

Laboratorio 5: MAC

1 Introducción

En este laboratorio se simulará una red de sensores inalámbricos utilizando la herramienta Cooja para identificar los diferentes tipos de tramas intercambiadas por los nodos con sus diferentes campos y estudiar los diferentes protocolos de acceso al medio (subcapa MAC) vistos en el curso. Se observará el comportamiento de una red utilizando utilizando CSMA (opción por defecto) y TSCH (*Time Slotted Channel Hopping*) con distintas configuraciones (minimal y Orchestra).

2 Objetivo

2.1 Objetivos generales

- Identificar los diferentes tipos de tramas.
- Comprender los mecanismos de acceso al medio y describir el funcionamiento de los protocolos MAC.

2.2 Objetivos específicos

- Identificar los diferentes tipos de tramas, modos de direccionamiento y los diferentes campos utilizados en la norma IEEE 802.15.4.
- Identificar las ventajas y desventajas de CSMA y TSCH, enfocado en el impacto en el *duty cycle* de la radio.
- 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 *Enhanced Beacons* (EB) y el proceso de *join* (unión).
- Observar las diferencias entre TSCH minimal y Orchestra.

3 Fundamentos

3.1 Formatos de trama de IEEE 802.15.4

Estudiar los formatos de trama de la norma [IEEE 802.15.4-2020](#)¹ centrado la atención en los diferentes tipos de trama: datos, ACK y EB (*enhanced beacons*) descriptos en el Capítulo 7.

¹Disponible a través del programa GET 802(R) Standards de la IEEE (requiere registro), o via timbo.org.uy.

3.2 Fundamentos de TSCH

Leer la documentación de Contiki-NG relacionada con TSCH. En particular:

- [Programming Contiki-NG TSCH and 6TiSCH](#) para familiarzarse con su implementación. En la misma hace referencia a los siguientes documentos.
- [Tutorials TSCH and 6TiSCH](#) es una guía de uso de TSCH y 6TiSCH partiendo de la aplicación `hello-world`. Seguiremos parte de esta guía en las actividades.
- [Programming Contiki-NG TSCH example applications](#) lista aplicaciones de ejemplo de TSCH y 6TiSCH que se pueden encontrar en `examples/6tisch/` que servirán como referencia.
- [Tutorials Switching to TSCH](#) lista los pasos necesarios para pasar de TSCH aplicaciones que usen CSMA (MAC por defecto).
- [Programming Contiki-NG Orchestra](#) para su descripción general y pasos para su utilización.

4 Actividades

Para las simulaciones utilizaremos nodos de tipo Cooja. No se utilizará un nodo *border-router* sino que se configurará uno de los nodos como *root* utilizando el comando shell: `rpl-set-root 1`.

Se pide guardar capturas de pantalla donde se señale la información marcada con `{#}`, las que formarán parte del entregable.

4.1 Análisis de formato e intercambio de tramas

Para esta parte se continuará utilizando el protocolo de acceso al medio por defecto que es CSMA.

1. Crear una nueva simulación en Cooja y agregar un nodo *server* y un nodo *client* con el código de la Tarea 3 del Laboratorio 2 (envío de mensajes UDP unicast).
2. Modificar el `Makefile` para incluir el módulo de [Shell](#), y modificar el nivel de [Logging](#) de la subcapa MAC a `LOG_LEVEL_DBG` en el archivo `project-conf.h`.
3. Correr la simulación.
4. Configurar el nodo *server* como *root* utilizando el comando shell `rpl-set-root 1`.
5. En la ventana de Radio messages, seleccionar 6LoWPAN Analyzer with PCAP para posteriormente analizarlos con Wireshark.
6. Esperar a que se envíen varios paquetes unicast.
7. Abrir con Wireshark el archivo PCAP generado por Cooja durante la simulación. Se recuerda que los archivos PCAP son generados automáticamente por Cooja y guardados en la carpeta `tools/cooja/`.

Para analizar las tramas se pide:

1. Identificar un mensaje enviado como *broadcast* `{#}` y otro que sea *unicast* `{#}`, observando las direcciones destino utilizadas en cada caso.
2. Para los mensajes que solicitan reconocimiento (ACK), identificar en qué campos del encabezado es solicitado `{#}`.
3. Analizar si las direcciones de capa MAC son largas o cortas y ubicarlas `{#}` en cada trama.
4. Identificar el PANID asignado por Contiki `{#}`.

4.2 Análisis de mecanismo de acceso al medio

Se analizarán los protocolos de acceso al medio CSMA (opción por defecto) 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.

Partir de la simulación de la parte anterior y agregar tres nodos más, totalizando una red de 5 nodos.

4.2.1 CSMA

Determinar en la ventana de Timeline el estado de la radio (Event / Radio on/off) e identificar en que intervalos se está transmitiendo o recibiendo (Event / Radio traffic). Tener presente la descripción de CSMA realizada en [Tutorials Switching to TSCH](#).

Las observaciones anteriores se tomarán como referencia para reflexionar posteriormente sobre las ventajas y desventajas del uso de TSCH.”

4.2.2 TSCH (configuración minimal)

1. Modificar el archivo `Makefile` para utilizar TSCH como MAC, agregando la definición `MAKE_MAC = MAKE_MAC_TSCH2`.
2. Correr la simulación.
3. Configurar el nodo `server` como root utilizando el comando shell `rpl-set-root 1`.
4. En la ventana de Radio messages, seleccionar 6LoWPAN Analyzer with PCAP para posteriormente analizarlos con Wireshark.
5. En el timeline seleccionar Events / Radio channel para ver el cambio de canal en el tiempo.
6. Identificar en Cooja o Wireshark:
 - a. Envío de EB por parte del nodo `receiver` 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 utilizar Wireshark. `{#}`.
 - d. Tiempo de `slot` y largo del `slotframe`. Estos valores deben ser medidos en el timeline de Cooja. `{#}`.
7. Modificar el archivo `project-conf.h` para que los saltos de canal sean entre 2 canales a elección. Volver a correr la simulación, verificando que el cambio surtió efecto `{#}`.

4.3 TSCH Orchestra

1. Seguir los pasos de [Programming Contiki-NG » Orchestra](#) para agregar el módulo de *Orchestra* al `Makefile`.
2. Correr la simulación.
3. Configurar el nodo `server` como root utilizando el comando shell `rpl-set-root 1`.

²Recordar que es importante ejecutar `make distclean` antes de hacer una compilación que omite 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. Adicionalmente se recomienda usar `make viewconf` especificando el `target` correspondiente.

4. Identificar en Cooja o Wireshark:
 - a. las diferencias de comportamiento respecto a TSCH minimal, en particular la existencia de varios *slotframes* simultáneamente. `{#}`.
 - b. Identificar los períodos de los *slotframes* observando el timeline de Cooja `{#}`.
5. Explorar el uso de los comandos *shell* asociados a TSCH para verificar los datos obtenidos en el punto anterior `{#}`.

5 Entregables

Se deberá entregar un archivo comprimido que contenga lo siguiente:

1. Carpeta con el código de los nodos clientes incluyendo `Makefile`, `project-conf.h`, `.c`, `.h` y simulaciones `.csc`.
2. Carpeta con las capturas de pantalla indicadas en las diferentes actividades identificadas por su número (por ejemplo: 4.1.1 para la captura del punto 1 de la sección 4.1)

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

- “IEEE Standard for Low-Rate Wireless Networks,” in IEEE Std 802.15.4-2020 (Revision of IEEE Std 802.15.4-2015) , vol., no., pp.1-800, 23 July 2020, doi: [10.1109/IEEESTD.2020.9144691](https://doi.org/10.1109/IEEESTD.2020.9144691).
- Documentación de Contiki-NG
 - Programming Contiki-NG [TSCH and 6TiSCH](#)
 - Tutorials [TSCH and 6TiSCH](#)
 - Programming Contiki-NG [TSCH example applications](#)
 - Tutorials [Switching to TSCH](#)
 - Programming Contiki-NG [Orchestra](#)
- Publicaciones científicas:
 - G. Oikonomou, S. Duquennoy, A. Elsts, J. Eriksson, Y. Tanaka, and N. Tsiftes, “The Contiki-NG open source operating system for next generation IoT devices,” 2022 SoftwareX, 18, p.101089. <https://doi-org.proxy.timbo.org.uy/10.1016/j.softx.2022.101089>
 - S. Duquennoy, A. Elsts, B. A. Nahas and G. Oikonomo, “TSCH and 6TiSCH for Contiki: Challenges, Design and Evaluation,” 2017 13th International Conference on Distributed Computing in Sensor Systems (DCOSS), Ottawa, ON, Canada, 2017, pp. 11-18, doi: [10.1109/DCOSS.2017.29](https://doi.org/10.1109/DCOSS.2017.29).
 - S. Duquennoy, B. Al Nahas, O. Landsiedel, and T. Watteyne. 2015. Orchestra: Robust Mesh Networks Through Autonomously Scheduled TSCH. In Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems (SenSys '15). Association for Computing Machinery, New York, NY, USA, 337–350. doi: [10.1145/2809695.2809714](https://doi.org/10.1145/2809695.2809714).
- IETF RFC
 - P. Thubert, “An architecture for IPv6 over the time-slotted channel hopping mode of IEEE 802.15. 4 (6TiSCH).” In RFC 9030. 2021. <https://datatracker.ietf.org/doc/rfc9030/>
 - X. Vilajosana, K. Pister, and T. Watteyne, “Minimal ipv6 over the tsch mode of ieee 802.15. 4e (6tisch) configuration.” 2017. (No. rfc8180). <https://datatracker.ietf.org/doc/rfc8180/>

Los materiales de este curso fueron parcialmente financiados por:



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