

1. INTRODUCCIÓN

El presente documento describe la realización del Proyecto de Fin de Carrera “Módulo de Encaminamiento para Simulador de Redes”. El objeto de este Proyecto es, pues, el desarrollo del módulo de encaminamiento para dicho Simulador. El módulo de encaminamiento está basado en el Protocolo **BGP** (*Border Gateway Protocol*) en su versión número cuatro, descrito por la Norma **RFC 1771**. Por su parte, el Simulador, desarrollado en el Área de Ingeniería Telemática, es un simulador orientado a eventos.

1.1. ¿Qué es BGP?

BGP (Protocolo de Pasarela Frontera) es un protocolo orientado a conexión para el encaminamiento, que genera un grafo de Sistemas Autónomos (ASs) basado en la información que intercambian los routers BGP que podemos ver como un árbol de encaminamiento. Basándonos en BGP podemos ver Internet como un gran árbol de Sistemas Autónomos, cada uno de ellos con un identificador único. Cada conexión entre dos Sistemas Autónomos vecinos forma un camino, y el conjunto de caminos que llevan de un determinado Sistema Autónomo a otro constituyen una ruta.

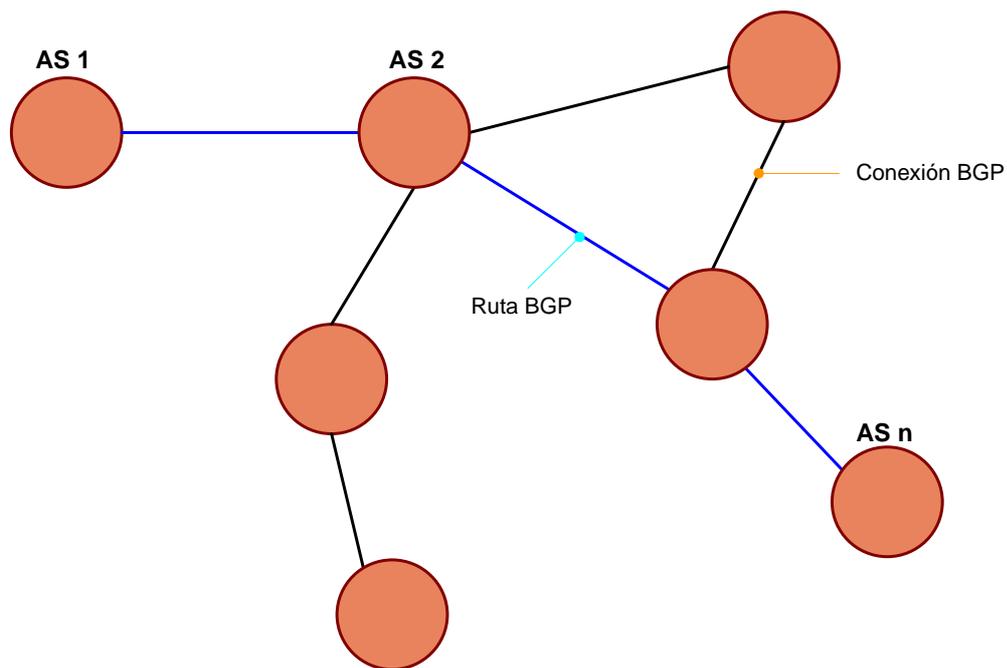


Figura 1. Sistemas Autónomos.

BGP es un protocolo de vector de rutas utilizado para el intercambio de información de encaminamiento entre Sistemas Autónomos. Dicho vector está constituido por los identificadores de los Sistemas Autónomos por los que ha pasado la información de encaminamiento, y se emplea para evitar bucles. Por otra parte, un nodo BGP sólo establece conexiones con sus nodos vecinos, y es el intercambio de información de encaminamiento entre vecinos el que permite la propagación de dicha información a todos los Sistemas Autónomos.

Desde el punto de vista de este protocolo, una ruta es la unidad de información

que relaciona un destino con el conjunto de atributos del camino hacia dicho destino. Una ruta, por tanto, se compone del destino al que va dirigida (dirección IP en formato CIDR del nodo BGP que supone la puerta de entrada en el Sistema Autónomo destino) junto con una serie de atributos que describen el camino hacia el destino; algunos de estos atributos son el conjunto de identificadores de los Sistemas Autónomos que hay que atravesar hasta llegar al Sistema Autónomo destino, la preferencia local (del nodo que nos da a conocer la ruta en cuestión) por la ruta anunciada o la dirección del siguiente nodo al que enviaremos información con destino final el Sistema Autónomo anunciado. Los atributos pueden ser obligatorios u opcionales.

Las rutas se almacenan en las Bases de Información de Enrutamiento (RIBs *Route Information Bases*), estas RIBs se dividen en tres categorías o grupos, la RIB Local (*Loc-RIB*), la RIB de Entrada (*Adj-RIB-In*) y las RIBs de Salida (*Adj-RIB-Out*). La RIB Local almacena aquellas rutas que han sido seleccionadas como preferidas por un determinado nodo, ésta será la que se consulte para llegar a un determinado Sistema Autónomo. La RIB de Entrada es dónde se almacenan todas aquellas rutas que se aprenden de los nodos vecinos, de entre estas rutas se tomarán aquellas que forman parte de la RIB Local utilizando para ello un proceso de decisión de rutas, puede haber una o varias según la implementación. Las RIBs de Salida almacenan las rutas que se han seleccionado de la RIB de Entrada y que están, por tanto, en la RIB Local, y que deben ser anunciadas a los nodos BGP vecinos, de modo que así se propagarán las rutas favoritas y todos los nodos del sistema tendrán conocimiento de los diversos destinos accesibles, una vez se anuncian las rutas preferidas, éstas son borradas de las RIBs de Salida.

BGP provee mecanismos para que un nodo BGP pueda advertir a sus vecinos que una o varias rutas que fueron anunciadas previamente como buenas han dejado de ser válidas, bien porque ya no están operativas, bien porque se tiene una ruta mejor a un destino concreto, o bien porque una conexión con un nodo BGP vecino se ha cortado y por tanto se deben eliminar todas aquellas rutas que fueron aprendidas a través de dicha conexión. En el caso de tener una ruta mejor que una que ya fue anunciada no hay que indicar que esta última ya no es válida, basta con indicar la nueva ruta para que el nodo receptor de dicha información la seleccione a través de su proceso de decisión; este hecho da consistencia al protocolo, ya que la ruta que fue conocida inicialmente y que ya no es favorita sigue siendo válida y podría recurrirse a ella en el caso de que la favorita dejase de estar operativa.

En BGP se definen cuatro tipos de mensajes: OPEN, NOTIFICATION, KEEPALIVE y UPDATE. Todos los mensajes tienen los mismos campos cabecera, de entre los cuales hay un identificador del tipo de mensaje que permite al nodo receptor discriminar el tipo de mensaje que recibe y procesarlo del modo correcto. A continuación se describirá la funcionalidad de cada uno de ellos:

- **OPEN:** una vez que se ha establecido la conexión del protocolo de la capa de transporte sobre la que se sustenta la conexión BGP (TCP), el primer mensaje que se envía es de este tipo; en él se establecen los parámetros de la conexión BGP. Por tanto, con este mensaje se inicia una conexión BGP. Si el nodo receptor da por válido este mensaje, lo asiente con un mensaje KEEPALIVE.
- **NOTIFICATION:** este tipo de mensajes se envía cuando se detecta una condición de error. Por tanto este mensaje indica dicha condición y provoca el cierre de la conexión BGP previamente establecida.

- **KEEPALIVE:** este tipo de mensajes se utiliza para mantener una conexión activa cuando no se producen cambios en la red, es decir, cuando no hay rutas nuevas o rutas que han dejado de ser válidas. BGP establece una temporización para determinar si una conexión está activa, de modo que si en ese tiempo no se recibe ningún mensaje se da la conexión por muerta y se cierra, los mensajes KEEPALIVE evitan que se active el proceso anterior sin que realmente sea necesario. Estos mensajes no contienen información alguna, sólo tienen cabecera.
- **UPDATE:** este tipo de mensajes se emplea para transferir información de enrutamiento entre nodos BGP. A través de éstos un determinado nodo anuncia a otro nodo vecino con el que tiene establecida una conexión BGP la presencia de una nueva ruta válida, o bien que una o varias rutas ya no están operativas. Es aquí donde se introducen las rutas de las RIBs de Salida y es de aquí de donde se obtienen las rutas que conforman la RIB de Entrada. En un mensaje UPDATE sólo se anuncia una ruta válida, de modo que si se tienen varias habrá que enviar tantos mensajes como rutas válidas para anunciar se tenga. Antes de anunciar una determinada ruta que ha sido seleccionada a través del proceso de decisión, dicha ruta debe ser actualizada con los valores del nodo considerado como se verá más adelante.

Cuando dos nodos BGP establecen una conexión se envían todas las rutas válidas (todo el contenido de la RIB Local) y luego sólo se enviarán mensajes UPDATE en el caso de haya nuevas rutas o haya rutas que han dejado de ser válidas, es decir, sólo se manda información de la red cuando se producen cambios en ésta, no hay un refresco periódico de las tablas de rutas, ya que sería un envío redundante y un uso innecesario de los recursos de red, es por esto que se hace necesario el envío de mensajes KEEPALIVE para mantener activas las conexiones.

En BGP el criterio de selección de rutas se basa en la preferencia local por dicha ruta. La preferencia, como se ha indicado anteriormente, es un atributo de la ruta, inicialmente es máxima (con un valor de 65535) y a medida que va atravesando Sistemas Autónomos se va decrementando. Por tanto, a la hora de elegir entre dos rutas que llevan al mismo Sistema Autónomo, se elegirá aquella que tenga un mayor valor en dicho atributo. Si dos rutas tienen el mismo valor para este atributo entonces se tomará la decisión en función del camino recorrido y más concretamente en función del último valor de dicho camino, que identifica el Sistema Autónomo vecino por el que ha sido conocida cada una de las rutas, la seleccionada será la que tenga un menor valor en dicho campo.

Para más detalles sobre este Protocolo se recomienda la consulta de la Norma RFC 1771 que lo describe.

1.2. Modelado del Sistema

El modelo del sistema con el que se va a trabajar lo podemos caracterizar como:

- Dinámico: puesto que su estado cambiará a lo largo de la simulación.
- Determinista: puesto que el comportamiento del sistema está predefinido y

podemos prever su comportamiento frente a los distintos eventos que le pueden afectar. No obstante, también se puede pensar en un cierto carácter estocástico en el caso de los errores en transmisión, pues aunque, como se explicará más adelante, a priori se sabe a partir de qué instante de la simulación se producirá, no se conoce ni el instante exacto ni a qué nodo afecta ni a cuál de sus conexiones.

- Discreto: por cuanto que la evolución del sistema se produce en instantes concretos.

1.3. Simulador Orientado a Eventos

Como se ha indicado anteriormente, el Simulador es un Simulador Orientado a Eventos. Un evento es un suceso elemental que altera el estado del Sistema y en función del evento concreto que se produzca, el Sistema evolucionará en un sentido u otro. Podemos clasificar los eventos en los siguientes grupos:

- Transmisión de un mensaje BGP: estos eventos se encolan cada vez que se desea enviar un mensaje; a su vez provocan el evento “Recepción de un mensaje BGP”.
- Recepción de un mensaje BGP: este evento provocará un funcionamiento determinado en función del tipo de mensaje BGP de que se trate (OPEN, NOTIFICATION, KEEPALIVE o UPDATE). Como ya se ha dicho, se generan como consecuencia del envío de un mensaje en el nodo destino.
- Inicio de un nodo BGP (start): este evento habilita a un determinado nodo BGP para la recepción/transmisión de mensajes BGP. Se genera por configuración.
- Parada de un nodo BGP (stop): este evento deshabilita a un determinado nodo BGP para la recepción/transmisión de mensajes BGP. Este evento se emplea para la simulación de la caída de un nodo BGP. Se genera por configuración.
- Caída de un enlace (crashLink): este evento deshabilita a un nodo BGP para la transmisión de mensajes a través de una determinada conexión. Este evento se utiliza para simular la caída de un enlace entre nodos BGP. Se genera por configuración.
- Comprobación de los *Hold Timer* (checkHoldTime): verificación de los temporizadores de las distintas conexiones de un nodo BGP. Con este evento se comprueba que las distintas conexiones de un nodo BGP estén activas. Este evento se genera cada vez que recibe un mensaje que no sea NOTIFICATION.

De la base de simulación desarrollada anteriormente a este Proyecto, se tiene el Manejador de eventos (clase *Scheduler*), éste emplea una cola de eventos donde se almacenan los eventos generados hasta que son activados. Los eventos mencionados anteriormente generan a su vez nuevos eventos y de este modo irá evolucionando la simulación. Los eventos de partida son el inicio de los distintos nodos que conforman el Sistema y el envío de un mensaje OPEN en uno de los extremos de cada una de las conexiones, además, se encolarán los eventos concernientes a la parada de los distintos nodos del Sistema según convenga.

1.4. Herramientas de desarrollo

Para el desarrollo del presente proyecto se ha empleado el lenguaje de programación C++, junto con un editor de distribución libre, *PSPad*. Para la compilación y ejecución del Simulador se ha empleado *Cygwin B20*, que emula un entorno Unix sobre el SO Windows XP.

1.5. Estructura de la Memoria

A continuación se va a describir cómo está estructurada la presente memoria, de este modo el lector se podrá hacer una idea general de la misma y sabrá el sentido de la misma.

Inicialmente se presenta una librería auxiliar (*libr.h*), en la que se describen las estructuras que se emplearán más tarde y que sirven de apoyo para una programación más sencilla. Posteriormente se describen las clases que soportarán los mensajes BGP. Seguidamente se verá la clase que soporta a los gestores de conexión (*gestConn*), que como se verá, gestionan cada una de las conexiones de un nodo BGP. Finalmente se presenta la clase que implementa a los nodos BGP y que, lógicamente, es la clase principal de este desarrollo (*BGPnode*). Con esto tendremos descrita toda la programación que da soporte al Simulador.

Tras lo anterior se presentan las distintas pruebas que se han realizado para comprobar el buen funcionamiento del Simulador. En primer lugar se prueba un sistema sencillo, y luego se harán pruebas de un sistema con topología más compleja (un anillo), en el que tras comprobar el funcionamiento normal, se probará la consistencia frente a caídas de nodos, caídas de enlaces y errores en transmisión.