Laboratorio 6: 6LoWPAN y medidas de consumo

1 Introducción

En este laboratorio se simulará una red de sensores inalámbricos utilizando la herramienta Cooja (disponible en el repositorio de Contiki). 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

2.1 Objetivos generales

- Comprender el funcionamiento de 6LoWPAN
- Adquirir nociones básicas de fuentes de consumo energético de un nodo.

2.2 Objetivos específicos

- Comprender el funcionamiento de 6LoWPAN, analizando la compresión de encabezado y la fragmentación de paquetes.
- Identificar la fuentes de consumo energético de un nodo.
- Realizar una estimación del consumo energético de los nodos utilizando *Energest*.
- Utilizar Wireshark como complemento de análisis de tráfico de una red.

3 Fundamentos

3.1 6LoWPAN

Leer los documentos:

- RFC-4944 Transmission of IPv6 Packets over IEEE 802.15.4 Networks.
  - En particular estudiar las secciones 1 a 5.
- RFC-6282 Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks.

3.2 Energest

Leer la documentación de Contiki-NG relacionada al módulo *Energest*, en particular:

- Instrumenting Contiki NG applications with energy usage estimation explica cómo utilizar el módulo *Energest* para hacer una estimación del consumo energético.

3.3 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…

4 Actividades

Para las simulaciones utilizaremos nodos de tipo *Z1*[1]. Es muy probable que en algún punto del laboratorio se enfrenten a problemas de restricciones de memoria de la plataforma *Z1*. Para sortearlas, se sugiere seguir las recomendaciones disponibles en el tutorial RAM and ROM usage, en particular en la sección Saving RAM and ROM. También recordar que aumentar o disminuir los niveles de logging, así como la incorporación de módulos, tienen un alto impacto en el uso de memoria.

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

4.1 6LoWPAN

1. Crear una nueva simulación en Cooja y agregar un nodo *server* y dos nodos *client* con el código de la Tarea 3 del Laboratorio 2 (envío de mensajes UDP unicast). Armar la topología de forma que los paquetes que envía el tercer nodo tengan que hacer dos saltos para llegar al nodo *server*.
2. Modificar el código de los nodos *server* para que se auto-configuren como *root* de la red RPL. Para ello incluir el encabezado #include `"net/routing/routing.h"` y llamar a la función `NETSTACK_ROUTING.root_start();` en el proceso principal.
3. Modificar el código de los nodos *client* para que envien 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`
4. Modificar el nivel de Logging de 6LoWPAN a `LOG_LEVEL_DBG` en el archivo `project-conf.h`.
5. Correr la simulación.
6. En la ventana de Radio messages, seleccionar 6LoWPAN Analyzer with PCAP para posteriormente analizarlos con Wireshark.
7. Esperar a que se envíen varios paquetes unicast.

Se pide:

1. Verificar que efectivamente el mensaje es fragmentado.
2. Ver que todos los fragmentos llegan al nodo *server* **{#}**.
3. Identificar casos en que 6LoWPAN comprime encabezados. Guardar una captura de pantalla mostrando un encabezado comprimido e indicar qué tipo de compresión usa **{#}**.

4.2 Estimación del consumo

4.2.1 Pruebas en Simulación

1. Agregar 9 nodos de tipo *client* a la simulación de la tarea anterior formando una topología a elección.
2. Modificar el `Makefile` para incluir el módulo de Energest, y asegurarse que se esté utilizando TSCH en la subcapa MAC, agregando la definición `MAKE_MAC = MAKE_MAC_TSCH`.
3. Correr la simulación.
4. Elegir un nodo de la red, y analizar su consumo energético.

[1] Notar que las direcciones unicast son distintas que las utilizadas por los nodos de tipo *cooja*.

Se pide:

1. Abrir la herramienta *PowerTracker*, correr 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 *Energest*, 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 (*Energest*) 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.

4.2.2 Pruebas en Hardware

1. Cargar un nodo de tipo *server* y otro de tipo *client* en hardware, y ajustar las direcciones de destino para que los mensajes se envíen correctamente.
2. Modificar el `Makefile` para incluir el módulo de Energest, y asegurarse que se esté utilizando TSCH en la subcapa MAC, agregando la definición `MAKE_MAC = MAKE_MAC_TSCH`.
3. Dejar la aplicación corriendo por aproximadamente 10 minutos.
4. Analizar el consumo energético de ambos nodos.

Se pide:

1. Observar qué porcentaje del tiempo total está en cada estado del nodo (*duty cycle*). Para ello utilizar la herramienta *Energest*, analizando la salida en consola de cada nodo.
2. Usar los resultados del punto anterior para calcular la duración de las baterías de los nodos, suponiendo que se alimenta con dos pilas AA en serie de 2300 mAh cada una.

5 Entregables

Se deberá entregar un archivo comprimido que contenga lo siguiente:

1. Carpeta Tarea 4.1 con las capturas de pantalla requeridas **{#}**.
2. Carpeta Tarea 4.2.1 con los datos relevados en los puntos 2 a 4, incluyendo la comparación entre los datos obtenidos con las herramientas *Energest* y *PowerTracker*, y el detalle del cálculo para estimar la duración de las pilas.
3. Carpeta Tarea 4.2.2 con los datos relevados en el punto 1, y el detalle del cálculo para estimar la duración de las pilas del punto 2.

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

6 Referencias

- Documentacón de Contiki-NG
  - Tutorials Instrumenting Contiki NG applications with energy usage estimation
  - Tutorials Energy monitoring
  - Programming Contiki-NG Energest
- IETF RFC
  - RFC-4944 Transmission of IPv6 Packets over IEEE 802.15.4 Networks.
  - RFC-6282 Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks.

Los materiales de este curso fueron parcialmente financiados por: