Laboratorio 4: Capa de Red

Redes de Sensores Inalámbricos
IIE, Facultad de Ingeniería, UDELAR

Tabla de Contenidos.

1 Introducción

En este laboratorio se creará una red IPv6/RPL formada por un nodo *Border Router* y varios nodos *Cliente* para estudiar su formación y mantenimiento. Se simulará la red utilizando la herramienta Cooja (disponible en el repositorio de Contiki) y analizando los paquetes intercambiados por los nodos. La simulación se realizará utilizando los nodos z1, ya que los RE-Mote no están soportados. Se analizará el camino que recorre un paquete y los tiempos involucrados. En ediciones anteriores del curso, se ponía una red en funcionamiento con nodos reales, re-compilando las aplicaciones para RE-Mote. Dicha actividad está plasmada en la Tarea 3, pero **no se realizará en este laboratorio de la edición 2022 del curso**.

2 Objetivo

Comprender el funcionamiento del armado y mantenimiento de rutas en una RSI mediante el análisis de los mensajes intercambiados por los nodos y su interpretación de acuerdo con lo visto en las clases teóricas de RPL.

2.1 Objetivos de aprendizaje

Se espera que el estudiante al final del laboratorio sea capaz de:

- Utilizar el simulador Cooja para analizar el comportamiento de una red, visualizando los mensajes intercambiados.
- Utilizar Wireshark como complemento de análisis de tráfico de una red.
- Interpretar los distintos mensajes de RPL intercambiados por los nodos.
- Ver cómo la posición relativa de un nodo, para cierta topología de enrutamiento, impacta en el tiempo de respuesta.
- Ver el comportamiento de una red real a nivel de RPL (**suspendida en 2022**).

3 Lecturas y actividades previas

Durante el laboratorio se analizará una red formada por 3 nodos *Cliente* y un *Border Router*. Los nodos *Cliente* enviarán mensajes a otro nodo destino. La dirección de destino podrá ser modificada con la red funcionando.
Se pide como actividad previa desarrollar el código de los nodos *Cliente* y crear la simulación según se describe en los siguientes pasos:

Actividad Previa 1

1) Familiarizarse con Cooja, la herramienta de simulación de Contiki. Para eso se recomienda:
    a) Leer el material de referencia disponible en el wiki [1] (pueden ignorar lo referido a external tools de la sección "*Getting started*" y la sección entera "*Configuration Wizard*").
    b) Simular la parte 1 del laboratorio 2 (multicast), siguiendo los pasos de "*Create a Hello World simulation*" de [1].
2) Para la simulación en Cooja se dejará la radio prendida en recepción (cuando no se esté transmitiendo) para facilitar la visualización e identificación de paquetes. Esto se logra modificando el archivo `project-conf.h` de las aplicaciones, tanto del *Border Router* como de los demás nodos, agregando las líneas indicadas a continuación:

```
#undef NETSTACK_CONF_RDC
#define NETSTACK_CONF_RDC      nullrdc_driver
```

   De esta manera estamos desactivando el protocolo de ciclado de la radio.
3) Crear una nueva simulación y agregar un *Border Router*. Para ello:
    a) Agregar un nodo tipo *z1*.
    b) Seleccionar el código `border-router.c` de la carpeta `contiki/examples/ipv6/rpl-border-router`. Al compilarlo se creará el nodo *Border Router* con `ID=1`.
    c) En la ventana “Network”, hacer clic derecho en el *Border Router* y seleccionar en el menú “Mote tools for z1/Serial socket (SERVER)”. Una

nueva ventana debe aparecer diciendo que el nodo *Border Router* está escuchando en el puerto 60001.

4) Agregar a la simulación tres nodos *Cliente*.
   Para ello:
   a) Copiar los códigos fuente desarrollados en la parte 3 del laboratorio 2 (unicast) a una nueva carpeta.
   b) RPL debe estar activado y en el laboratorio 2 había sido desactivado. Por tanto, se debe modificar el código copiado para que RPL esté activado.
   c) Agregar un nodo tipo *z1*.
   d) Elegir el código modificado en la parte b) y crear 3 nodos.
5) Iniciar la simulación.
6) Tomar nota de la dirección IPv6 de los nodos creados, y actualizar la dirección de destino de los paquetes unicast que envían los nodos *Cliente* por la de otro nodo que no sea el *Border Router*.
7) Recargar la simulación para recompilar el código.
8) Abrir una terminal en la carpeta donde se encuentra border router y ejecutar

   ```
   $ make connect-router-cooja TARGET=z1
   ```

   Esto iniciará una herramienta llamada `tunslip6` que configura una interfaz en la capa IP de Linux y conecta esta interfaz al *Border Router* a través de un socket. La interfaz de Linux estará configurada de manera que todo el tráfico IP destinado a direcciones que comienzan con **`fd00`** irán a la interfaz en Cooja.
9) Observar las direcciones IPv6 asignadas a cada nodo y verificar que los nodos envían mensajes a un nodo determinado y éste los recibe mostrándose en la ventana de Cooja llamada `mote output`.

Sugerencia: Modificar el radio de interferencia de los nodos para que sea pequeño en relación al alcance. Para ello hay que hacer click izquierdo sobre cualquiera de los nodos, y presionar donde dice “Change transmission ranges”. Bajar el INT range a 55 m.

Actividad previa 2

Al código de los nodos se le adicionará la funcionalidad de modificación de la dirección de destino a la que envía los mensajes, utilizando el botón de usuario. El nodo al que deben ser enviados los paquetes tendrá el `ID=n`, donde `n` es el número de veces que se presione el botón[1]. Es decir, cuando el usuario quiera que los mensajes se envien al nodo 3 debe presionar 3 veces el botón de usuario. Si luego quiere cambiar el destino al nodo 2, presionará 2 veces. Para ello:

1. Agregar un nuevo proceso al código de los nodos que cuente el número de veces que cada usuario presiona el botón. Para detectar cuántas veces se presionó, se implementará un timeout luego del cual se dará por finalizado el ingreso del usuario. Se sugiere repasar las actividades del laboratorio 1, en particular las indicaciones para realizar simulaciones con nodos z1.
2. El nuevo proceso creado, cada vez que alguien realiza una nueva configuración de destino, envía el número de veces que se presionó el botón mediante *post* de un evento al proceso principal.

[1] No interesa el número total de veces que se presionó el botón desde el inicio del programa, sino el número de veces que el usuario presiona cada vez que quiere configurar el destino.

3. Comprobar que luego de realizado este cambio es posible cambiar dinámicamente el destino de los mensajes mientras se ejecuta la simulación. Investigar cómo hacer para presionar el botón de un nodo simulado en Cooja, y limitar la velocidad de la simulación al 100% para que sea posible probar la funcionalidad del botón.

4 Procedimientos y Tareas

4.1 Tarea 1

Utilizando la simulación de la actividad previa se pide:

1. Analizar los mensajes RPL intercambiados entre los nodos (en la ventana Radio messages) y dibujar un diagrama del grafo (DODAG) de la red indicando para cada nodo el rank y el padre preferido elegido.
2. Observar el mecanismo utilizado por el root del DODAG (border-router en este caso) para propagar el prefijo global.
3. Pulsar `n` veces el botón de un nodo alejado del border router. Observar la secuencia de paquetes intercambiados por los nodos verificando la ruta utilizada hasta llegar al nodo con `ID=n`.

NOTA: Se recomienda utilizar Wireshark para analizar los paquetes a partir del archivo PCAP generado por Cooja. Para ello en la ventana *Radio messages* de *Analyzer* seleccionar “6lowpan Analyzer with PCAP”. Los archivos PCAP se guardan en el directorio:

```
./contiki/tools/cooja/build/radiolog*.pcap
```

4.2 Tarea 2

Realizar pings al border router y a diferentes nodos de la red, ejecutando por ejemplo desde una ventana de terminal:

```
$ ping6 fd00::c30c:0:0:nodeid
```

donde `nodeid` es el ID del nodo al que se le quiere hacer ping.

1. Observar las diferencias de tiempos de respuesta entre border-router y el resto de los nodos.
2. ¿Qué valores de período de vida (TTL) se observan para los diferentes nodos de la red? ¿Qué pasa con el tiempo de respuesta?
3. Identificar los paquetes ICMP correspondientes a los pings que se hicieron.

4.3 Tarea 3 (suspendida en 2022)

En esta parte se creará una red real utilizando todos los nodos disponibles de la clase. Se dispondrá de un nodo extra que tendrá el rol de *Border Router*. Los grupos utilizarán el código de la tarea 1 del laboratorio 2 (multicast).

Se deberá copiar y modificar dicho código para que RPL esté activado.

Se realizará un diagrama colectivamente en el pizarrón siguiendo el proceso de formación de la red. Para ello se modificará el código de los nodos para habilitar el uso de `printf` previstos para debug. Esto normalmente se realiza en Contiki modificando, en los archivos fuentes de interés, el siguiente macro (investigar qué valor definir):

```
#define DEBUG DEBUG_NONE
```

Para esta tarea se recomienda habilitar los `printf` de `rpl-dag.c` en la carpeta `contiki/core/net/rpl`.

5 Entregables

Se deberá entregar un archivo comprimido que contenga lo siguiente:

1. Carpeta Tarea 1 incluyendo Makefile, project-conf.h, .c, .h y simulaciones .csc.
2. El diagrama de la red indicando el rank de cada nodo y el padre preferido.
3. Un archivo de texto incluyendo los resultados de los pings a distintos nodos de la red realizados en la tarea 2 y las diferencias observadas.

Es imprescindible que el código esté comentado adecuadamente: cada línea o líneas consecutivas que sean agregadas, deberán ser acompañadas de un comentario que explique su propósito.

Las entregas se realizarán a través de la **plataforma EVA del curso**.

6 Referencias

[1] “Contiki Wiki - An Introduction to Cooja” [Online]. Available: https://github.com/contiki-os/contiki/wiki/An-Introduction-to-Cooja