Laboratorio 5: Subcapas MAC y RDC

Redes de Sensores Inalámbricos
IIE, Facultad de Ingeniería, UDELAR

Tabla de Contenidos.

1 Introducción

2 Objetivo
- 2.1 Objetivos de aprendizaje

3 Lecturas y actividades previas
- Lecturas previas
  - Fundamentos de TSCH
- Actividad previa
  - Estudio de código

5 Procedimientos y Tareas
- Tarea 1: nodo oculto
- Tarea 2: análisis de mecanismo de acceso al medio
  - 2.1 nullRDC
  - 2.2 ContikiMAC
  - 2.3 TSCH
    - 2.3.1 TSCH minimal
    - 2.3.2 TSCH Orchestra

6 Entregables

7 Referencias

1 Introducción

En este laboratorio se simulará una red de sensores inalámbricos utilizando la herramienta Cooja. Veremos el comportamiento de una red utilizando los distintos protocolos de las subcapas MAC y RDC vistos en el curso. Inicialmente estudiaremos el problema del nodo oculto. Luego analizaremos una red formada por nodos utilizando nullRDC, ContikiMAC y TSCH (*Time Slotted Channel Hopping*) con distintas configuraciones (minimal y Orchestra).
Para la aplicación se utilizará código suministrado por los docentes y se modificará según se indique. Se simulará en Cooja el envío periódico de datos.

2 Objetivo

Se busca familiarizar al estudiante con los mecanismos de acceso al medio y con los protocolos de las subcapas MAC y RDC vistos en el curso y su configuración.

2.1 Objetivos de aprendizaje

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

- Explicar el problema del nodo oculto y generarlo en simulación para su observación.
- Identificar las ventajas y desventajas de utilizar el *duty cycle* de la radio.
- Comprender el funcionamiento del método de acceso al medio ContikiMAC, diferenciando *broadcast* y *unicast*, e identificar los parámetros de configuración.
- Describir el funcionamiento de TSCH e identificar sus parámetros fundamentales (tiempo de *slot* y largo del *slotframe*).
- Analizar el envío de los parámetros de TSCH en *Enhaced Beacons* (EB) y el proceso de unión.
- Observar las diferencias entre TSCH minimal y Orchestra.
- Comparar los tres mecanismos de acceso al medio.

3 Lecturas y actividades previas

Lecturas previas

Fundamentos de TSCH

Se recomienda leer la documentación de la implementación de TSCH en Contiki de [1], en particular la sección "*Using TSCH*".

Actividad previa

Estudio de código

Estudiar el código suministrado: `lab-mac-sender.c, lab-mac-receiver.c, Makefile` y `project-conf.h`.
Los archivos `lab-mac-sender.c` y `lab-mac-receiver.c` están basados en los ejemplos `ipv6/simple-udp-rpl` e `ipv6/rpl-tsch` de Contiki. Al compilar estos archivos modificados, se seleccionará uno de tres modos de funcionamiento de la subcapa MAC:

- ContikiMAC
- TSCH minimal
- TSCH Orchestra

Para seleccionar el protocolo se utilizan las macros `MAKE_WITH_TSCH` y `MAKE_WITH_ORCHESTRA`, de modo tal que:

- para ContikiMAC se configurará
  `MAKE_WITH_TSCH = 0, MAKE_WITH_ORCHESTRA = 0`
- para TSCH minimal se configurará
  `MAKE_WITH_TSCH = 1, MAKE_WITH_ORCHESTRA = 0`
- para TSCH Orchestra se configurará
  `MAKE_WITH_TSCH = 1, MAKE_WITH_ORCHESTRA = 1`

Al momento de compilar debemos configurar las macros haciendo (por ejemplo para TSCH minimal en un nodo z1):

```
make MAKE_WITH_TSCH=1 MAKE_WITH_ORCHESTRA=0 TARGET=z1
```

En este caso se compilará todo el código a excepción de los fragmentos entre las guardas:

```
#if WITH_ORCHESTRA
/* código que se omite en este caso */
#endif /* WITH_ORCHESTRA */
```

Para configurar estos tres valores hay que modificar el `Makefile`, o ejecutar el comando `make` con los parámetros mencionados previamente.

El archivo `project-conf.h` suministrado contiene algunos parámetros de interés de la configuración de TSCH (minimal u Orchestra). Durante el laboratorio se modificarán algunos de estos parámetros y se analizará mediante simulaciones los efectos ocasionados.

Recordar que es importante ejecutar `make clean` antes de hacer una compilación que omita código que ya fue compilado, de otra manera se mantendrá parcialmente el código antiguo. Esto también se puede incluir en Cooja en la línea donde se especifica el comando para compilar y luego crear el nodo.

La actividad previa incluye inspeccionar los archivos suministrados y probar compilar seleccionando los diferentes protocolos o modos de comunicación.

Para complementar la lectura, y ver más detalles del funcionamiento, se sugieren las referencias [2], [3] y [4].

5 Procedimientos y Tareas

Para las simulaciones utilizaremos nodos de tipo z1, ya que el RE-Mote no se encuentra disponible en Cooja.

Tarea 1: nodo oculto

Crear una nueva simulación en Cooja seleccionando el Radio Medium “Unit Disk Graph Medium (UDGM): Distance Loss”.

Agregar 3 nodos de tipo z1:

- 1 nodo *receptor* con el código utilizado en la tarea 1 del laboratorio 2 (*multicast*), modificado para que no envíe mensajes.
- 2 nodos *emisores* con el código utilizado en la Tarea 1 del laboratorio 2 (*multicast*).

Verificar en la inicialización de los nodos que el protocolo de ciclo de trabajo de la radio sea ContikiMAC (protocolo por defecto de la plataforma).

Para poder observar el problema del nodo oculto, disponer los nodos de tal manera que los nodos *emisores* estén dentro del alcance del nodo *receptor* pero que no se escuchen entre ellos.

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 50 m.

Cambiar la disposición de los nodos para que todos se puedan escuchar entre sí. Observar que ahora se elimina el efecto del nodo oculto.

Guardar capturas de pantalla en situaciones donde se presenta el problema del nodo oculto **{#}**.

Tarea 2: análisis de mecanismo de acceso al medio

Se analizarán los protocolos nullRDC, ContikiMAC y TSCH (en sus versiones minimal y Orchestra).

Las distintas partes de esta tarea estarán basadas en una simulación inicial que se irá modificando para cambiar el mecanismo de acceso al medio, manteniendo la posición relativa de los nodos.

Crear una nueva simulación en Cooja y agregar 5 nodos de tipo z1: un nodo con el código **`lab-mac-receiver.c`** y los otros con el código **`lab-mac-sender.c`** .

2.1 nullRDC

Cambiar el protocolo de ciclo de trabajo de la radio a nullRDC (buscar la variable `NETSTACK_CONF_RDC` y definirla correctamente en el archivo `project-conf.h`).

Nota: Tener en cuenta que esta definición debe quedar por fuera de las guardas que definen los parámetros de TSCH, o será ignorada por el compilador.

Localizar en el timeline y en la ventana de radio messages un mensaje enviado como *broadcast* y otro que sea *unicast*. Guardar una captura de pantalla de cada uno **{#}**. Observar las direcciones destino utilizadas en cada caso, y cuándo se enciende la radio.

2.2 ContikiMAC

Cambiar el protocolo de ciclo de trabajo de la radio de nullRDC a ContikiMAC.

Nota: recordar que es importante ejecutar `make clean` antes de hacer una compilación para que tenga en cuenta la nueva configuración (macros).

Para cada mensaje que se pide identificar, el entregable asociado será una captura de pantalla del mismo.

1. Realizar una captura de pantalla con el nuevo comportamiento de la radio, observando cuándo se enciende **{#}**. Comparar con lo observado en la Tarea 2.1, cuando se utilizaba nullRDC.
2. Identificar nuevamente mensajes de *broadcast* y *unicast* comparando con lo observado en la Tarea 2.1 **{#}**.
3. Medir en el timeline el período de tiempo entre chequeos del canal de radio.
4. Identificar en qué casos y de qué forma se envían los reconocimientos de los mensajes (ACK).
5. Identificar con qué variable y en qué archivo de Contiki OS se especifica la tasa de chequeo del canal de radio, verificando si la misma es consistente con la medida de tiempo realizada en el punto 3.

6. Modificar en el archivo de configuración del proyecto la tasa de chequeo de canal a un valor diferente del especificado por defecto en Contiki. Medir nuevamente este período de tiempo en el timeline.

2.3 TSCH

2.3.1 TSCH minimal

1. Modificar los macros para utilizar ahora TSCH *minimal*.
2. En *Tools* > *Radio messages...* seleccionar el análisis (*Analyzer*) con la opción de PCAP para posteriormente analizarlos con Wireshark.
3. En el timeline seleccionar *Events* > *Radio channel* para ver el cambio de canal en el tiempo.
4. Simular la red identificando en Cooja o posteriormente en Wireshark:
    a. Envío de EB por parte del nodo *receiver* (nodo 1) y el valor del parámetro JOIN METRIC.
    b. Escaneo de canales de los nodos hasta poder recibir un EB.
    c. Envío de EB de los nodos que se unieron a la red TSCH y la información de los IE (*Information Elements*). Para esta parte en particular se recomienda usar Wireshark.
    d. Tiempo de *slot* y largo del *slotframe*. Estos valores deben ser medidos en el timeline de Cooja.

    Guardar capturas de pantalla donde se aprecien y se señalen las observaciones anteriores **{#}**.
5. Modificar el `project-conf.h` para que los saltos de canal sean entre 2 canales a su elección. Volver a correr la simulación, verificando que el cambio surtió efecto. Guardar una captura de pantalla **{#}**.

2.3.2 TSCH Orchestra

1. Modificar los macros para utilizar TSCH *Orchestra*.
2. Simular la red observando en Cooja o posteriormente en Wireshark las diferencias de comportamiento respecto a TSCH minimal, en particular la existencia de varios *slotframes* simultáneamente. Identificar los períodos de los *slotframes* observando el timeline de Cooja.
   Guardar una captura de pantalla **{#}**.

6 Entregables

Se deberá entregar un archivo comprimido que contenga lo siguiente:

1. carpeta Tarea 1 incluyendo `Makefile`, `project-conf.h`, `.c`, `.h`, simulaciones `.csc` y capturas de pantalla requeridas (#),
2. carpeta Tarea 2_1 incluyendo `Makefile`, `project-conf.h`, `.c`, `.h`, simulaciones `.csc` y capturas de pantalla requeridas (#),
3. carpeta Tarea 2_2 incluyendo `Makefile`, `project-conf.h`, `.c`, `.h`, simulaciones `.csc` y capturas de pantalla requeridas (#),
4. carpeta Tarea 2_3_1 incluyendo `Makefile`, `project-conf.h`, `.c`, `.h`, simulaciones `.csc` y capturas de pantalla requeridas (#),
5. carpeta Tarea 2_3_2 incluyendo `Makefile`, `project-conf.h`, `.c`, `.h`, simulaciones `.csc` y capturas de pantalla requeridas (#).

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**.

7 Referencias

[1] https://github.com/contiki-os/contiki/tree/master/core/net/mac/tsch.
[2] IETF RFC8180: Minimal IPv6 over the TSCH Mode of IEEE 802.15.4e (6TiSCH) Configuration, May 2017.
[3] S. Duquennoy, A. Elsts, B. Al Nahas and G. Oikonomou, "TSCH and 6TiSCH for Contiki: Challenges, Design and Evaluation," in DCOSS 2017, pp. 11-18, doi: 10.1109/DCOSS.2017.29.
[4] S. Duquennoy, B. Al Nahas, O. Landsiedel and T. Watteyne, "Orchestra: Robust Mesh Networks Through Autonomously Scheduled TSCH," in ACM SenSys 2015, pp. 337-350, doi: 10.1145/2809695.2809714.