



Proyecto Fase I

1. Objetivo

El objetivo de esta fase es experimentar con comunicación con sockets TCP y UDP en preparación para fases más complejas del proyecto.

2. Descripción


Se tienen dos tipos de nodos:

- Enrutadores de borde (entre Sistemas Autónomos)
- Enrutadores internos (a lo interno de un Sistema Autónomo)

Los enrutadores de borde se comunican entre sí con el protocolo *pseudoBGP* (que iremos construyendo en el curso), el cual utiliza **TCP**. Los enrutadores internos se comunican con diferentes protocolos propios (que también se construirán en el curso), pero todos utilizan los servicios de capa de transporte a través de **UDP**.

Para esta entrega van a hacer lo mismo para ambos tipos de nodos. La única diferencia va a ser que en un caso van a usar TCP y en el otro van a usar UDP.

Las operaciones que deben programar son las siguientes:

- Crear un nodo
- Borrar un nodo
- Comunicar cualesquiera dos nodos *del mismo tipo*. Específicamente: 
 - o Recibir un mensaje
 - o Decodificar el mensaje e imprimir en consola el resultado de la decodificación
 - o Responder el mensaje

Aspectos adicionales:

- En el caso de los enrutadores de borde (TCP), deben mantener la comunicación entre los dos nodos abierta hasta que se reciba un comando de desconexión
- Cada nodo debe almacenar las tablas de alcanzabilidad construidas con la información recibida

Detalles

2.1. Creación de nodo

Un nodo se crea digitando el comando `nodo-<X>` en la consola de la máquina en la que va a estar ejecutándose, donde `<X>` puede ser *pseudoBGP* o *intAS*.

Los parámetros son los siguientes:

- `direccionNodo` : Dirección IP
- `puertoNodo`: Puerto TCP o UDP en la que el nodo va a **recibir** conexiones.



Por ejemplo:

- creaNodo-pseudoBGP 10.10.120.7 940

Crea un nodo de tipo pseudoBGP en la dirección IP 10.10.120.7, en el puerto 940

2.2. Borrado de nodo

Una vez que un nodo esté corriendo, puede terminar de las siguientes dos maneras:

- **Ordenado:** Entre el menú de opciones del nodo debe estar cerrar el nodo. Si es seleccionado, el nodo debe enviar mensajes de cierre a todas las entidades con las que esté en comunicación en ese momento, y luego apagarse (concluir el proceso). El mensaje de cierre debe contener únicamente un byte como dato, con el número 0.
- **Por evento catastrófico:** Pérdida de conexión total, crash del proceso, y otros. Si otros nodos pares se dan cuenta de este evento, deben borrarlo de sus tablas de conexión.

2.3. Comunicación entre nodos del mismo tipo

Los nodos pueden enviar y recibir mensajes.

Emisión de mensaje

Entre las opciones de menú de un proceso nodo debe estar "enviar mensaje". Los únicos mensajes que se pueden enviar van a estar estructurados según se describe a continuación.

Primero se debe leer la dirección IP y el puerto del destinatario del mensaje. Esto implica leer una dirección IP, una máscara y una dirección de puerto. En el caso de TCP, una vez leídos los datos se debe comprobar si no se tiene una conexión ya abierta a dicho destino, y si es el caso, usar esa.

En segundo lugar, como se va a leer una lista de elementos, se debe determinar el número de elementos en esa lista (n).

Finalmente se debe leer la lista de tripletas de la siguiente manera:

- Dirección de red 1, máscara 1, costo 1
- Dirección de red 2, máscara 2, costo 2
- ...
- Dirección de red n, máscara n, costo n

El mensaje que se construye a partir de estos datos debe tener la siguiente estructura:

- 2 bytes con el valor de n
- 4 bytes de dirección de red 1, 1 byte de máscara 1, 3 bytes de costo 1
- 4 bytes de dirección de red 2, 1 byte de máscara 2, 3 bytes de costo 2
- ...
- 4 bytes de dirección de red n, 1 byte de máscara n, 3 bytes de costo n

El mensaje debe enviarse via TCP o UDP según corresponda al nodo destino.



Se debe imprimir el resultado del intento de transmisión según los códigos de error que devuelva la operación de envío.

Recepción de mensaje

La recepción de un mensaje no corresponde a ninguna opción de menú. Todos los nodos deben estar escuchando posibles conexiones en el puerto con el que fueron creados.

Si se recibe un mensaje de longitud 1, con el número 0 en el contenido, se deben borrar todas las entradas en las tablas de alcanzabilidad que fueron recibidas desde esta fuente.

2.4. Almacenamiento de tablas de alcanzabilidad

Cuando se recibe un mensaje de lista de alcanzabilidad, se debe almacenar. Las estructuras de almacenamiento deben diseñarse tomando en cuenta lo siguiente:

- Se debe poder asociar las entradas con su origen (el nodo que las envió)
- Para cualquier red en la lista de alcanzabilidad, únicamente se debe guardar aquella con el menor costo. Si el costo de una red es corregido por uno menor, se debe actualizar el origen de la entrada de red.

Adicionalmente, en el menú de un nodo debe estar la opción de imprimir la tabla actual de alcanzabilidad.

3. Testing

Deben crear al menos tres nodos de cada tipo, e intercambiar al menos 5 mensajes de alcanzabilidad por enlace, incluyendo entradas de menor y mayor costo para las mismas redes. Verifiquen que las tablas de alcanzabilidad sean consistentes.

4. Entregables

- Código fuente y código ejecutable del enrutador pseudoBGP
- Documentación de uso del programa
- Documentación de las pruebas.

Fecha de entrega: 19 de setiembre de 2018.