

**Examen de Introducción a las Redes de Computadoras
y Comunicación de Datos
(ref: sirc0512.doc)
26 de diciembre de 2005**

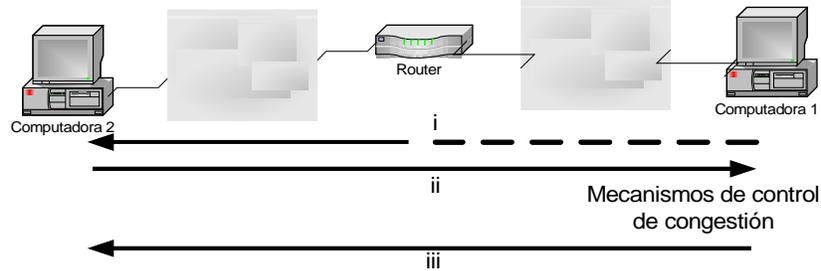
Atención: para todos los ejercicios, suponga que dispone de los tipos de datos básicos (p.ej. lista, cola, archivo, string, etc.) y sus funciones asociadas (ej: tail(lista), crear(archivo), concatenar(string, string).

Ejercicio 1

a) ¿Son adecuados los mecanismos de TCP de detección de congestión en la red en el caso en que el medio físico en el que se soporta la conexión es total o parcialmente inalámbrico? Justifique su respuesta. Considere solamente las pérdidas debidas a la hostilidad del medio y no a la congestión de la red.

b) Se pretende implementar un protocolo de notificación explícita de congestión que se soporte en 4 flags de señalización, 2 en el encabezado IP (ECT y CE) y dos en el encabezado TCP (ECE y CWR). El principio básico de operación sería el siguiente:

- Los extremos de una conexión que soporten éste protocolo, lo indicarán poniendo en "1" el bit ECT.
- Si un router que pretende encaminar un paquete con el bit ECT en "1", detecta la presencia de congestión incipiente en el camino hacia el destino, deberá poner en "1" el bit CE ("i" en el diagrama)
- Al recibir el paquete el destino final, si el bit CE está en "1", en el próximo segmento que genere hacia el origen de dicho segmento, deberá poner en "1" el bit ECE, lo que denominaremos "marcar el segmento" ("ii" en el diagrama).
- El origen, al recibir el segmento anterior, podrá aplicar los mecanismos de control de congestión que contemple el sabor de TCP implementado y lo informará al destino con el bit CWR en "1" ("iii" en el diagrama).
- El bit ECT cumple la única función de indicar a los routers intermedios qué paquetes tiene sentido marcar de ser necesario, pues los extremos de la conexión lo interpretarán adecuadamente.



La congestión en una red no aparece ni desaparece de forma repentina, por lo que es esperable encontrar ventanas de segmentos marcados y no simplemente segmentos marcados aislados.

Si se implementara éste protocolo en una red con algún segmento inalámbrico, ¿de qué forma ayudaría a la *performance* de TCP? Explique qué aspectos tendría en cuenta para contemplar éste protocolo en el funcionamiento de TCP. No se pide ninguna especificación en lenguaje de alto nivel.

- c) Especifique en lenguaje de alto nivel, el procedimiento que ejecuta un router intermedio que implementa éste protocolo de notificación y que considera como señal de congestión en determinado enlace, el tamaño del buffer de salida de la interfaz correspondiente.

Solución:

- a) ¿Son adecuados los mecanismos de TCP de detección de congestión en la red en el caso en que el medio físico en el que se soporta la conexión es total o parcialmente inalámbrico? Justifique su respuesta. Considere solamente las pérdidas debidas a la hostilidad del medio y no a la congestión de la red.

El protocolo TCP/IP busca controlar la congestión en la red desde los extremos de la conexión, ya que supone que la red (capa 3) hará su mejor esfuerzo para entregar el segmento al destino.

La protección en los extremos supone que cuando comiencen a perderse segmentos, esto se debe a que en el camino se debió descartar algún paquete (ocasionado por congestión en algún nodo), lo que ocasionó la pérdida de un segmento TCP. Así entonces el protocolo TCP disminuye entonces el flujo de información para buscar no saturar la red, y adaptarla a la capacidad existente sin saturar. Sin estas consideraciones no hubiese sido posible desarrollar una red con gran cantidad de nodos como internet.

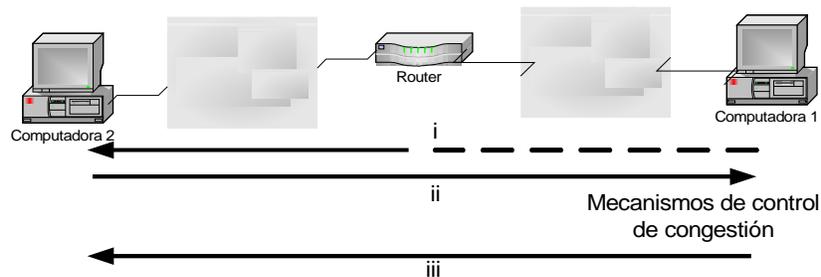
Esto en el caso de pérdidas ocasionadas por la hostilidad del medio, no se cumple. En este caso se supone que el paquete perdido fue el resultado de una falla en el enlace, y no de un descarte de la red. El protocolo TCP, igualmente considerará que el mismo se produjo por congestión y disminuirá

la tasa de transferencia (ya que no puede saber que la pérdida no fue causada por congestión), llevando a una tasa de transferencia efectiva menor, al protocolo.

De contar con la información del motivo de la pérdida, podría mejorarse el protocolo, ya que solamente en caso de congestión, tomaría la decisión de disminuir la tasa de transferencia.

b) Se pretende implementar un protocolo de notificación explícita de congestión que se soporte en 4 flags de señalización, 2 en el encabezado IP (ECT y CE) y dos en el encabezado TCP (ECE y CWR). El principio básico de operación sería el siguiente:

- Los extremos de una conexión que soporten éste protocolo, lo indicarán poniendo en "1" el bit ECT.
- Si un router que pretende encaminar un paquete con el bit ECT en "1", detecta la presencia de congestión incipiente en el camino hacia el destino, deberá poner en "1" el bit CE ("i" en el diagrama)
- Al recibir el paquete el destino final, si el bit CE está en "1", en el próximo segmento que genere hacia el origen de dicho segmento, deberá poner en "1" el bit ECE, lo que denominaremos "marcar el segmento" ("ii" en el diagrama).
- El origen, al recibir el segmento anterior, podrá aplicar los mecanismos de control de congestión que contemple el sabor de TCP implementado y lo informará al destino con el bit CWR en "1" ("iii" en el diagrama).
- El bit ECT cumple la única función de indicar a los routers intermedios qué paquetes tiene sentido marcar de ser necesario, pues los extremos de la conexión lo interpretarán adecuadamente.



La congestión en una red no aparece ni desaparece de forma repentina, por lo que es esperable encontrar ventanas de segmentos marcados y no simplemente segmentos marcados aislados.

Si se implementara éste protocolo en una red con algún segmento inalámbrico, ¿de qué forma ayudaría a la *performance* de TCP? Explique qué aspectos tendría en cuenta para contemplar éste protocolo en el funcionamiento de TCP. No se pide ninguna especificación en lenguaje de alto nivel.

El protocolo presentado permite a los extremos de la conexión determinar en forma exacta cuando existe congestión en la red. Así el protocolo TCP/IP podrá tomar la decisión de disminuir el flujo de la conexión (aplicando *congestion avoidance*) cuando tenga la certeza de problemas de congestión, y no cuando simplemente se pierda algún paquete.

El algoritmo en los extremos para la utilización del protocolo presentado, en forma resumida, funcionaría de la siguiente forma (cuando recibe o envía un segmento):

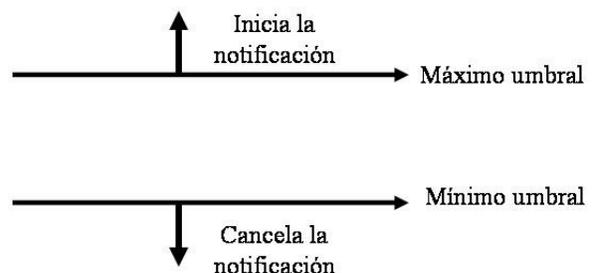
```
recibo_segmento(p);{  
    if p.CE = 1 then{  
        notif_congestion = true;  
    }else{  
        notif_congestion = false;  
    }  
    if p.ECE = 1 then{  
        estado_congestion = true;  
    }else{  
        estado_congestion = false;  
    }  
}
```

```
envio_segmento(p);{  
    if notifico_congestion then{  
        p.ECE = 1;  
    }  
    if estado_congestion then{  
        p.CWR = 1;  
        actualizo_congestion_Avoidance();  
    }  
}
```

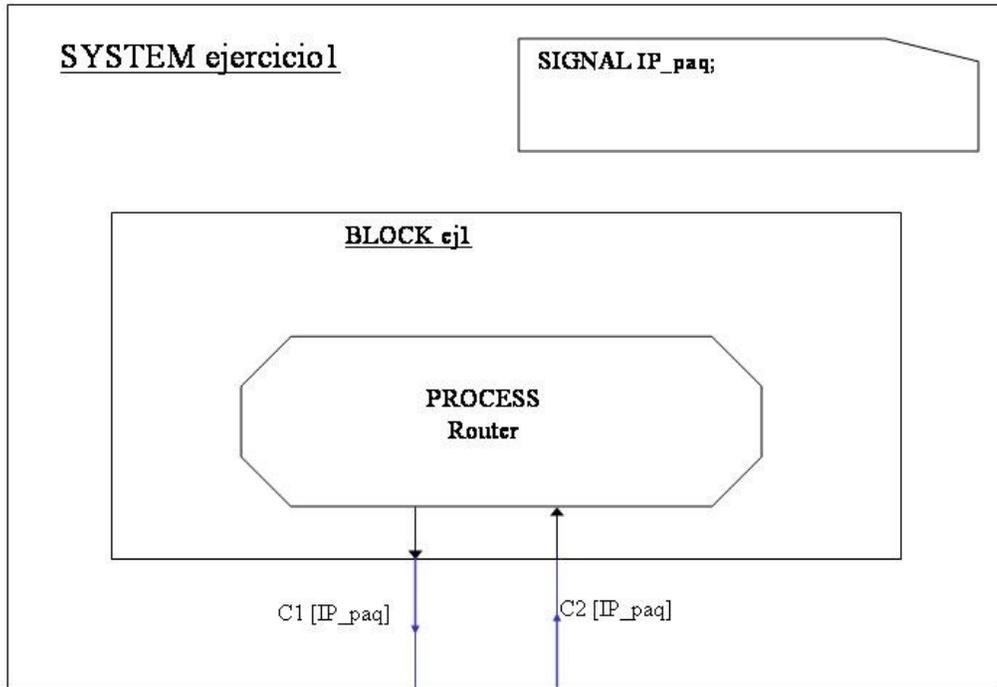
c)Especifique en lenguaje de alto nivel, el procedimiento que ejecuta un router intermedio que implementa éste protocolo de notificación y que considera como señal de congestión en determinado enlace, el tamaño del buffer de salida de la interfaz correspondiente.

La solución propuesta, plantea la existencia de un umbral donde, se activa la notificación por parte del protocolo cuando se supera un umbral (*max_umbral_congestión*) se comienza la notificación de la congestión y cuando se desciende del límite inferior del mismo (*min_umbral_congestión*) se desactiva.

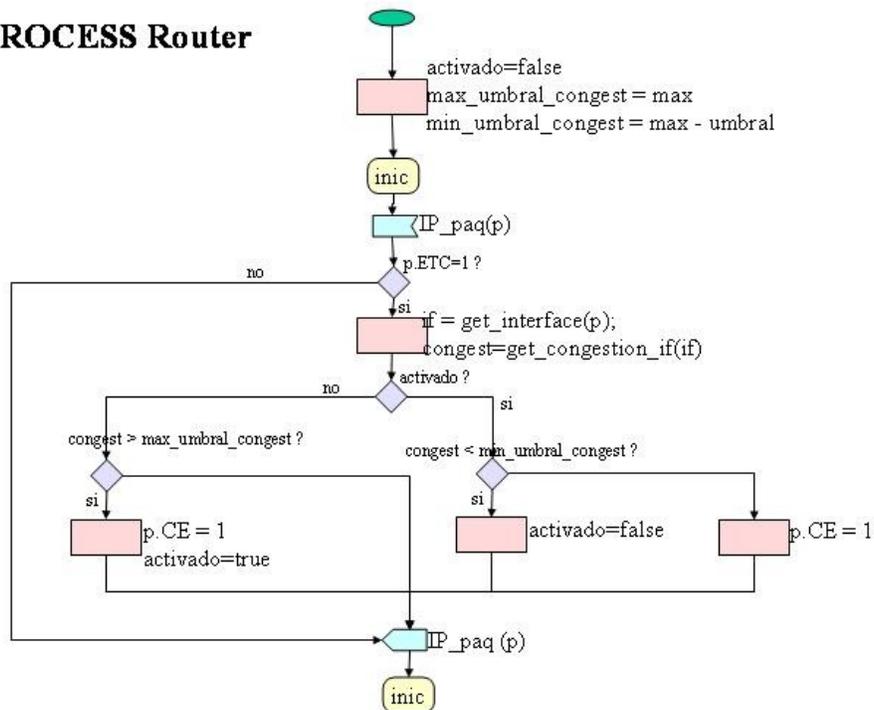
Esto busca no estar generando notificaciones contradictorias, cuando el router se encuentra cerca de la congestión, buscando dejar de notificar cuando se acerque a una situación estable.



En el procedimiento se definen dos variables, *max*, que corresponde al máximo umbral, y *umbral* que indica el ancho de la franja del umbral



PROCESS Router



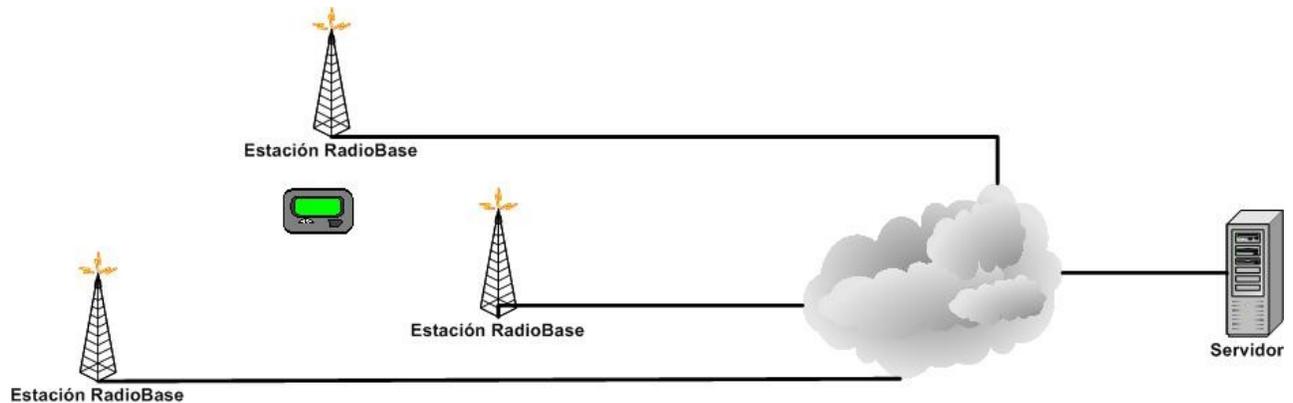
En el procedimiento se supone que se ejecuta un proceso por cada interfase de salida, y que se cuenta con las siguientes funciones:

- *function get_interface (paquete): interface_id*; función que devuelve la interfaz por donde saldrá el paquete para llegar a su destino, información relacionada con la tabla de ruteo del equipo.
- *function get_congestion_if(interface): integer*; función que devuelve el porcentaje de ocupación de los buffer de salida para la interfaz, en el momento de invocación.

El canal de salida corresponde a colocar el paquete IP en el buffer de salida. El proceso no toma decisiones sobre el descarte de paquetes, solamente los “marca”.

Ejercicio 2

Sea un sistema de comunicación de dispositivos móviles, los que interactúan con una serie de posibles estaciones radio-base, que efectivizan la comunicación. Se desea resolver únicamente la asignación de la estación radio-base que atenderá la comunicación que establecerá desde un dispositivo móvil.



Para la resolución del problema, se conoce los siguiente:

- Cuando un dispositivo móvil, desea realizar una comunicación, envía un mensaje a todas las estaciones radio-base que puedan recibirlo, incluyendo un timestamp que indica hora de enviado, y un identificador único del móvil (*movil_id*).
- Todas las estaciones radio-bases se encuentran en una red, donde se cuenta con un servidor que tiene como tarea definir qué estación radio-base atenderá la misma. La decisión se toma en base al siguiente criterio: la estación radio-base más cercana al móvil que cuente con canales libres.
- Todos los equipos (tanto móviles como estaciones radio-bases) se encuentran sincronizados en tiempo.
- El servidor esperará como máximo un tiempo estipulado (*tiempo_espera*) desde la llegada de la primera solicitud de comunicación desde cualquier estación radio-base, para informar cual es la seleccionada para realizar la comunicación.

Se cuenta con las siguientes funciones que se ejecutan en el servidor:

- **Function existe_canal_libre (radio-base) : boolean;** que devuelve true, en caso de que exista un canal disponible para la atención de una comunicación.

- `Function asigno_llamada (radio-base, movil_id);`, que asigna la atención de la comunicación a la estación radio- base pasada como parámetro.

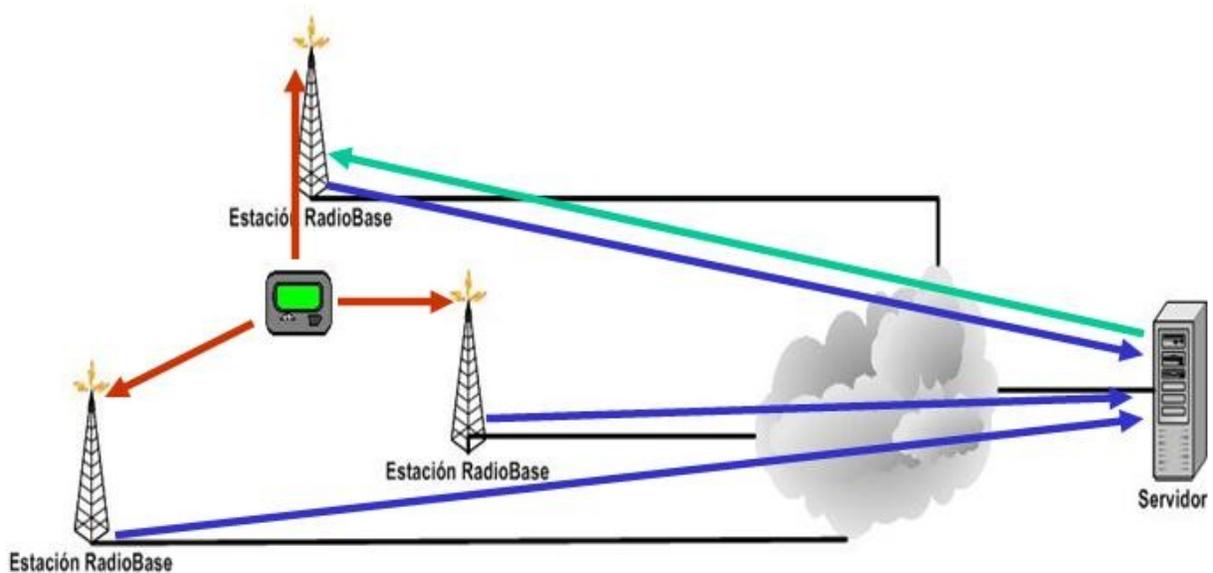
Se pide:

- Diseñar un protocolo implementado sobre la red TCP/IP que interconecta las estaciones radio- base, y el servidor, que permita resolver el problema planteado.
- Especificar en un lenguaje de alto nivel, el protocolo presentado en la parte a), que corre en la estación radio- base y en el servidor.

Solución:

- Diseñar un protocolo implementado sobre la red TCP/IP que interconecta las estaciones radio- base, y el servidor, que permita resolver el problema planteado.

La idea a implementar busca resolver asignar a la estación radio- base más cercana con canales libres, la comunicación solicitada.



Para esto supondremos un protocolo que actúa de la siguiente manera:

- Al recibir la estación radio-base el mensaje del móvil solicitando un canal para la comunicación, que incluye un time-stamp con la hora de solicitud (*time_movil*) flecha roja en el dibujo, la estación radio-base envía un mensaje al servidor, indicando el tiempo diferencia entre el instante de recepción y el time-stamp que incluía el mensaje de solicitud (*time_movil*), con el fin de determinar cual es la estación radio-base más cercana), y el identificador único del móvil (*movil_id*), flecha azul en el dibujo.
- Cuando el servidor recibe un mensaje de solicitud de canal de comunicación, desde una estación radio-base, almacena los datos recibidos (*movil_id*, *diff_time*). Además analiza si había recibido algún mensaje por parte del móvil, en caso contrario, setea el tiempo de espera de las repuestas.
- Una vez vencido el tiempo de espera de las respuestas, analiza comenzando en la menor diferencia *diff_time* ($time_de_recepciónRB - time_movil$) si dispone la estación radio-base de canales, en caso afirmativo, le asigna la comunicación (flecha color verde), y en caso contrario descarta y toma la diferencia siguiente.

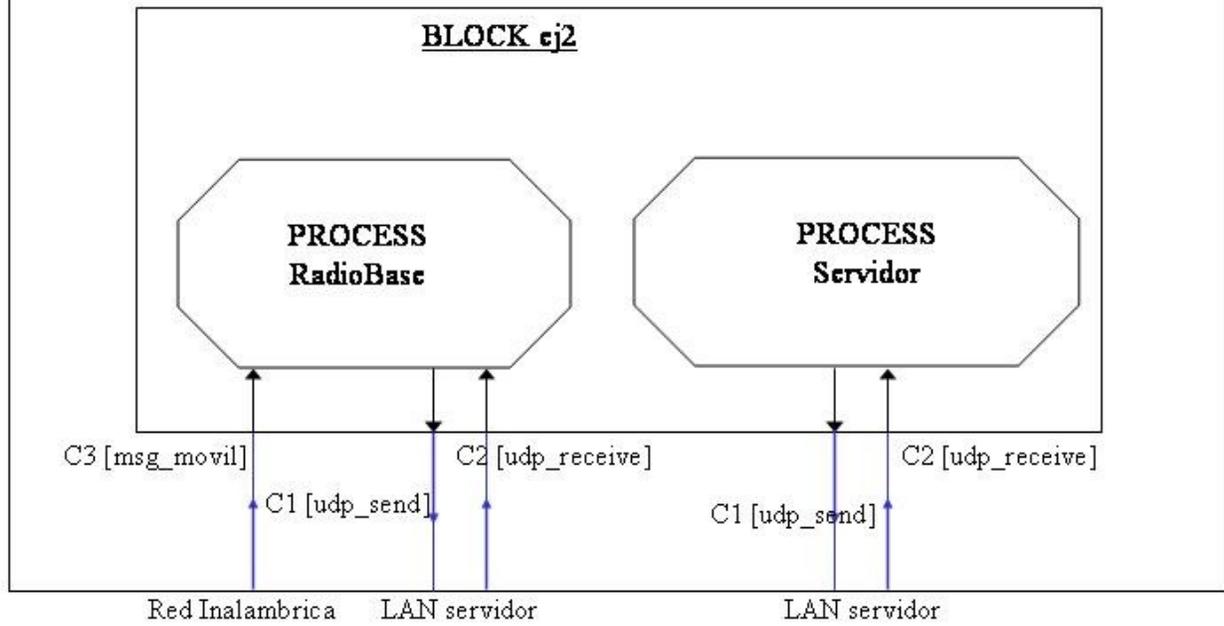
Para la mensajería del protocolo se propone la utilización de UDP, lo que independiza a las estaciones radio-base de establecimientos de conexión, que pueden consumir tiempos innecesario, considerando además que el over-head generado para el envío del mensaje sería altísimo.

Se toma como hipótesis además, que la precisión dada por la hora de los móviles y las estaciones radio-base, permite estimar la distancia entre la antena y el móvil en forma razonable.

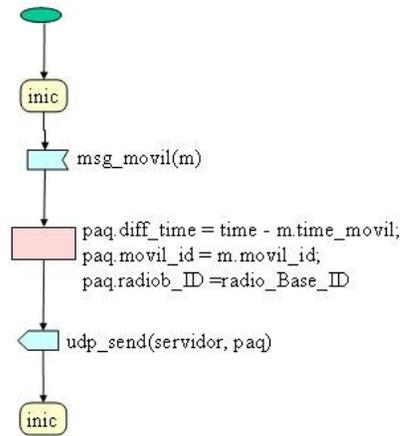
- b) Especificar en un lenguaje de alto nivel, el protocolo presentado en la parte a), que corre en la estación radio-base y en el servidor.

SYSTEM ejercicio2

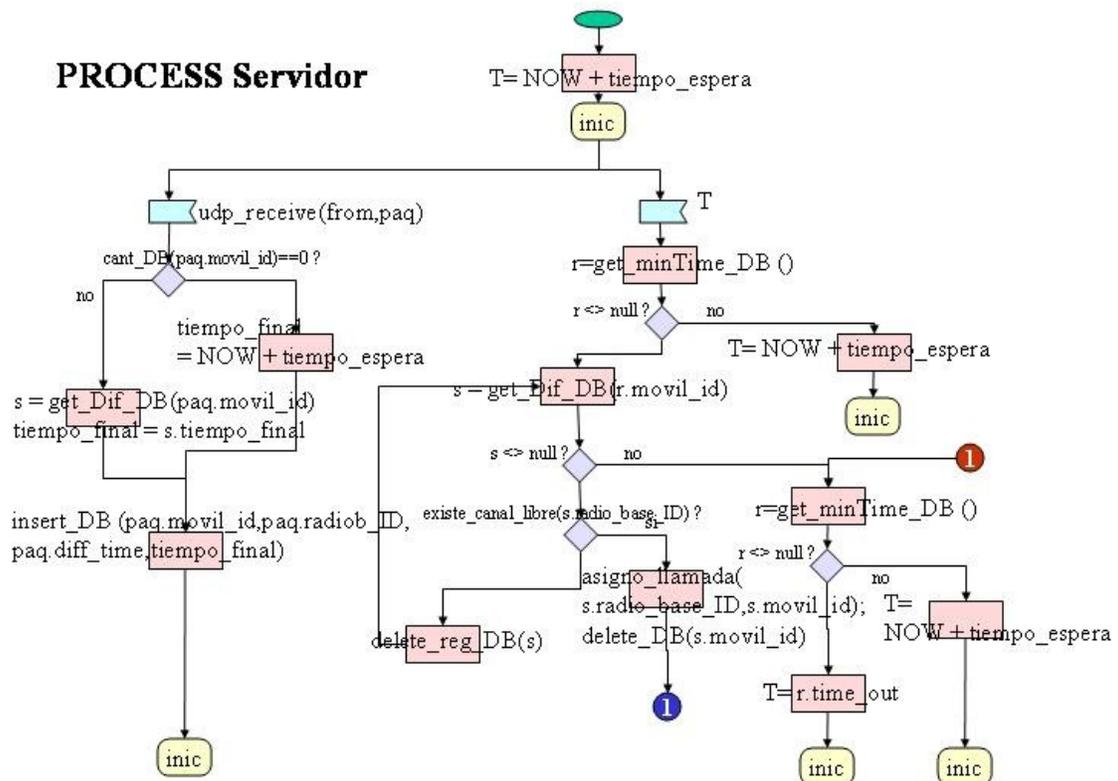
SIGNAL msg_movil, udp_send,
udp_recevie;



PROCESS RadioBase



PROCESS Servidor



Funciones auxiliares utilizadas:

- *insert_DB* (*movil_id*, *radio_base_ID*, *diferencia_tiempo_mobil-RB*, *vencimiento_tiempo_espera*), esta función inserta un registro en una base de información que contiene la información recibida en el segmento UDP, y el tiempo de vencimiento del registro.
- *cant_DB* (*movil_id*) devuelve cantidad de registro con el *movil_id* pasado como parámetro, 0 en caso de no existir.
- *get_Dif_DB*(*movil_id*) devuelve un registro del *movil_id* pasado como parámetro con menor *diferencia_tiempo_mobil- RB*. Null en caso de no existir registro
- *get_minTime_DB* () devuelve el registro con menor *vencimiento_tiempo_espera* de toda la base. Null en caso de no existir registro
- *delete_reg_DB*(*s*) borra el registro *s* de la base.
- *delete_DB*(*movil_id*) borra todos los registros de la base cuyo *movil_id* sea igual al pasado como parámetro.

Ejercicio 3

Se considera el enrutamiento en una red de gran porte basada en el protocolo IP. Se desea resolver el problema de la alcanzabilidad dentro de la red de manera dinámica ya que la envergadura de la red hace inviable la posibilidad de una configuración estática.

Los protocolos de enrutamiento dinámicos que se utilizan en este caso de estudio son protocolos que realizan “anuncios” de tablas de enrutamiento.

Se pide:

- a) Para la red mostrada en la figura, indique la cantidad de rutas que cada router necesita almacenar en su tabla de enrutamiento asumiendo que a cada red le corresponde una única ruta (es decir, no hay subdivisión interna de los rangos de IP asignados a c/u).

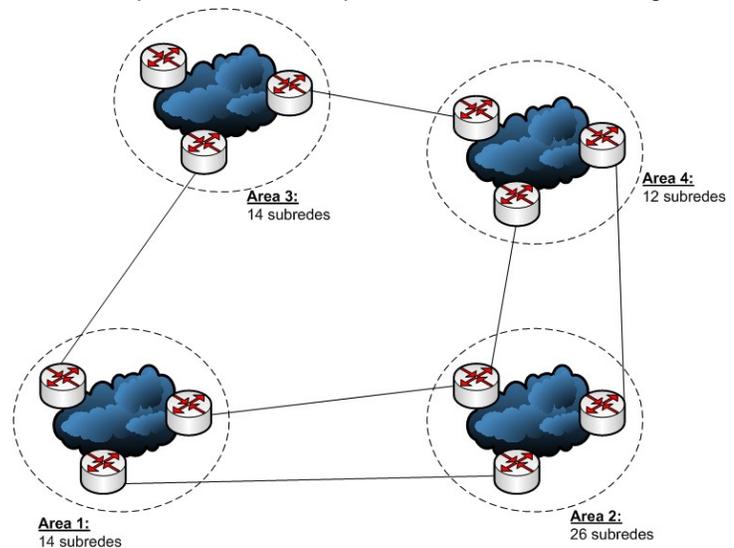
Tener en cuenta que los totales de subredes de cada area no incluyen a los enlaces de interconexión, pero deben tenerse en cuenta.

Considerar los siguientes casos:

- 1) Realizar esta estimación para el caso en que no es posible utilizar enrutamiento por defecto para las áreas.
- 2) Realizar esta estimación para el caso en que fuera posible utilizar enrutamiento por defecto entre áreas. ¿Todas las áreas pueden enrutarse por defecto? En caso negativo, ¿Por qué?

- b) Desarrollar un proceso a ejecutarse en c/u de los enrutadores ABR (Area Border Routers) que opere junto con el protocolo de enrutamiento dinámico y que permita realizar lo que se llama *sumarización automática* de rutas.

El proceso deberá analizar la tabla de enrutamiento y buscar agrupar las rutas en rutas menos específicas (con menos bits a “1” en sus máscaras). Esto no siempre será posible, ya que pueden faltar “partes” de una subred para poder sumarizarla completamente.



El proceso se activa solamente en las interfaces que comunican al ABR con otras áreas, afectando solamente a los anuncios que se realizan a través de las interfaces donde está activo. En este punto, actúa como un “filtro” de los anuncios que realizan los ABR.

La ruta menos específica a buscar será de N_{min} bits.

- c) ¿Por qué lo desarrollado en la parte 2 debe actuar como un filtro y no puede simplemente realizar sustituciones en la tabla de enrutamiento? Explique.

Solución:

a) Analizamos los casos separadamente, recordando que el objetivo es siempre tener *alcanzabilidad completa* (es decir, que no existan partes de la red que no se “vean” entre si):

1. Al no ser posible utilizar ninguna forma de enrutamiento por defecto, la información de alcanzabilidad debe estar completamente contenida en las rutas a las redes específicas. En este caso, cada ABR va a tener que tener todas las rutas, de su área y de las demás. Además, deberá tener las rutas a las subredes de los enlaces:

Área 1	$14 + 14 + 12 + 26 + 6 = 72$
Área 2	$14 + 14 + 12 + 26 + 6 = 72$
Área 3	$14 + 14 + 12 + 26 + 6 = 72$
Área 4	$14 + 14 + 12 + 26 + 6 = 72$

2. Si es posible utilizar enrutamiento por defecto en las áreas, podemos razonar de la siguiente manera: Cada ABR va a necesitar mantener la información completa de su área solamente, mas las rutas de los enlaces directamente conectados que pueda tener, mas una ruta por defecto.

No todas las áreas pueden enlutar por defecto, ya que dependiendo de la topología de la red, podrían generarse loops de enrutamiento. Lo que puede hacerse es definir una de las áreas como “área de tránsito”, la cual mantiene información completa de alcanzabilidad para toda la red, mientras que el resto solamente conoce sus rutas propias, sus enlaces y sus rutas por defecto.

Si definimos por ejemplo el Área 1 como “área de tránsito”:

Área 1	$14 + 14 + 12 + 26 + 6 = 72$	todas las rutas
Área 2	$26 + 4 + 1 = 31$	el “1” es la rpd
Área 3	$14 + 2 + 1 = 17$	el “1” es la rpd
Área 4	$12 + 3 + 1 = 16$	el “1” es la rpd

Podemos ver que se pueden lograr interesantes ahorros en la tabla de enrutamiento, pero al costo de que el recorrido de los paquetes no sea óptimo (todo el tráfico pasa siempre por el Área 1)

b) Primero algunas observaciones: buscaremos agrupar las entradas de la tabla de enrutamiento para formar bloques mayores que puedan anunciarse hacia otras áreas como una única ruta. La idea detrás de este principio es que para los routers exteriores a un área a la que solo pueden llegar por un enlace, es indistinto conocer una ruta agregada o desagregada.

Supongo la siguiente estructura para la tabla de enrutamiento:

```
TablaE = List of record
    netAdd: Integer32;
    netMask: Integer32;
    nextHop: Integer32;
end;
```

Dividimos el proceso en dos funciones:

1. function agrupar1(TablaE T): TablaE R

Esta función busca agrupar cada ruta individual con algún vecino, y en el caso de poder agrupar, borra dos elementos e inserta uno nuevo, colocando como next hop en la nueva ruta a si mismo. Recordar que el operador “&” es el AND lógico bit a bit (*bitwise AND*) y que el operador << es el “shift left” (sin acarreo en nuestro caso).

El algoritmo propuesto admite muchas optimizaciones pero la mismas quedan fuera del alcance de un ejercicio de examen.

```
function agrupar1(TablaE T): TablaE R
begin
    R := T
    foreach (t in R) do
        upMask = t.netMask << 1
        upNet = t.netAdd & upMask
        foreach (u in R) do
            if ( t != u ) then
                tmpNet = t.netAdd & upMask
                if ( upNet == tmpNet ) then
                    // se agrupan!! Los borro...
                    R.delete(t)
                    R.delete(u)
                    // me inserto a mi mismo como next hop
                    R.insert(
                        tmpNet,
```

```

                                upMask,
                                "yo mismo"
                                )
                                end
                                end
                                end
                                end
                                end

```

2. function consolidarTabla()

Esta función ejecuta `agrupar1()` sobre la tabla de enrutamiento hasta que la misma no cambia mas o hasta que se llegan a rutas de N_{min} bits de prefijo (como pide la letra). En cada ciclo de invocar a `agrupar1()` sobre una tabla temporal se logra "sumarizar" algunas rutas.

```

function consolidarTable(TablaE T): TablaE R
begin
    tmpT1 := T
    tmpT2 := T
    repeat
        tmpT2 := agrupar1(tmpT1)
        // funcion Min() se aplica a todos los netMask
        nBits := Min(tmpT2.netMask)
    until (tmpT2 == tmpT1) or (nBits<=Nmin)
    R := tmpT2
end

```

- c) El proceso de sumarización automático debe ejecutarse como filtro debido a que si el mismo modificara la tabla de enrutamiento local, el ABR perdería la información de alcanzabilidad local interna de su área. Esto no puede ocurrir.

La idea del proceso de sumarización es presentar a otras áreas mi área local como un todo opaco, pero en la interna debo conservar la información de alcanzabilidad completa.