

# Examen de Introducción a las Redes de Computadoras y Comunicación de Datos (ref: eirc0508.doc) 5 de agosto de 2005

**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

Sea un router que almacena las siguientes estructuras de datos internas, con el fin de cumplir el objetivo de ruteo:

```
// lista de interfaces que presenta el equipo
interface_list list of {
    if interface; // identificador de la interfaz
    network IP_address;
    netmask IP_address;
    cost:integer; //costo asignado a la salida por esta interfaz
}
// Estructura que recopila información con el fin de armar posteriormente
// la tabla de ruteo (table_route)
topology list of {
    if interface;
    network IP_address;
    netmask IP_address;
    distancia_local integer;
    distancia_adyacente integer;
}
// estructura que consulta el router para realizar el forwarding
table_route list of {
    if interface;
    network IP_address;
    netmask IP_address;
    distancia integer;
    tipo [active|backup]; // indica tipo de ruta
                        // active esta siendo utilizada
                        // backup se puede utilizar en caso de falla
}
}
```

Se encuentra implementado sobre estas estructuras un protocolo de ruteo que presenta las siguientes características:

- Inicialmente el protocolo conoce todas las redes que tiene conectadas (definidas en las interfaces).
- El protocolo presenta los siguiente mensajes

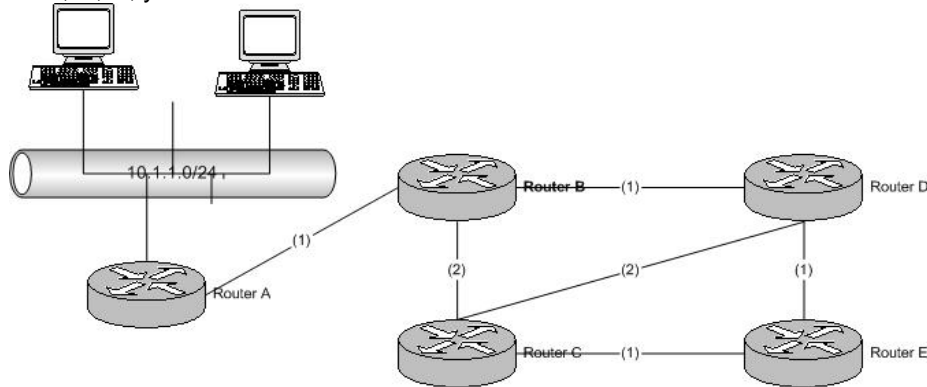
hello	Informa que el protocolo está activo
update	Envía a router adyacente las líneas activas de su tabla de ruteo, con excepción de las que rutea por la interfaz hacia donde lo envía.
inform	Informa a router adyacente caída de una interfaz, indicando las redes que quedaron inaccesibles

- Cada nodo pasa su tabla de ruteo (*table\_route*) a los nodos adyacentes, con la única condición que haya recibido un *hello* recientemente, por la interfaz donde está conectado . Está será incluida en la tabla de topología (*topology*), colocando en *distancia\_adyacente* la distancia indicada por el nodo, y en *distancia\_local* guardando la suma de la distancia indicada por el adyacente más el costo de la interfaz.
- La tabla de ruteo (*table\_route*), se genera a partir de la tabla de topología (*topology*), según el siguiente criterio:

- En caso de existir una sola ruta a una red se incluye en *table\_route*
- En caso de existir más de una ruta se seleccionarán la de menor *distancia\_local* (*active*). Además se tomarán como rutas alternativas (*backup*), todas las rutas cuya *distancia\_adyacente* sea estrictamente menor que la *distancia\_local*.
- En caso de existir más de una ruta que tengan igual valor de *distancia\_local*, se colocarán todas en la tabla de ruteo.
- En caso de caída de una interfaz, el protocolo eliminará las rutas que quedaron sin funcionar, y en caso de existir rutas de backup, pasará a activa la que corresponda. En caso de no existir rutas de backup, notificará la caída por todas las interfaces, con un mensaje de *inform*. El equipo que recibe un mensaje de *inform*, en caso que la información modifique tabla de ruteo (*table\_route*), notificará la modificación por todas sus interfaces (exceptuando por donde la recibió).

Se pide:

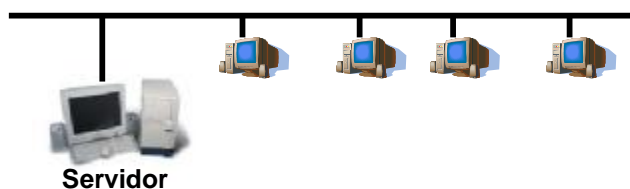
- a) Especifique en un lenguaje de alto nivel, el procedimiento que ejecuta el protocolo, cuando recibe un mensaje de *update*, conteniendo la tabla de ruteo del nodo adyacente.
- b) Dada la red presentada en el siguiente dibujo, donde el costo de las interfaces se encuentra expresado en el enlace que los une (X), y el costo en el Router A de la LAN 10.1.1.0/24 es 0, indicar la tabla de ruteo (*table\_route*) generada por el protocolo, para la red 10.1.1.0/24 en los Routers B, C, D, y E.



- c) Indique como se comportaría el protocolo en caso de caída del enlace *Router B – Router D*, y como se reformularían las tablas de ruteo.

## Ejercicio 2

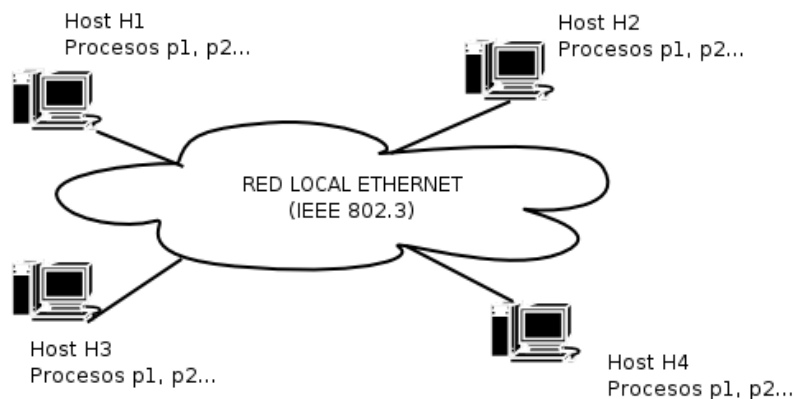
En una LAN se instalaron varias máquinas sin disco que no cuentan ni con dirección IP ni sistema operativo. Las máquinas disponen de una EPROM donde se puede instalar el software necesario que para obtener una dirección IP desde un Servidor dedicado a esa tarea, y de ser necesario descargar una imagen del sistema operativo a correr. Se desea implementar un protocolo sobre UDP, tipo RARP, que se encargue de obtener dicha dirección IP. El protocolo utilizará los puertos BOOTSRV en el Servidor y BOOTCLI en las máquinas cliente.



- Explique el protocolo que deberá correr en el Servidor y las maquinas cliente, y como interactúan (paquetes que intercambian), incluyendo la estructura de los paquetes a utilizar.
- Implemente en un lenguaje de alto nivel el protocolo que deberá correr en el Servidor y en las máquinas cliente.
- Suponiendo ahora que la LAN esta dividida en 2 segmentos conectados por un Router, explique si este protocolo representa alguna ventaja o desventaja con respecto al RARP estándar y realice una comparación de ambos protocolos.

### Ejercicio 3

Se considera una red compuesta por una cierta cantidad de hosts que se comunican a través de una red Ethernet.



Existen procesos en los hosts que necesitan mantener información de estado propia sincronizada. Los vecinos no se conocen entre si a priori, y no existe ninguna forma de directorio centralizado donde ésta información, ni ninguna otra, esté disponible. Se desea que esto sea así para evitar los cuellos de botella y puntos únicos de falla asociados con la centralización de información. Cuando un host envía un mensaje, asume que el mismo le va a llegar **a todos** sus vecinos de manera **confiable**.

Para comunicarse entre si los procesos disponen de llamadas al sistema operativo. En particular nos interesan dos:

- enviar\_vecinos(mensaje)**, donde "mensaje" es la información que el proceso quiere enviar a todos sus vecinos.
- recibir\_de\_vecinos(mensaje, mac\_addr)**, donde "mensaje" es el mensaje recibido y "mac\_addr" es la dirección MAC originaria del mensaje.

Se pide:

- Diseñar un protocolo implementado directamente sobre Ethernet, que permita cumplir con todos los requisitos. Debe tenerse en cuenta que la red local puede perder tramas, que los hosts pueden reiniciarse y que eventualmente alguna trama podría duplicarse. En este punto debe especificarse claramente el formato de las PDUs a utilizar (es decir el encabezado del protocolo), los campos que se van a utilizar así como el significado de los mismos.
- Implementar el protocolo diseñado en la parte (a) en un lenguaje de alto nivel, tanto a nivel de transmisión como de escucha. Se pide ser muy claro en el armado de las tramas, direcciones de origen y destino, campos del protocolo, etc. El driver de la tarjeta de red de cada host provee dos funciones: **send\_eth()** y **rcv\_eth()**.