

Examen de Introducción a las Redes de Computadoras y Comunicación de Datos (ref: sirc0303.doc) 19 de marzo de 2003

Atención: 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)*).

Ejercicio 1

Se dispone de un enlace WAN de 64 kbps de capacidad para interconectar dos redes LAN de alta velocidad. Para hacer frente a esta diferencia de velocidades entre enlaces, los routers de los extremos del enlace deben encolar paquetes antes de transmitirlos.

Cada router posee dos colas diferentes de paquetes, una de mas alta prioridad que la otra. Los routers saben como manejar paquetes IP fragmentados.

Hay dos tipos de tráfico que se desean considerar :

- Tipo A: trafico sensible al retardo. El retardo en la línea no debe superar los 20 ms. Los paquetes que genera este protocolo son siempre pequeños. Podría ser, por ejemplo, tráfico de voz sobre IP.
- Tipo B: trafico de gran ancho de banda, pero no sensible al retardo. Este trafico genera paquetes potencialmente grandes. Podría ser, por ejemplo, tráfico HTTP o FTP.

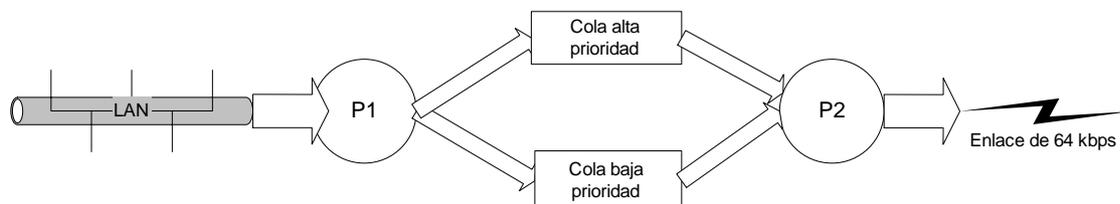
En cada router hay dos procesos que interesan considerar:

- Proceso P1 : este proceso recibe los paquetes provenientes de la interfaz LAN y los guarda en alguna de las dos colas disponibles. El proceso puede optar por colocar el paquete entero o por fragmentarlo y colocar el mismo fragmentado.
- Proceso P2 : este proceso toma paquetes de las colas y los transmite por la interfaz WAN. Este proceso debe servir la cola de alta prioridad siempre que esta tenga paquetes, y solamente en caso de que esta esté vacía, servirá paquetes de la cola de baja prioridad.

Se dispone de una función que devuelve el numero de paquetes que están esperando ser servidos en una cola, *getNPKets(cola)*. También se dispone de una función que es capaz de fragmentar paquetes IP en fragmentos de un cierto tamaño, generando también los encabezados de los fragmentos, *fragmentPacket(paquete, bytes)*.

Se pide:

- a) Determinar el largo máximo que pueden tener los paquetes a transmitir por la línea WAN para satisfacer los requerimientos de retardo del trafico de tipo A.
- b) Especificar en un lenguaje de alto nivel el proceso P1
- c) Especificar en un lenguaje de alto nivel el proceso P2



Ejercicio 2

Se cuenta con un Servidor de Nombres de Dominio (DNS), el que brinda servicios para la zona *dyndns.com.uy*.

Se desea lograr un servicio de nombres de dominio (DNS), que permita responder consultas sobre equipos que se encuentran sobre servicios de red que proporcionan IP dinámica (que varía con el tiempo). La idea es que los diferentes equipos que atienden los servicios en la red, ante la consulta respondan la IP actual de los mismos.

Para esto se cuenta en el servidor de la siguiente primitiva:

```
updatedns(dominio: string, tipo: [A,NS,MX,CNAME],IP : IP_address);
```

que actualiza la respuesta para el nombre *dominio*, para los registros tipo especificado (A address, NS Name Server, MX Mail Exchanger, CNAME Canonical name), a la *IP* especificada.

Se pide:

- Especificar en un lenguaje de alto nivel, los procesos que deben ejecutarse en el equipo con IP dinámica y en el servidor de DNS.
- ¿Que consideraciones deben realizarse sobre el tiempo de vida de los registros, debe cumplir el servidor de dominios (DNS) para que el funcionamiento sea correcto?

Solución:

a)

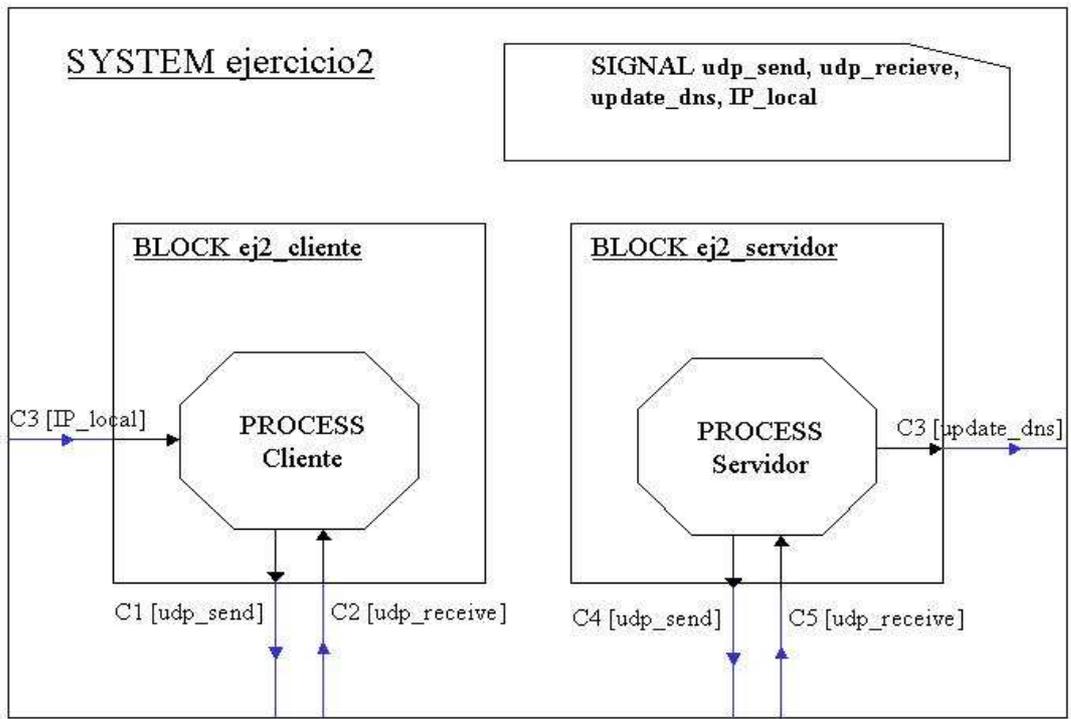
La solución, en términos generales tiene los siguientes lineamientos,

- Supongo la existencia de una señal que indica cuando la IP ha cambiado, así como una primitiva *local_IP*, que devuelve la IP actual de la máquina.
- Supongo además la existencia de una tabla que contiene todos los registros registrados en el DNS. La tabla tiene el siguiente formato.

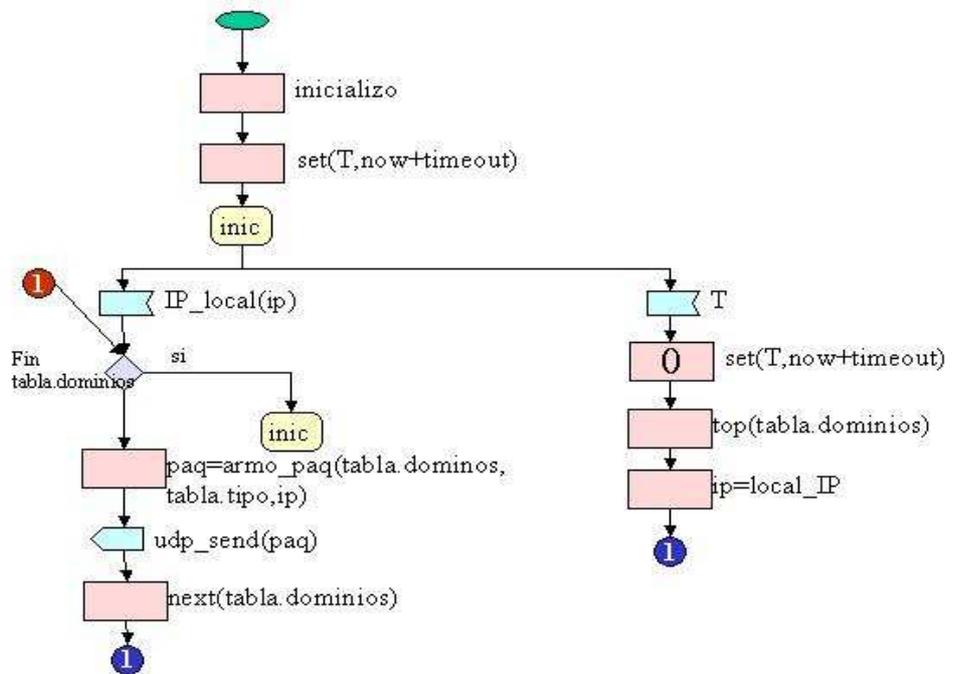
```
tabla array [1..MAX_REG] of {
    dominios: string;
    tipo: [A,MX,NS,CNAME];
}
```

La solución por parte del cliente, recibe una señal *IP_local*, que informa la nueva dirección IP de la máquina. Cuando se recibe esta señal se envía al servidor un paquete UDP, que contiene por cada registro en la tabla de dominios del cliente, una actualización. Además cada un TimeOut, el cliente envía una actualización de todos los registros.

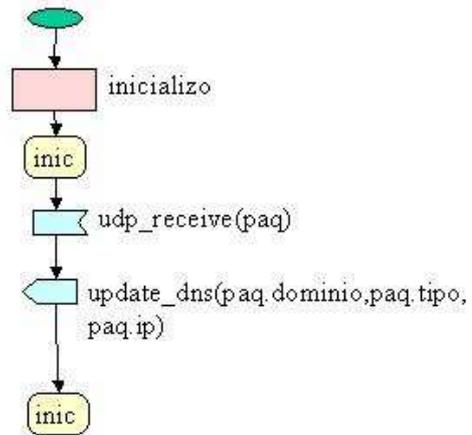
El servidor por cada actualización que recibe, invoca una actualización del DNS, lo que permitirá que conteste la IP correcta de los dominios.



PROCESS Cliente



PROCESS Servidor



b)

Se debe buscar que la transferencia de la información entre servidores DNS, tenga un tiempo de vida lo más corto posible, ya que de lo contrario, los servidores que hayan consultado al servidor del dominio *dyndns.com.uy*, mantendrán cacheada esta información contestando información equivocada.

Ejercicio 3

Se considera un equipo en el que es necesario obtener (transferir), todos los días a la misma hora, un archivo de gran tamaño. El nombre del archivo es siempre el mismo, y (el archivo) siempre tiene el mismo largo.

Existen diez servidores, siempre los mismos, que contienen copias idénticas actualizadas del archivo.

En lugar de utilizar ftp, se desea aprovechar que, en todos esos servidores, existe un servicio que atiende el port "x" UDP, que acepta datagramas conteniendo un número de port, un número de origen y un largo. Al recibir dicho requerimiento, los servicios envían un datagrama al port indicado del equipo solicitante, conteniendo el bloque de datos del archivo en cuestión que comienza en el origen indicado y tiene el largo solicitado.

Los enlaces entre el cliente y los servidores experimentan variaciones frecuentes en su capacidad de tráfico. Debido a ello, se requiere distribuir la transferencia del archivo entre los diez servidores. La distribución debe ser (aproximadamente) proporcional al tiempo de respuesta observado en los requerimientos a cada servidor, el que se re-evaluará periódicamente.

Se pide:

 Especificar en un lenguaje de alto nivel, el proceso que ejecuta en el cliente.