



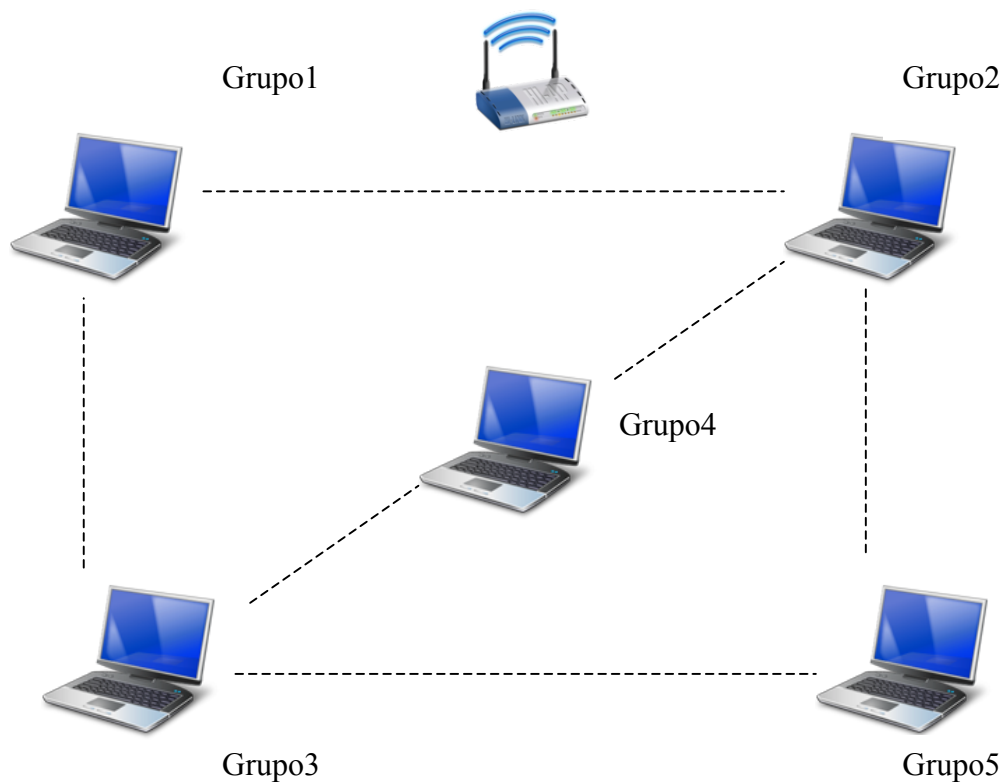
---

# **IMPLEMENTANDO UN ROUTER**



En este proyecto usted debe desarrollar un Router, implementando los módulos de Forwarding y Routing. Para ello se construirá una red de computadoras utilizando interfaces inalámbricas.

Dicha red está conformada por diferentes nodos, y cada uno de los grupos será el responsable de la implementación de uno de los nodos. En base a la cantidad de grupos se estará enviando un mapa con la topología de la red. A continuación encontrarán un ejemplo de una posible topología para una red de 5 nodos.



El mapa que será enviado con la topología de cada sección pretende mostrar las únicas conexiones válidas que se podrán hacer a través del medio inalámbrico. Es decir, en el ejemplo anterior, a pesar que el Grupo2 y el Grupo3 pueden verse mutuamente gracias a que están conectados al mismo Access Point, para cumplir



los objetivos didácticos de este proyecto la única forma en la cual estos dos nodos pueden comunicarse entre sí será a través de los nodos Grupo1, Grupo4 o Grupo5, y la ruta será seleccionada en base a las políticas de ruteo establecidas para esta red.

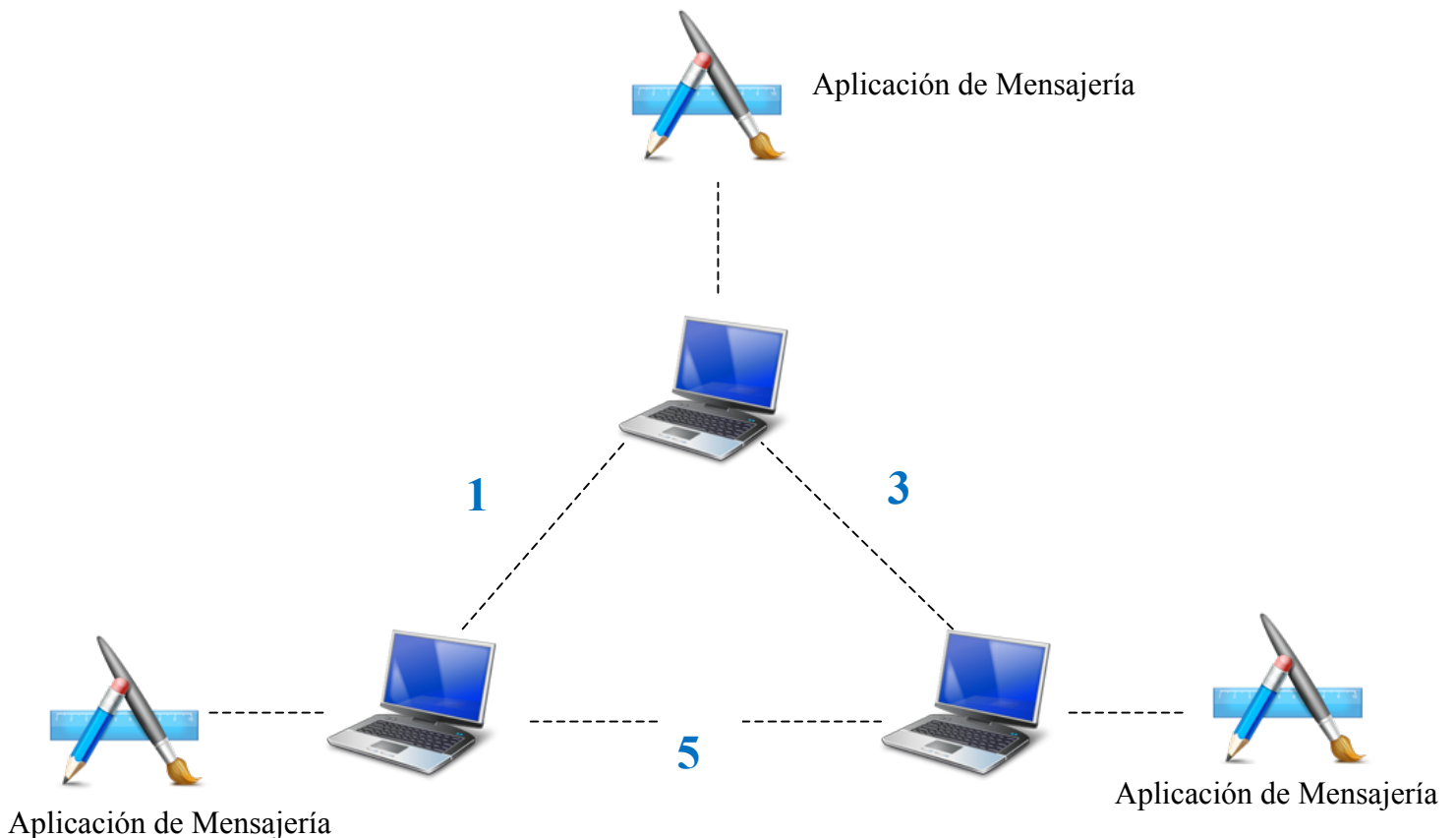
A continuación se describen los módulos a implementar por cada nodo:

## Routing

Este modulo se encarga de ejecutar el algoritmo de ruteo y establecer las rutas más cortas hacia todos los destinatarios en la red.

Cada grupo estará representado por una computadora donde estarán corriendo los procesos de Ruteo, de Forwarding y una Aplicación de Mensajería.

De manera tal que una topología para 3 grupos se verá de la siguiente forma:





Cada uno de los enlaces entre dos routers tendrá un costo asociado. El mapa de la Red se dará a conocer hasta el día de la calificación, por lo que se recomienda que se corran pruebas intra e inter grupos para que puedan validar la correcta implementación del protocolo.

Al iniciar, cada Router debe recibir como parámetro un listado con los nodos adyacentes y los costos entre los nodos. Estos parámetros deben ser almacenados en un archivo de configuración, el cual será interpretado cada vez que el Router sea iniciado. En el caso de la figura anterior, el Router del GrupoC debe ser inicializado con el conocimiento de la Red B y el costo para alcanzar dicha red es de 5 y además, también debe inicializarse con el conocimiento de la red A con un costo de 3. Este conocimiento inicial acerca de sus nodos adyacentes debe ser tomado en cuenta para que cada router comience a configurar sus tablas de ruteo en el momento de ser encendido. Las rutas estarán calculadas en términos de las direcciones de las Aplicaciones de Mensajería, en el ejemplo anterior, las direcciones a “rutear” serían A, B y C.

El algoritmo a implementar, será un caso de Distance Vector (DV), donde se mantendrán sesiones abiertas entre los routers adyacentes y se comunicarán entre sí toda la información relativa a los cálculos y actualizaciones de sus DV para todos los posibles destinos de esta red. Para ello debe poner en práctica los conocimientos adquiridos en clase. Queda a discreción de cada grupo la implementación del algoritmo de DV y las estructuras de datos a utilizar, sin embargo debe existir un protocolo estándar de comunicación entre los routers, el cual se describe a continuación:

- Cada Router se conecta al puerto 9080 de sus adyacentes. Por lo que los Routers deben tener dos modalidades de operación, un thread cliente que está enviando sus notificaciones de DV a través del puerto 9080 de sus vecinos y un thread servidor que está escuchando en el puerto 9080 las notificaciones de DV que sus vecinos le están enviando.

La inicialización de conexión requiere el envío de un saludo inicial con el siguiente formato:

```
From:<Router que notifica>  
Type:HELLO
```



El Router que recibe la solicitud de conexión debe asignar recursos para atender a su Nuevo vecino y en todo momento mantener un log que muestre cuantos routers adyacentes han iniciado sesión.

Al aceptar la conexión, el Router que recibe la solicitud debe contestar con el siguiente mensaje:

```
From:<Router que notifica>  
Type:WELCOME
```

En todo momento los Routers deben garantizar dos conexiones con cada adyacente. Una conexión donde se comportan como el Servidor del módulo de ruteo donde únicamente escuchan mensajes y notificaciones provenientes de su vecino y la otra conexión donde participan como Cliente en la comunicación de ruteo, siendo a través de este segundo canal donde ocurre la comunicación desde su router hacia su vecino, informando de cambios en los Distance Vector.

- Cada Router debe tener un parámetro de configuración donde se pueda configurar el intervalo T en el cual se estarán enviando los mensajes entre los routers. Por defecto este intervalo será inicializado en 30 segundos.
- Al vencerse dicho intervalo, el Router debe enviar una notificación en base al estado de su tabla de Distance Vectors, teniendo dos posibles escenarios:
  - 1) Se han realizado cambios en los Distance Vectors. Es decir, se ha encontrado una nueva ruta con un costo más corto para un destino determinado. En este caso se debe enviar la actualización de rutas a los routers adyacentes, siguiendo este formato:

```
From:<Router que notifica>  
Type:DV  
Len:<Cantidad de Líneas>  
<Dirección Destino> :<Costo de la Ruta>  
<Dirección Destino> :<Costo de la Ruta>  
  
.....  
<Dirección Destino> :<Costo de la Ruta>
```



Por ejemplo, en el gráfico anterior, al iniciarse el Router del GrupoA debe configurar los DV hacia sus adyacentes, por lo que su primera notificación debe incluir dicha información, quedando de la siguiente manera:

El mensaje desde el GrupoA hacia el GrupoC

From:A  
Type:DV  
Len:2  
C;3  
B:1

El mensaje desde el GrupoA hacia el GrupoB

From:A  
Type:DV  
Len:2  
C;3  
B:1

De esta forma, el router del GrupoC podrá conocer que el Router del grupo A conoce dos rutas, una hacia la dirección C con un costo de 3, y una hacia la dirección B con un costo de 1. El análisis se debe llevar a cabo de la manera vista en clase.

Recuerde qué únicamente se transmiten los Distance Vector que hayan sido actualizados o cambiados desde el último envío de notificaciones.

2) No han ocurrido cambios en los Distance Vectors. Si durante el intervalo T no han ocurrido cambios en los Distance Vectors, entonces únicamente se envía un mensaje “keep alive” con el siguiente formato:

From:<Router que notifica>  
Type:KeepAlive



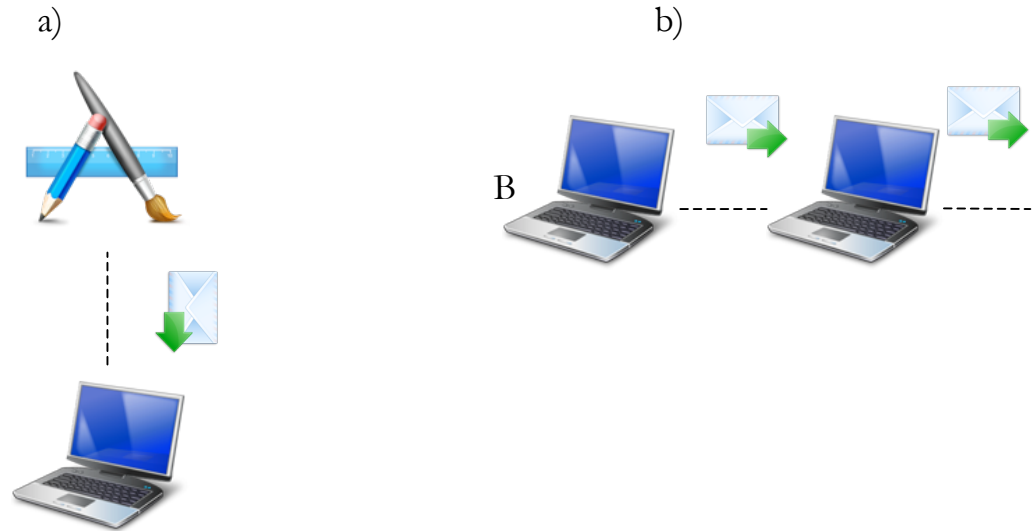
- Si un Router no recibe mensajes de su adyacente por un intervalo de tiempo  $U$ , entonces el Router es libre de asignar un costo infinito a la conexión entre él y el Router que no se ha comunicado, provocando una actualización en las rutas. Este intervalo  $U$  tiene un valor por defecto de 90 segundos y debe ser parametrizable. En este proyecto, un valor infinito será representado como un número lo suficientemente grande, 99.

El módulo de ruteo también debe ser diseñado para soportar reconexiones con los router vecinos. Por ejemplo, el GrupoB tiene un problema con su router y debe detener la ejecución de su proyecto para realizar cambios. Por lo que sus conexiones con los Grupos A y C será detenidas. Tanto A como C eventualmente marcarán al Router B como un router que se ha caído (Al momento de cumplirse el intervalo de tiempo  $U$ ). Cuando el GrupoB logra resolver el problema que tenía, procede a levantar de nuevo su router, por lo que este intentará la reconexión con sus routers adyacentes, todos los routers deben soportar este tipo de reconexiones provenientes de sus vecinos sin tener que detener su programa para volver a levantarlo. Recuerde tomar en cuenta para este proceso de reconexión que en todo momento deben existir dos canales de comunicación abiertos, uno Cliente/Servidor y otro Servidor/Cliente.

## Forwarding

El modulo de forwarding es el encargado de analizar cada mensaje de capa de aplicación y reenviarlo por la ruta que corresponda en base al análisis de la capa de ruteo. Este modulo estará escuchando en el puerto 1981 y podrá recibir mensajes de dos fuentes:

- a) La aplicación que está corriendo en ese Nodo.
- b) Un mensaje reenviado por otro Nodo.



En cualquiera de los dos casos, esta capa debe revisar la tabla de ruteo válida en ese momento para reenviar el mensaje por la ruta que corresponda. La decisión se realiza en base a la dirección destino del mensaje a enviar.

El reto consiste en hacer el forward del mensaje recibido. Si la dirección destino corresponde a la dirección del Router, entonces el módulo de Forwarding debe entregar el mensaje al módulo Aplicación (descrito a continuación.) Este caso se da cuando el mensaje es para el Grupo A y efectivamente, el router A está recibiendo el mensaje a través de alguno de sus vecinos. El otro escenario es que el Router A reciba un mensaje con una dirección diferente a la propia, por lo que en este caso el módulo de Forwarding debería reenviar dicho mensaje hacia el router adyacente que corresponda, tomando en cuenta la tabla de Distance Vectors construida por el módulo de Ruteo.

Esta capa debe mostrar en pantalla una impresión de cada mensaje que ha pasado por este módulo, mostrando quien le ha enviado el mensaje y hacia donde fue reenviado/entregado el mensaje. Esta impresión nos permitirá realizar el “debugging” de la red.





## Aplicación

Cada Router está ejecutando una Aplicación de Mensajería, cuya única función es el envío y recepción de mensajes que probarán la red.

Esta aplicación debe permitir al usuario enviar mensajes a cualquiera de los posibles destinos de la red. (Direcciones del tipo A, B, C, etc.).

El formato de los mensajes es el siguiente:

```
From:<Dirección de la aplicación origen>  
To:<Dirección de la aplicación destino>  
Msg:<Mensaje>  
EOF
```

Por ejemplo, A quiere enviarle un mensaje a B:

```
From:A  
To:B  
Msg:Hola Mundo  
Como estan?  
EOF
```

Cada aplicación debe poder mostrar los mensajes que está recibiendo y dar la opción de enviar uno Nuevo, especificando el destinatario y el cuerpo del mensaje.

Los mensajes serán enviados y recibidos a través del módulo de Forwarding.

### Entregables:

- Especificación Funcional (Detalle de requerimientos funcionales, no funcionales y de dominio.)
- Especificación de Diseño y Técnica (Detalle de la arquitectura de la solución propuesta. Acompañe su documentación con Diagramas UML)
- Código fuente del proyecto
- Checklist de pruebas realizadas.