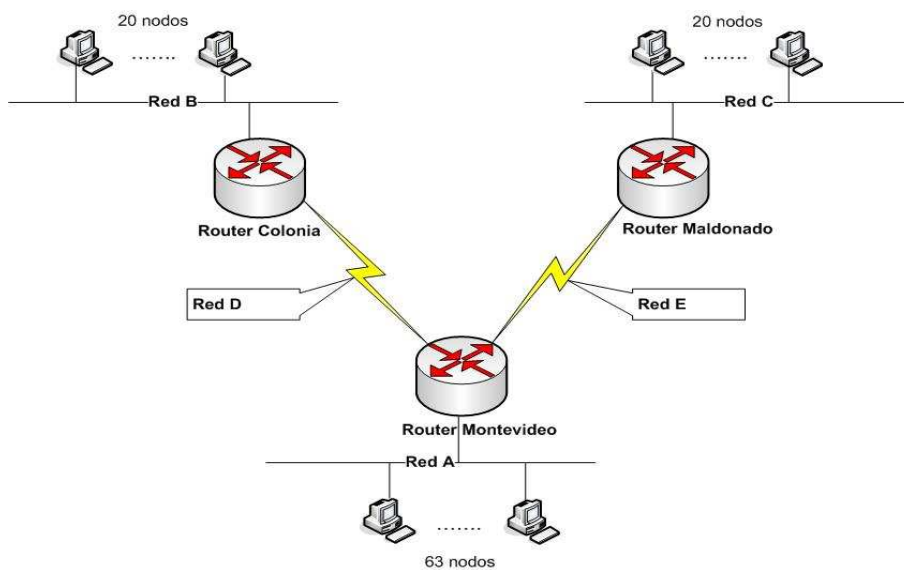


Figure 1 Esquema de red, de la empresa

- Indique la configuración de rutas a realizar en cada equipo de la red, para asegurar conectividad total.

### Nota:

A,B y C son redes LAN. D y E son redes WAN (punto a punto).

### Solución:

- Asignar las direcciones IP a cada subred (dirección de red y máscara para las subredes A,B,C,D,E) utilizando la menor cantidad de direcciones IP posibles de la red 192.168.1.0/24 (192.168.1.0/255.255.255.0)

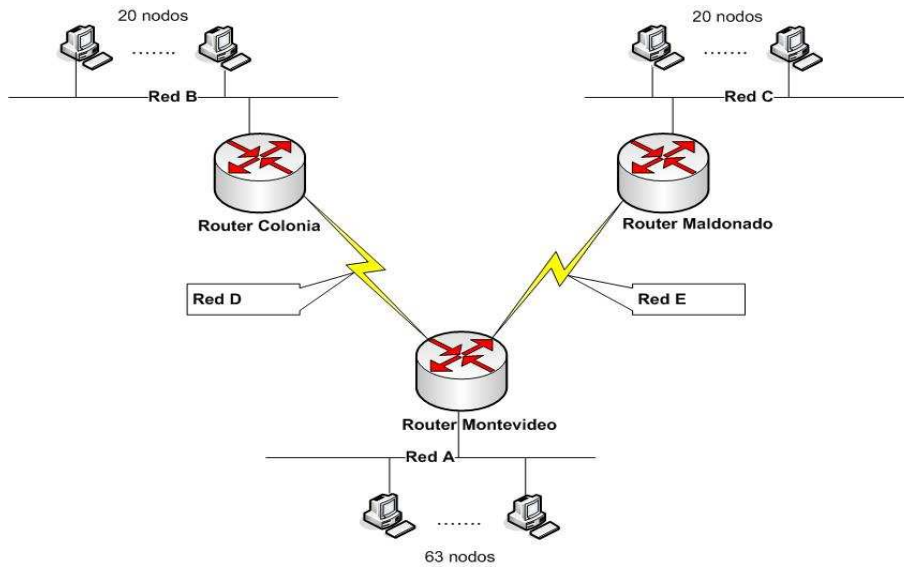


Figure 2 Esquema de red, de la empresa

Conteo de cantidad de direcciones IP requeridas por RED:

Subred	Cantidad Nodos	Cant. Dir. Red y Broadcast	Total de IP's requeridas	Mascara de Red
A	63	2	65	/25
B	20	2	22	/27
C	20	2	22	/27
D	2	2	4	/30
E	2	2	4	/30

Asignación de direcciones IP:

/24	/25	/26	/27	/28	/29	/30	Sub Red asignada
192.168.1.0/24	192.168.1.0/25	..					A
	192.168.1.128/25	..	192.168.1.128/27				B
		..	192.168.1.160/27				C
		..	192.168.1.192/27	..	..	192.168.1.192/30	D
		..		..	..	192.168.1.196/30	E

b) Indique la configuración de rutas a realizar en cada equipo de la red, para asegurar conectividad total.

Previamente, debo asignar las direcciones IP's en los routers:

Router	interfaz	IP asignada
Colonia	Red D	192.168.1.194
Colonia	Red B	192.168.1.129
Montevideo	Red D	192.168.1.193
Montevideo	Red E	192.168.1.197



Montevideo	Red A	192.168.1.1
Maldonado	Red E	192.168.1.198
Maldonado	Red C	192.168.1.161

### Configuración de los routers

Router Colonia:

Subred	Máscara	Next Hop
192.168.1.128	/27	Directamente conectado
192.168.1.192	/30	Directamente conectado
Default gateway		192.168.1.193

Router Montevideo:

Subred	Máscara	Next Hop
192.168.1.0	/25	Directamente conectado
192.168.1.192	/30	Directamente conectado
192.168.1.196	/30	Directamente conectado
192.168.1.128	/27	192.168.1.194
192.168.1.160	/27	192.168.1.198

Router Maldonado:

Subred	Máscara	Next Hop
192.168.1.160	/27	Directamente conectado
192.168.1.196	/30	Directamente conectado
Default gateway		192.168.1.197

### Configuración de los hosts

Red A:

Subred	Máscara	Next Hop
192.168.1.0	/25	Directamente conectado
Default gateway		192.168.1.1

Red B:

Subred	Máscara	Next Hop
192.168.1.128	/27	Directamente conectado
Default gateway		192.168.1.129

Red C:

Subred	Máscara	Next Hop
192.168.1.160	/27	Directamente conectado
Default gateway		192.168.1.161