



## Proyecto

### Fase III

### Enrutamiento

#### 1. Objetivo

El objetivo de esta fase es explorar aspectos de enrutamiento.

#### 2. Descripción

Un grupo de nodos comunicándose por UDP deben encontrar formas de encontrar las mejores rutas, de forma adaptativa, entre cualesquiera dos nodos.

La topología inicial de la red se lee de un archivo de configuración CSV, donde cada línea tiene el siguiente formato y representa una relación de vecindad directa entre dos nodos  $N_i$  y  $N_j$ :

IP  $N_i$ , máscara  $N_i$ , puerto  $N_i$ , IP  $N_j$ , máscara  $N_j$ , puerto  $N_j$ , distancia<sub>ij</sub>

Las máscaras están dadas por un número entre 2 y 30. La distancia es un número entero entre 20 y 100.

Para simplificar, los nodos leen el archivo completo, pero únicamente deben usar las entradas de vecindad que les correspondan. Una implementación más adaptable puede crear un servidor de vecinos, que ante la consulta de un nodo le devuelva la información de sus vecinos.

Cada nodo se crea independientemente, con tres parámetros: su dirección IP, su máscara, y el puerto en el que va a operar. Para efectos de pruebas, pueden activar los nodos uno por uno, o tener un despachador que con base a un archivo de configuración los levante todos (chequeando que efectivamente los haya logrado levantar, por supuesto). Sin embargo, eventualmente se intentará tener una red que combine nodos de todos los grupos, por lo que posiblemente solo deban realizar una activación parcial. Note que la red debería poder manejar cientos de nodos. En ese caso, la profesora proveerá un archivo de topología, y le indicará a cada grupo cuál subgrupo de nodos está bajo su administración.

Una vez que cada nodo ha descubierto quienes son sus vecinos en teoría, pueden comenzar a intercambiar información de enrutamiento. Deben considerar todos los posibles errores que podrían ocurrir y actuar acorde a cada uno de ellos.

Vamos a utilizar Bellman-Ford distribuido para construir las rutas. Las actualizaciones deben contener dos bytes indicando la cantidad de tuplas en la actualización y esa cantidad de tuplas a continuación. Cada tupla está organizada de la siguiente forma:

- Dirección de nodo reportado (IP – 4 bytes, máscara - 1 byte, puerto – 2 bytes), distancia (esta tiene que ser mayor a 20, pero puede ser muy superior a 100, dado que ya es una combinación de saltos)

Las actualizaciones deben repetirse a intervalos de 30 segundos. NUNCA se debe incluir en la actualización el nodo que la recibe.

Los nodos pueden desaparecer, cambiar su costo, fallar enlaces y otros percances. Si se detecta un problema serio propio (por ejemplo la desaparición de un nodo) se debe enviar un mensaje especial por inundación a toda la red. Si se cambia el costo de un enlace, el nodo que origina el cambio se lo debe notificar al vecino afectado únicamente. Deben ponerse de acuerdo en el formato de estos mensajes.



**UNIVERSIDAD DE COSTA RICA**  
**Escuela de Ciencias de la Computación e Informática**  
**CI-1320 Redes de Computadoras**  
**II Ciclo 2018**



En cada nodo deben poder proveer las siguientes herramientas (menú):

- Cambiar el costo del enlace con un vecino dado
- Matar al nodo
- Imprimir la tabla de enrutamiento vigente

Cada nodo debe mantener una bitácora donde anote aquellas las actualizaciones recibidas que causaron cambios en la tabla de enrutamiento, eventos detectados, eventos causados y otros.

Finalmente, la red debe proveer la capacidad de enviar paquetes a través de las rutas establecidas. Para ello, el menú debe contar con una opción de enviar mensaje a un destino dado dentro de la red. Conforme el mensaje se vaya propagando, deben imprimir su camino.

**Entregables**

- Código fuente
- Descripción de todas las decisiones importantes de diseño (al menos 5)
- Requerimientos adicionales descubiertos durante la programación
- Presentación (10 filminas)

**Fecha de entrega: 14 de noviembre de 2018.**