

Informe Forwarding Redes

Alexander Reisenegger

Se utiliza una estructura de Router que contiene ip, puerto, socket, tabla de rutas y diccionario de round-robin.

Para crear el round-robin se utiliza un diccionario que guarda como llave un par que corresponde a los límites del área de la red accedida, y la última redirección que se eligió para dicha área. Cuando se quiere redirigir un mensaje, se busca en el diccionario si fue accedida el área antes, y en caso de serlo, se guarda una variable “r” en 1 (0 si no está en el diccionario). Luego se recorre el archivo de la tabla hasta llegar a la línea correspondiente a la última redirección para esa área, y se reasigna r a 0, para indicar que ya se llegó a la última ruta tomada, luego de esto se continúa el recorrido hasta encontrar una ruta viable, y si no se encuentra ninguna, se recorre una vez más para empezar desde cero y que sea circular el round robin

El archivo para los routers sin TTL se llama router.py

Sin TTL:

- Al configurar mal las tablas de rutas, y devolver el mensaje a r1, se queda en un ciclo dando vueltas el mensaje
 - Al utilizar round robin, depende del camino que se tome la cantidad de saltos, no necesariamente toma el camino más corto, si no que el siguiente al último en ser utilizado, por lo que hay puede haber variación en la cantidad de saltos cada vez que se envía el mensaje
 - Ocurre lo mismo con las distintas rutas al agregar r0 y r6 como fue indicado, dependiendo de la selección del round robin, el largo de la ruta será variable, incluso puede que un paquete pase 2 veces por un mismo router.
- Las tablas creadas para añadir r0 y r6 son las siguientes

R0

```
127.0.0.1 8881 8886 127.0.0.1 8881
127.0.0.1 8881 8886 127.0.0.1 8882
```

R1

```
127.0.0.1 8882 8886 127.0.0.1 8882
127.0.0.1 8882 8886 127.0.0.1 8880
127.0.0.1 8880 8880 127.0.0.1 8880
```

R2

```
127.0.0.1 8880 8881 127.0.0.1 8881
127.0.0.1 8880 8881 127.0.0.1 8880
127.0.0.1 8883 8886 127.0.0.1 8883
127.0.0.1 8883 8886 127.0.0.1 8884
127.0.0.1 8883 8886 127.0.0.1 8886
```

R3

```
127.0.0.1 8880 8882 127.0.0.1 8882
127.0.0.1 8880 8882 127.0.0.1 8885
127.0.0.1 8880 8882 127.0.0.1 8880
127.0.0.1 8884 8886 127.0.0.1 8882
127.0.0.1 8884 8886 127.0.0.1 8885
127.0.0.1 8884 8886 127.0.0.1 8886
```

R4

```
127.0.0.1 8880 8883 127.0.0.1 8882
127.0.0.1 8880 8883 127.0.0.1 8885
```

127.0.0.1 8885 8886 127.0.0.1 8885

R5

127.0.0.1 8880 8884 127.0.0.1 8884

127.0.0.1 8880 8884 127.0.0.1 8883

127.0.0.1 8886 8886 127.0.0.1 8883

R6

127.0.0.1 8880 8885 127.0.0.1 8882

127.0.0.1 8880 8885 127.0.0.1 8883

El archivo del mini-internet con TTL se llama routerTTL.py

Con TTL:

- Si se utiliza TTL cuando se introduce un error en la tabla de rutas, ya no se queda dando vueltas en la red, si no que se detiene despues de 10 saltos.

Para la prueba del archivo, como no se entrega el archivo como argumento, se creo un archivo message.txt, y se asume ese nombre

Al enviar multiples lineas, el orden de ellas llega distinto al original, esto se debe a que toman diferentes rutas, algunas mas largas que otras.