

Laboratorio 2: Comunicación UDP en IPv6

1 Introducción

En este laboratorio se realizará una primera aproximación al uso de comunicación inalámbrica entre nodos de una red de sensores utilizando Contiki-NG. Se utilizarán diferentes canales de radio y se experimentará con mensajes unicast y multicast.

El trabajo en este laboratorio es grupal, sin embargo sugerimos que cada participante trabaje con su propia computadora, implementando las funcionalidades discutidas dentro del grupo.

2 Objetivos

2.1 Objetivos generales

- Familiarizarse con el envío de mensajes de radio en Contiki.
- Profundizar en el uso de Contiki-NG y la modificación de los parámetros por defecto del sistema operativo.
- Familiarizarse con las simulaciones en la herramienta Cooja.

2.2 Objetivos específicos

- Comprender el manejo básico de `make` para compilación de aplicaciones de Contiki-NG (redefinición de parámetros por defecto).
- Uso básico de la radio y la configuración del stack de comunicaciones.
- Comprensión del uso de la API UDP de Contiki-NG.
- Realizar simulaciones en el entorno Cooja.

3 Fundamentos

Estudiar en el libro [IoT in five days](#) la Sección 4.1 - “Addressing and Radio Frequency Basics” donde se tratan los temas básicos de direccionamiento y la comunicación de RF. En el libro se hace referencia, entre otras cosas, a diversos parámetros configurables. Para ver cuáles son dichos parámetros en Contiki-NG referirse a la [Radio API](#). Estudiar de [Programming Contiki-NG](#) la sección [UDP communication](#) donde se introduce `simple-udp` y se hace referencia a su documentación [A simple UDP API](#). También se sugiere ver el ejemplo de uso en [examples/rpl-udp](#) (disponible en el repositorio de Contiki-NG).

4 Actividades

4.1 Tarea: Envío de mensajes UDP multicast

1. Descargar el archivo `udp-multicast.c` y copiarlo a una nueva carpeta. Por ejemplo, `./contiki-ng/compartida/Laboratorio_2/Tarea_1`.
2. Generar el `Makefile` (puede ser útil basarse en algún ejemplo de `contiki-ng/examples`).
3. Generar una simulación en Cooja de la siguiente manera:
 - a. Abrir el simulador Cooja ejecutando `./contiker cooja` o el comando `cooja` desde dentro del contenedor `contiker`.
 - b. Crear una nueva simulación y agregar tres nodos de arquitectura `cooja` con el código `udp-multicast.c`. En caso de dudas consultar la documentación [Running Contiki-NG in Cooja](#).
 - c. En la opción *Speed limit*: seleccionar `1x`.
 - d. Guardar la simulación (*File -> Save simulation as...*).
4. Correr la simulación (*Start/Pause*).
5. Observar los registros (*logs*) en la ventana *Mote output*.
6. Observar los LEDs (botón derecho sobre los nodos, *Show LEDs on <mote>*)¹.
7. Explorar en el simulador qué otras cosas permite analizar.

Nota: El archivo de simulación contiene las ventanas abiertas (y su lugar en la pantalla); puede ser útil elegir qué se está visualizando al momento de guardar la simulación.

4.2 Tarea: Envío de mensajes UDP unicast

1. A partir del ejemplo `udp-multicast.c` generar dos programas más: `udp-server.c` y `udp-client.c`.
2. El nodo `udp-server.c` debe modificarse para lo siguiente:
 - a. El proceso principal debe simplemente inicializar el nodo y enviar un único mensaje multicast notificándose como servidor e indicando el número de grupo (por ejemplo con el texto “¡Soy el servidor del grupo X!”).
 - b. Cada vez que recibe un mensaje debe contestar con otro acusando recibo de recepción (por ejemplo con el texto “Recibido.” pero pueden ser más creativos).
3. El nodo `udp-client.c` debe enviar periódicamente mensajes al nodo servidor. Para esto:
 - a. Configurarlos para enviar mensajes a una IP cualquiera.
 - b. Correr la simulación de más abajo. Tomar nota de las direcciones MAC e IP con las que quedó configurado el servidor.
 - c. Configurar el nodo `udp-client.c` para que la IP de destino sea la que tiene el nodo servidor.
4. Generar una nueva simulación con tres nodos: El primero será a partir del código `udp-multicast.c`, el segundo a partir del código `udp-server.c` y el tercero a partir del código `udp-client.c`.
5. Correr la simulación y utilizar el monitor serial de Cooja para obtener las direcciones MAC e IP del nodo servidor. Para esto utilizar el `Shell` agregándolo al `Makefile`. Realizar las modificaciones necesarias en el archivo `udp-client.c` y recargar la simulación.

¹Puede que el color de los LEDs no coincida con lo que se espera viendo el código; parecería que los desarrolladores de Contiki-NG/Cooja no le prestaron mucha importancia a este tema.

4.3 Tarea: Prueba en hardware

1. Probar la tarea anterior en hardware. En caso de contar únicamente con dos *launchpad* programar por separado primero los dos nodos con `udp-multicast.c` y luego uno con `udp-client.c` y el otro con `udp-server.c`.
2. Al momento de programar los nodos se les puede asignar un número de identificador (*node_id*), agregando al comando `make` el parámetro `NODEID=0xn`, siendo `n` el número de nodo. Ejemplo:

```
make TARGET=cc26x0-cc13x0 BOARD=launchpad/cc1350 NODEID=0x05 laboratorio2.upload
```

3. Verificar y tomar nota de las direcciones MAC e IPv6 con las que quede configurado cada nodo. Analizar cómo se genera la dirección IP ya sea a partir de la dirección MAC o del `NODEID` utilizado en cada nodo. Verificar que la dirección IPv6 a la que el nodo cliente envía el mensaje unicast es correcta.
4. Configurar el canal de la radio para que cada grupo utilice un canal exclusivo. El número de canal del grupo X será `10+X`.

- a. Se puede agregar el siguiente código para verificarlo:

```
static uint8_t ch_num;
NETSTACK_RADIO.get_value(RADIO_PARAM_CHANNEL, &ch_num);
printf("RF_CHANNEL: %d\n", ch_num);
```

- b. Comprobar que en esa situación se reciben sólo los mensajes de los nodos del grupo (cambiar el canal de a un nodo a la vez para confirmar este hecho).
5. Realizar un ping a un nodo vecino:
 - a. Utilizar el `Shell` para conocer la IP de un nodo vecino. Consultar el tutorial [Shell](#) y el ejemplo disponible en `contiki-ng/examples/libs/shell`.
 - b. Enviar un ping a un nodo vecino.

4.4 Tarea: Descubrir el animal

Envío de mensajes UDP unicast

- a. Configurar el canal de la radio para que ahora utilice el canal 26, canal que utilizará un nodo servidor dispuesto por los docentes en el momento de la defensa.
- b. Modificar el programa para que mande mensajes unicast al nodo servidor, cuya dirección es `fd00::212:4b00:060d:1`. El servidor responderá solamente a mensajes unicast con un mensaje diferente en función de la dirección del nodo origen. Este mensaje contendrá el nombre de un animal.
- c. Programar el nodo con el número de identificador `NODEID=0xn`, siendo `n` el número del grupo.
- d. Durante la defensa observar la respuesta enviada por el servidor.

5 Entregables

Se deberá entregar un archivo comprimido que contenga el archivo de simulación y los archivos de la primera tarea (`Makefile`, `.c`, `.h`).

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

6 Referencias

- Colina, Antonio Linan, Alvaro Vives, Antoine Bagula, Marco Zennaro, y Ermanno Pietrosemoli. 2016. IoT in five Days. E-Book. <https://github.com/marcozennaro/IPv6-WSN-book/releases/>.

- [Radio API](#)
- [UDP communication](#)
- [A simple UDP API](#)
- Ejemplo de uso: [examples/rpl-udp](#)
- [Running Contiki-NG in Cooja](#)
- [Tutorial Shell](#)

Los materiales de este curso fueron parcialmente financiados por:



Co-funded by the
Erasmus+ Programme
of the European Union