

Examen de Introducción a las Redes de Computadoras y Comunicación de Datos (ref: eirc0312.doc) 20 de diciembre 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

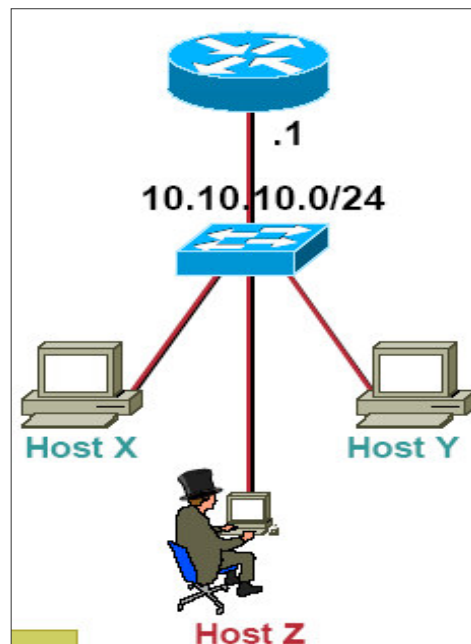
Sea la LAN con TCP/IP sobre ethernet, mostrada en la figura, donde todos los equipos están interconectados a través de un *switch* central único, y su salida a INTERNET, es a través del *router* mostrado en la figura (IP: 10.10.10.1), definido como *default gateway* en todos los equipos de la LAN.

Un usuario con malas intenciones, logra conectar su equipo, *Host Z* al switch de la red y realiza el envío de paquetes de respuesta ARP (formato broadcast), que indican que la IP 10.10.10.1, tiene la MAC address (dirección ethernet) que corresponde a su equipo *Host Z*.

Se supone que todos los equipos toman la respuesta de ARP como válidas y modifican su tabla ARP.

Se pide:

- Explicar que ocurre en la red cuando el *Host Z*, realiza lo expuesto, aclarando que ocurre en especial con flujos de información hacia Internet, y flujos de información entre equipos de la misma LAN.
- Especificar en un lenguaje de alto nivel un procedimiento que permita al *Host Z* efectuar la maniobra, y guardar todo el tráfico recibido, generando una situación tal que el resto de los equipos no detecten lo que se está realizando (que sea imperceptible para el resto de los equipos).



Solución:

Parte a)

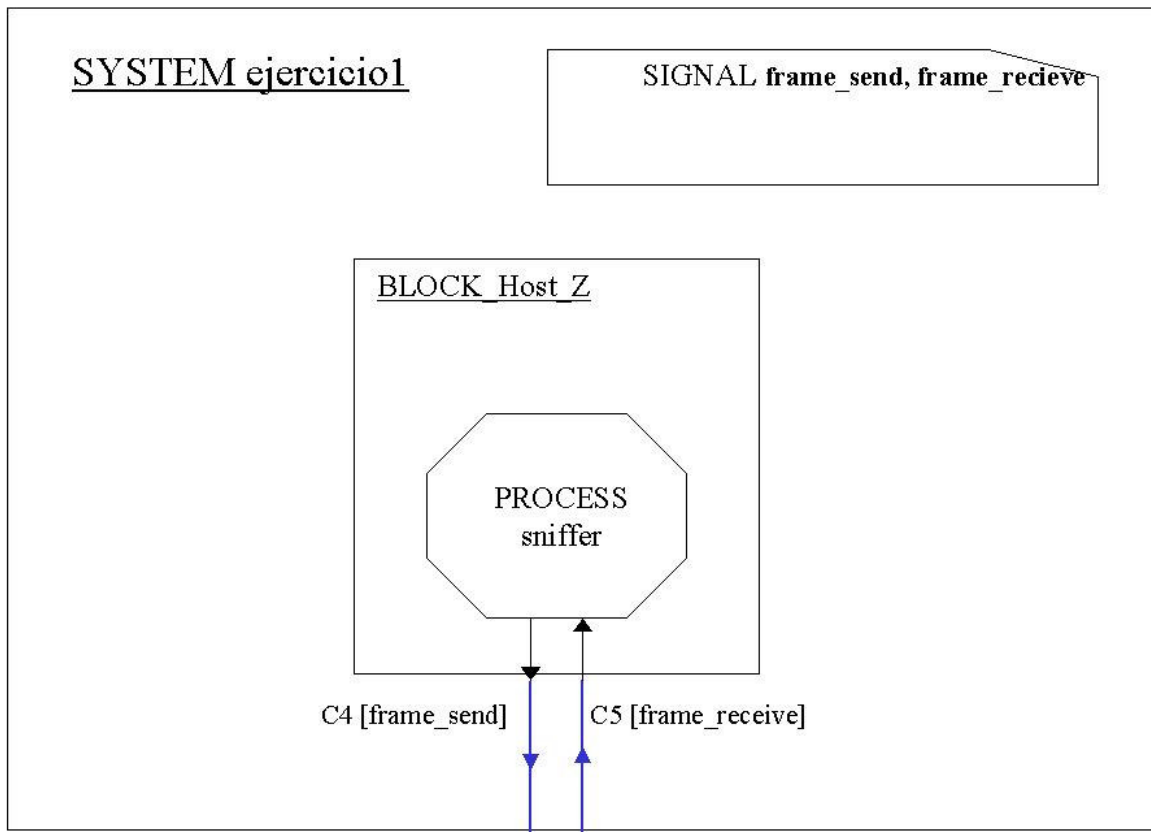
El switch es un dispositivo de capa 2 que aprende la dirección ethernet que tiene en cada puerto, por lo que una vez que lo aprende no envía los frames por todos los puertos, sino que solamente por los que reconoce tener el equipo. Esto hace que por un puerto del switch solamente se puede tener los frames dirigidos a mi MAC, y los broadcast.

El generar frames tipo ARP, donde el host define su MAC (ethernet address) como la del router de salida hace que todos los equipos modifiquen su tabla ARP. En consecuencia, todos los equipos que deseen enviar un paquete a una IP fuera de la LAN, lo dirigirán con un frame ethernet hacia la default gateway, enviando un paquete IP (con dirección IP origen y destino), pero enviado en un frame hacia el router.

Por lo tanto, todos los flujos TCP/IP, UDP, que tengan como destino un equipo externo a la LAN, (que no sean de la LAN 10.10.10.0/24) el host lo enviará en un frame ethernet encapsulado en un frame ethernet con dirección destino la MAC del router, y al haberse modificado la tabla ARP, eso genera que recibirá el *Host Z* todos los flujos con destino saliente. El equipo no podrá obtener la respuesta, la parte del flujo que proviene del equipo externo a la LAN, con dirección un equipo, ya que este se resolverá directamente en ARP, realizado por el router.

Todos los flujos de datos internos a la LAN (entre equipos de la 10.10.10.0/24) estos quedan sin modificarse, ya que son encapsulados en frames ethernet con dirección destino el propio equipo (resuelto por ARP), y el switch no publica (como realizaría un HUB) los frames por todos los puertos.

Parte b)

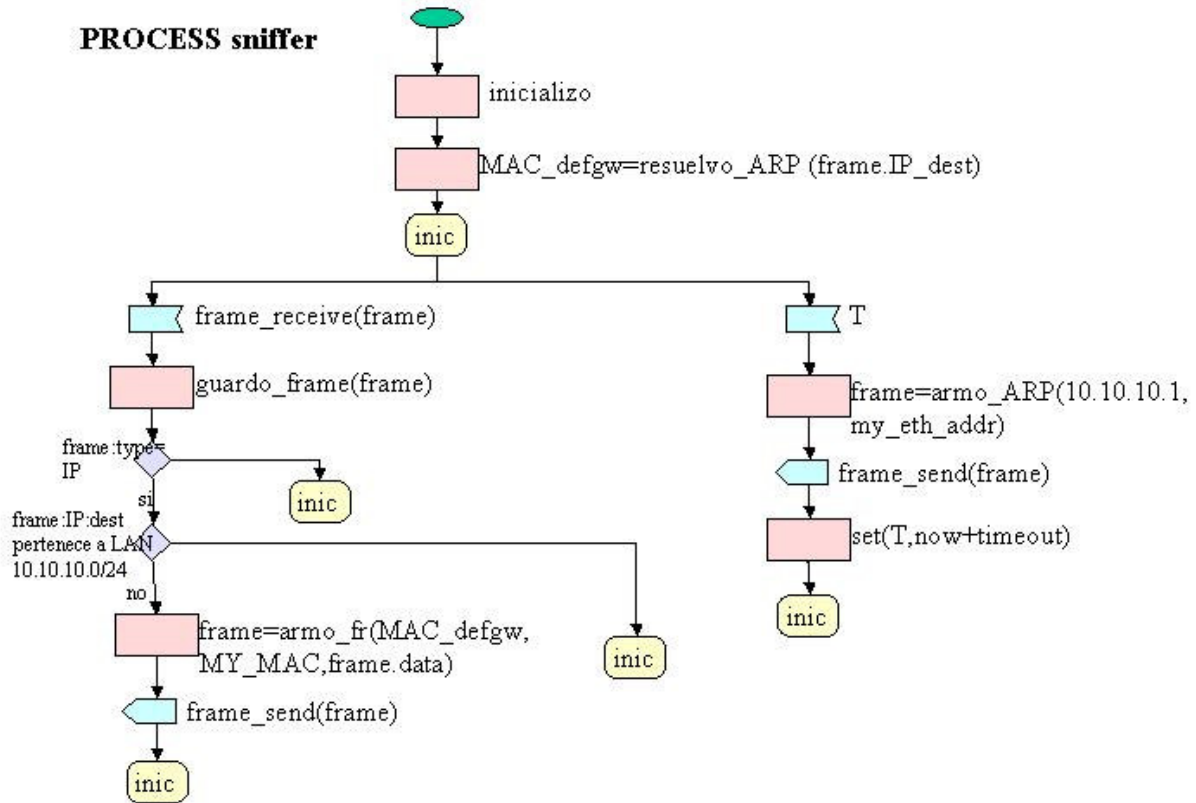


Para resolver el problema propuesto, se realizará el siguiente procedimiento:

- Trabajaré a nivel de capa 2, recibiendo frames y dirigiéndolos hacia donde corresponda con el fin de que los restantes equipos no noten las modificaciones realizadas en la LAN.
- Inicialmente hallaré la dirección MAC (ethernet) del router de salida (default gateway).
- Cada cierto tiempo T, enviaré una respuesta ARP, para asegurar que el equipo *Host Z*, figure en las tablas ARP de los diferentes equipos, como la dirección ethernet (MAC) del router.
- Todo frame recibido, lo almacenaré en un archivo.
- Los frames recibidos los procesaré de la siguiente forma:
- Todos los frames que no contengan paquetes IP, no serán retransmitidos.
- Si la IP destino del frame no pertenece a la LAN, entonces debe ser dirigido a el router de salida, por lo que armo un frame ethernet, y envío el paquete a la salida.
- El resto de los frames los descarto.

- Resumiendo, los paquetes originados en los hosts, saldrán en un frame con destino el *Host Z*, y los enviará al router., los paquetes que provengan del router no podrá verlos, ya que el router resolverá el ARP y enviará directamente.

Si el Host Z quisiera tener el flujo concreto de un host, debería mentir su dirección ethernet y hacer pasar el flujo por su equipo.



Utilizaré para la resolución las siguientes funciones auxiliares:

Function `armo_ARP(IP_address, eth_addr)` : frame

Devuelve un frame tipo ARQ response (broadcast) donde coloca la en los campos correspondientes a la respuesta, la IP definida en el campo `IP_address` y en el dirección ethernet, la pasada en el campo `eth_addr`

Function `armo_fr(eth_addr_destino, eth_addr_orig, data)`: frame

Devuelve un frame ethernet donde coloca como dirección destino la definida en `eth_addr_destino`, en la dirección origen del mismo la dirección `eth_addr_orig`, y en los datos lo que se pase en `data`

Function `resuelvo_ARP(IP_addr): eth_addr`

Dado una dirección IP, a la que se desea enviar un paquete, y dado que esta IP pertenece a la LAN, la función devuelve la dirección ethernet (MAC) del equipo (ya sea porque existe en la tabla ARP, o porque lo resuelve a través del protocolo ARP). En caso de no resolverse, debe contestar un valor especial

Procedure `guardo_frame(frame)`

Almacena en un archivo el frame que se le pase como parámetro.

Ejercicio 2

Se considera un restaurante que ha informatizado el mecanismo por el cual los comensales llaman a los mozos. A esos efectos, cada mesa cuenta con un dispositivo que se comunica, en forma inalámbrica, con otro dispositivo ubicado tras el mostrador.

Los dispositivos ubicados en las mesas cuentan con un botón, que es pulsado por los comensales al requerir servicio. El dispositivo ubicado tras el mostrador muestra el número de la(s) mesa(s) llamante(s) que haya detectado.

Todos los dispositivos mencionados (los de las mesas y el ubicado detrás del mostrador), utilizan un mismo canal de comunicación (misma frecuencia, misma forma de modulación, etc..).

Se pide:

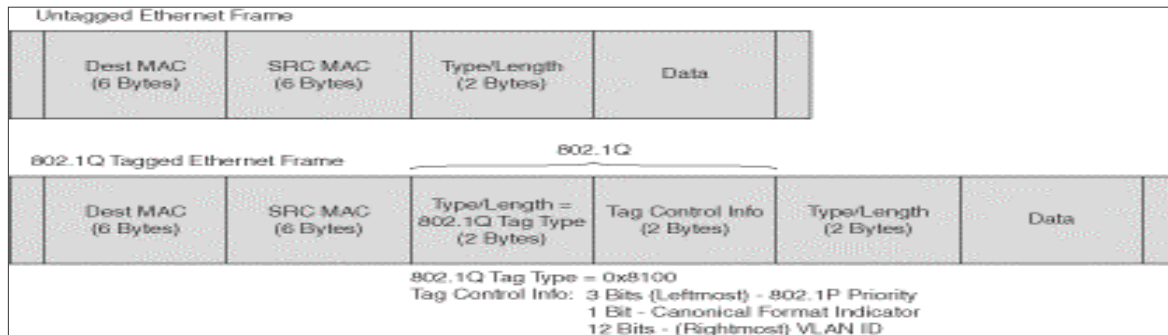
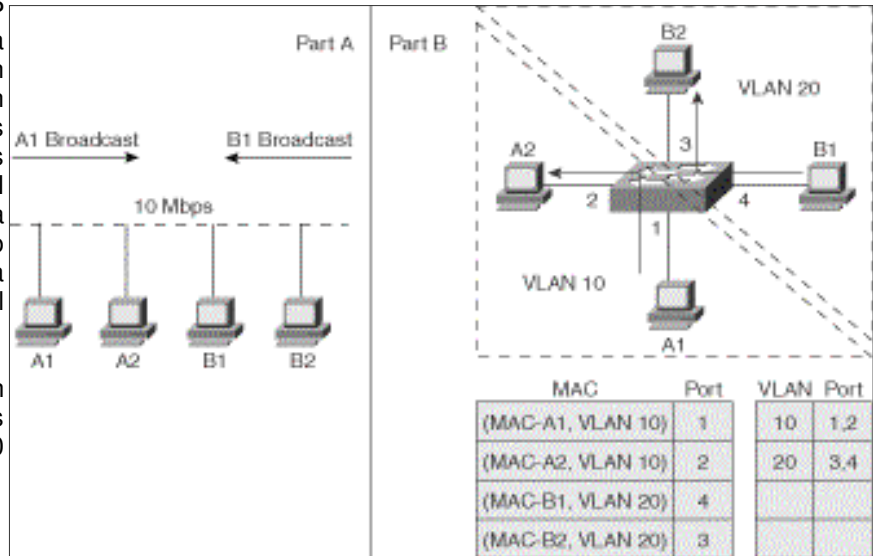
- a) A partir de los datos disponibles, realizar un diseño detallado del protocolo de comunicaciones, asumiendo que los dispositivos ubicados en las mesas sólo transmiten (no “escuchan”), y que el ubicado detrás del mostrador solo recibe. Prestar especial atención al tratamiento de las colisión.
- b) Mejorar el tratamiento de las colisiones, en beneficio de los comensales, asumiendo que todos los dispositivos pueden transmitir, escucha lo que transmiten , y recibir.

Solución: a implementar.

Ejercicio 3

Se cuenta con una red TCP/IP conectada a través de un dispositivo de capa 2, con funcionalidad extendida. El switch soporta el protocolo de VLANs 802.1Q. Una VLAN, o virtual lan, es una forma de particionar una LAN (en una o más redes virtuales). Esta forma es definida en el propio dispositivo (*switch*), asignando a cada VLAN uno o más puertos del mismo.

En la figura se puede ver un ejemplo, donde se definen las VLAN 10 (equipos A1 y A2) y 20 (equipos B1 y B2).



El protocolo de comunicación es básicamente una extensión del ethernet, introduciendo “tags” o etiquetas, que son propias del protocolo. Se muestran en la siguiente figura, dos frames comparados, siendo el primero un ethernet común y el segundo uno extendido (perteneciente a una VLAN).

Se pide:

- Especificar la diferencia entre broadcast generados en las VLAN, y fuera de las mismas
- Proponer estructuras de datos para almacenar las VLANs, sabiendo que los VLAN_ID (tags) son como máximo de un byte de tamaño (0..255).
- Implementar en un lenguaje de alto nivel el proceso que debe correr en el switch. Se debe considerar:
 - que no todos los puertos deben estar en alguna VLAN
 - se debe mantener la funcionalidad normal de capa 2 (se pueden recibir frames sin tag).

Solución:

La idea de este ejercicio es en base a las partes a) y b) pensar cuales son los problemas a resolver y en base a lo pensado en estas partes solucionar c). Hay varias soluciones posibles, en esta

solución se pondrán algunas variantes, pero no todas. Lo básico en principio es reconocer la diferencia en los broadcasts generados en ambos casos, y para esto se planteó la parte a).

Una posible respuesta sería la siguiente:

a) *Los broadcast generados en una VLAN sólo se reproducen a todas las bocas del switch que pertenecen a la misma. En cambio, si se produce un broadcast fuera de ellas, no debería reproducirse a las VLANs.*

Podría pensarse que los puertos que no están en una VLAN pertenecen a una VLAN especial con valor de VLAN ID = 1 o algo así. Esto podría simplificar el algoritmo. Pero dejémoslo en el caso más común que es el anterior.

Para la parte siguiente, la estructura básica está prácticamente en el recuadro. Deben tener en cuenta que se necesita la información de puertos asociando a VLANs, y a su vez los datos típicos de un switch, que sería la asociación entre MAC y puerto. Puede estar en una misma tabla, en varias, o como se les ocurra a Uds. También pueden agregar más información. Para este caso utilizaremos dos tablas.

b)

```
typedef struct mac_table {
    mac_t mac;
    port_t port;
} mac_table_t;

typedef struct vlan_table {
    vlan_id_t vlan_id;
    port_list_t port_list;
} vlan_table_t;
```

y definimos

```
#define mac_t byte[6] /* el ethernet address es de 6 bytes */
#define port_t byte /* a lo sumo habrán 24 puertos en un switch */
#define vlan_id_t byte /* la letra dice que no hay más de 255 vlans distintas */
#define port_list_t list<port_t> /* es una lista de puertos */
```

Estas serían las estructuras utilizadas.

Luego, para resolver el algoritmo, se debe clasificar el paquete recién llegado si es un broadcast o no, y luego si está dentro de una VLAN o no.

El proceso que ejecuta en el switch tiene las siguientes consideraciones:

- ☞ `es_frame_extendido(frame)` – retorna verdadero si el frame es del tipo extendido (`vlan_id`).
- ☞ `es_broadcast(frame)` – retorna verdadero si el frame es un broadcast.
- ☞ `buscar_destino(ports, vlan_id)` – devuelve el port que pertenece a dicha vlan o null si no lo encuentra. En el caso de `vlan_id` ser 'no_vlan', lo busca en los ports que no tienen vlan.
- ☞ `first, tail` -

SYSTEM
ejercicio 3

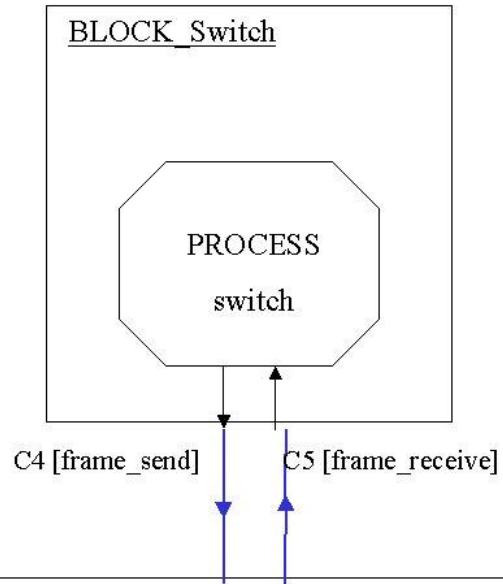
SIGNAL **frame_send**, **frame_recieve**

BLOCK_Switch

PROCESS
switch

C4 [frame_send]

C5 [frame_recieve]



PROCESS switch

