

# Laboratorio 6: Capas de Enlace y de Adaptación a IPv6 y Análisis de Consumo

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](#)

### [4 Fundamentos sobre herramientas de estimación de consumo](#)

#### [4.1 Energest](#)

#### [4.2 PowerTracker](#)

### [5 Procedimientos y Tareas](#)

#### [Tarea 1: análisis de formato e intercambio de tramas](#)

#### [Tarea 2: 6LoWPAN](#)

#### [Tarea 3: estimación del consumo](#)

### [6 Entregables](#)

### [7 Referencias](#)

## 1 Introducción

En este laboratorio se simulará una red de sensores inalámbricos utilizando la herramienta Cooja (disponible en el repositorio de Contiki). Se analizarán las tramas de enlace de datos intercambiadas por los nodos para comprender el funcionamiento y las limitaciones de la capa

de enlace de datos. También se analizará el funcionamiento de 6LoWPAN observando la fragmentación de paquetes y compresión de encabezados. Finalmente se hará un análisis del consumo energético de los nodos. Se utilizará la herramienta Wireshark para complementar el análisis anterior.

## **2 Objetivo**

El objetivo de este laboratorio es simular una red de sensores inalámbricos utilizando el simulador Cooja para identificar los diferentes tipos de tramas intercambiadas por los nodos y sus diferentes campos. También se buscará comprender el funcionamiento de 6LoWPAN, analizando particularmente la fragmentación de paquetes y compresión de encabezados. Finalmente se buscará que el estudiante comprenda las nociones básicas del consumo energético de un nodo y cómo estimarlo utilizando las distintas herramientas disponibles.

### **2.1 Objetivos de aprendizaje**

Se espera que el estudiante al final del laboratorio pueda:

- Identificar los diferentes tipos de tramas, modos de direccionamiento y los diferentes campos utilizados en la norma IEEE 802.15.4.
- Comprender el funcionamiento de 6LoWPAN, analizando la fragmentación de paquetes y compresión de encabezados.
- Hacer una estimación del consumo energético de los nodos.
- Utilizar Wireshark como complemento de análisis de tráfico de una red.

## **3 Lecturas y actividades previas**

1. Previo al laboratorio el estudiante deberá familiarizarse con los formatos de trama de la norma IEEE 802.15.4. Para eso se recomienda:
  - Consultar la norma misma accesible en [\[5\]](#), sección 5.2: MAC frame formats.

## **4 Fundamentos sobre herramientas de estimación de consumo**

### **4.1 Energest**

El nodo Z1 está compuesto por el microcontrolador ( $\mu$ C), la radio y circuitos auxiliares. El consumo energético está dominado por el consumo del microcontrolador y la radio.

Los modos de operación del microcontrolador se pueden reducir a: “ $\mu$ C Activo” y “ $\mu$ C en LPM” (Low Power Mode). La radio, en cambio, puede estar en alguno de los siguientes estados: “Radio Transmitiendo”, “Radio Escuchando” y “Radio en LPM”.

El estado del nodo surge de la composición (producto cartesiano) de los estados del microcontrolador y la radio. Sin embargo, no todos los estados son válidos: si la radio está transmitiendo o escuchando, el microcontrolador estará necesariamente activo.

Entonces los estados válidos de un nodo son: **“Nodo Inactivo”** (“μC en LPM” y “Radio en LPM”), **“Nodo Activo”** (“μC Activo” y “Radio en LPM”), **“Nodo Escuchando”** (“μC Activo” y “Radio Escuchando”) y **“Nodo Transmitiendo”** (“μC Activo” y “Radio Transmitiendo”).

A su vez, dentro del estado **“Nodo Inactivo”**, se diferencia entre los casos en que el microcontrolador está ejecutando rutinas de atención a las interrupciones (**“Nodo Inactivo IRQ”**) o está en modo de bajo consumo (**“Nodo Inactivo LPM”**).

El consumo del nodo en cada uno de estos estados es aproximadamente constante. El módulo Energest de Contiki estima el tiempo en cada uno de los estados y permite calcular el consumo total del nodo.

Energest mide, en tiempo real, el tiempo acumulado que el nodo permanece en cada “tipo” de Energest. Para consultar dichos tiempos, se utiliza la siguiente función:

```
energest_type_time(TIPO_A_CONSULTAR);
```

Los tipos de Energest se muestran en la siguiente tabla, mostrando la equivalencia con los estados del nodo explicados previamente.

Tipo de Energest	Estado del Nodo
CPU	“Nodo Activo” + “Nodo Escuchando” + “Nodo Transmitiendo”
LPM	“Nodo Inactivo LPM” + “Nodo Inactivo IRQ”
LISTEN	“Nodo Escuchando”
TRANSMIT	“Nodo Transmitiendo”
IRQ	“Nodo Inactivo IRQ”

Notar que a partir de los tiempos que nos da Energest, podemos calcular el tiempo de cada estado del nodo. De esta tabla se desprende que el tiempo total es la suma de los estados CPU y LPM.

La función `energest_type_time()` devuelve un tiempo cuya unidad son ticks del `RTIMER`. Para pasarlo a segundos se divide entre el valor del macro `RTIMER_SECOND`, el cual se debe averiguar para el nodo Z1.

Luego multiplicando el tiempo por la corriente de cada estado se puede estimar el consumo de carga del estado correspondiente.

En función de estos tiempos, se puede estimar la carga promedio consumida con la siguiente ecuación:

$$Q_{pilas} = \sum_{estados} t_{estado} \cdot I_{estado} = t_{total} \cdot \sum_{estados} DC_{estado} \cdot I_{estado}$$

siendo  $DC_{estado}$  el ciclo de trabajo de cada estado del nodo ( $t_{estado} = t_{total} \cdot DC_{estado}$ ).

Asumiendo que el nodo sigue operativo hasta que las baterías agoten su carga completamente, se puede calcular el tiempo de duración de las mismas.

Para el cálculo de la carga consumida se utilizarán los siguientes consumos de corriente nominales:

Estado	Corriente
Nodo Escuchando	21.8 mA
Nodo Transmitiendo	19.5 mA
Nodo Activo	6.25 mA
Nodo Inactivo IRQ	6.25 mA
Nodo Inactivo LPM	54 µA

## 4.2 PowerTracker

Por otro lado, la herramienta *PowerTracker* disponible en la interfaz gráfica de Cooja, permite ver el tiempo que pasa un nodo en cada uno de los siguientes estados: *Radio on*, *Radio TX* y *Radio RX*.

Cuando la radio está transmitiendo (*Radio TX*), a su vez debe estar encendida (*Radio on*). Lo mismo sucede con el modo de recepción (*Radio RX*). Sin embargo, la radio puede estar encendida sin transmitir ni recibir (escuchando).

Se puede acceder al *PowerTracker* en el menú de Cooja: Tools → Mote Radio Duty Cycle...

## 5 Procedimientos y Tareas

### Tarea 1: análisis de formato e intercambio de tramas

Crear una nueva simulación en Cooja y agregar 2 nodos de tipo z1 que utilicen ContikiMAC: un nodo con el código `lab-mac-sender.c` y otro con el código `lab-mac-receiver.c`.

Desplegar la herramienta “Radio Messages” de Cooja y configurarlo para que en la sección Analyzer tenga “6LoWPAN Analyzer with PCAP”.

Correr la simulación hasta que se reciba un paquete y su reconocimiento (ACK). 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/build`.

Para analizar las tramas se pide:

1. Para los mensajes que solicitan reconocimiento (ACK), identificar en qué campos del encabezado es solicitado. Observar qué otra información se envía en los ACK (por *piggybacking*) **{#}**.
2. Analizar si las direcciones de capa MAC son largas o cortas y ubicarlas **{#}** en cada trama.
3. Identificar el PANID asignado por Contiki **{#}**.

Guardar capturas de pantalla donde se señale la información marcada con '**{#}**'.

### Tarea 2: 6LoWPAN

Agregar otro nodo de tipo *sender* a la simulación de la Tarea 1 de forma que los paquetes que envía tengan que hacer dos saltos para llegar al nodo de tipo *receiver*. Modificar el código de la Tarea 1 para enviar mensajes largos que deban ser fragmentados. Si el mensaje es suficientemente largo como para que no entre en el buffer de simple-udp, no se enviará. Por este motivo, se recomienda incrementar el tamaño del buffer de UDP definiendo en el archivo `project-conf.h`:

```
#define UIP_CONF_BUFFER_SIZE 250
```

Se pide:

1. Verificar que efectivamente el mensaje es fragmentado.
2. Ver que todos los fragmentos llegan al nodo *receiver*. Guardar una captura de pantalla mostrando dos fragmentos del mismo mensaje recibidos en el nodo *receiver* **{#}**.

3. Identificar casos en que 6LoWPAN comprime encabezados. Guardar una captura de pantalla mostrando un encabezado comprimido e indicar qué tipo de compresión usa **{#}**.

Guardar capturas de pantalla donde se señale la información marcada con '**{#}**'.

### Tarea 3: estimación del consumo

Agregar 9 nodos de tipo *sender* a la simulación de la Tarea 1 formando una topología a elección. Modificar el código de la Tarea 1 para que se impriman los tiempos que cada nodo pasa en cada estado, haciendo uso de la siguiente función:

```
printf("LPM: %ld, CPU: %ld, LISTEN: %ld, TRANSMIT: %ld, IRQ: %ld\n",
      energest_type_time(ENERGEST_TYPE_LPM),
      energest_type_time(ENERGEST_TYPE_CPU),
      energest_type_time(ENERGEST_TYPE_LISTEN),
      energest_type_time(ENERGEST_TYPE_TRANSMIT),
      energest_type_time(ENERGEST_TYPE_IRQ));
```

Elegir un nodo de la red, y analizar su consumo energético.

Se pide:

1. Abrir la herramienta *PowerTracker*, arrancar la simulación y esperar a que transcurran 30 minutos simulados para que se forme la red RPL y se estabilice.
2. Elegir un nodo y observar qué porcentaje del tiempo total está en cada estado del nodo (*duty cycle*). Para ello utilizar la herramienta *Energgest*, analizando la salida en consola de dicho nodo. Corroborar que el tiempo total (la suma del tiempo de todos los estados del nodo) es de 30 minutos.
3. Comparar los resultados del punto anterior (*Energgest*) con los valores que muestra la herramienta *PowerTracker*.
4. Usar los resultados del punto 2 para calcular la duración de las baterías de ese nodo, suponiendo que se alimenta con dos pilas AA en serie de 2300 mAh cada una.

## 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 con las capturas de pantalla requeridas (#),
3. carpeta Tarea 3 con los datos relevados en los puntos 2 a 4, incluyendo la comparación entre los datos obtenidos con las herramientas *Energgest* y *PowerTracker*, y el detalle del cálculo para estimar la duración de las pilas.

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

## 7 Referencias

- [1] <https://github.com/marcozennaro/IPv6-WSN-book/blob/master/Releases>
- [2] [http://processors.wiki.ti.com/index.php/Category:Code\\_Composer\\_Studio\\_v7](http://processors.wiki.ti.com/index.php/Category:Code_Composer_Studio_v7)
- [3] <https://github.com/contiki-os/contiki/wiki/An-Introduction-to-Cooja>
- [4] <http://www.contiki-os.org/start.html>
- [5] <http://standards.ieee.org/about/get/>