Peer Assignment 2

Energy consumption estimation

Power consumption is always one of the most important issues when designing or using nodes in wireless sensor networks. Often, in order to save power consumption, sensor nodes turn radio receivers off and these radio receivers are merely waken up to receive incoming packets from senders (i.e. neighbour nodes). The concept behind this method is that when sensor nodes are in idle listening, energy consumption is minimized. However, in some applications, this method might be ineffective or cannot be used. Therefore, in this assignment, we experience different radio duty cycling (RDC) mechanisms and Medium Access Control (MAC) in Contiki. Via this, students learn how apply the specific mechanism for particular requirements of wireless sensor networks applications. In addition, students learn how to measure energy consumption of sensor nodes in Cooja.

In Contiki, MAC and RDC are important in the communication stack because they directly impact on total energy consumption of sensor nodes.

Contiki has several RDC drivers such ContikiMAC, X-MAC, LPP, and NullRDC. RDC drivers switch the radio off as much as possible for saving power consumption. In addition, it checks the radio medium for radio activity. The default check rate is 8 Hz

- Contiki MAC: is specific for 802.15.4 radio
- X-MAC is a short-preamble protocol
- LPP: receiver-initiated RDC protocol
- NullRDC: that never switches the radio off (used for testing)

Contiki has types of MAC: nullmac and csma.

- nullmac: is a simple pass-through protocol
- Csma: implements addressing, sequence number and retransmissions

Configuring the RDC Driver

In order to setup configuration for RDC drivers, students must include the configuration file named “project-conf.h” in the project. Note that each project has a particular configuration file. In addition, to enable the RDC configuration, the following line must be added in the project’s Makefile:

```
CFLAGS += -DPROJECT_CONF_H=\"project-conf.h\"
```

The full Makefile may now look something like this:

```
CONTIKI = /home/user/contiki
CFLAGS += -DPROJECT_CONF_H=\"project-conf.h\"
include $(CONTIKI)/Makefile.include
```

The configuration file is used for override one or several default configurations of Contiki. In this assignment we merely override the RDC layer driver.

Firstly, we modify RDC channel check rate (Hz) by adding the line:

```
#define NETSTACK_CONF_RDC_CHANNEL_CHECK_RATE 16
```

Next, to specify a RDC driver, we add the line to the configuration file

#define NETSTACK_CONF_RDC nullrdc_driver

To specify what MAC driver, we add the line
#define NETSTACK_CONF_MAC nullmac_driver

After enabling the configuration file (project-conf.h) in the Makefile, it is recommended that existing dependencies should be clean up by the command:
make TARGET=sky clean

Please replace “sky” with the target node you want to deploy. This command is needed for the first time only. After that just compile by the command
make TARGET=sky

Note: In default, Contiki stack layers:

1. **Network layer** - rime_driver
2. **MAC layer** - nullmac_driver
3. **RDC** - nullrdc_driver
4. **Framer** - framer_nullmac
5. **Radio** - nullradio_driver

Energy Estimation

In Contiki, the Energest module is used for estimating energy consumption. In this assignment, we only focus on energy consumption of transmit, listen, CPU and LPM.

- Transmit and listen refer to radio transmitting and receiving,
- CPU refers to an active mode of microcontroller without radio transmitting or receiving.
- LPM stands for low power mode (i.e. sleep mode)

Energy consumption is calculated by multiply the voltage and the current altogether with a duration which the sensor node spends for operating (i..e transmitting, or receiving).
Note: the voltage and current details can be found from the data sheet of the node.

Another important note:. The Energest module does not return time in seconds but in ticks. Therefore, to get time in seconds, you must divide by a constant RTIMER_SECOND. This constant represents for a number of ticks per second. Depending on a specific platform, this value can be different. Therefore in order to reduce the complexity of assessing the peer assignment, we use the Clock library’s macros in Contiki.
Note: CLOCK_SECOND: a number of ticks per second.

The following equations can be used for calculating the energy consumption of a sensor node

$$E_{transmit} = t_{transmit}(ticks) \,/\, timer\ (ticks/second) * I_{transmit} * V$$

$E_{reciving} = t_{receiving}(ticks) / timer\ (ticks/second) * I_{receiving} * V$
$E_{CPU} = t_{CPU}(ticks) / timer\ (ticks/second) * I_{CPU} * V$
$E_{LPM} = t_{LPM}(ticks) / timer\ (ticks/second) * I_{LPM} * V$

Task 1
Create your own folder under **/home/user/contiki-2.7/examples/** directory. Copy **/home/user/contiki- 2.7/examples/powertrace** and save it in your folder. Configure the channel check rate to be 4, 8 and 16 Hz with ContikiMAC and CSMA configuration. Use the energest module and measure the energy dissipation for the three rates.

Task 2
Configure the RDC driver into Null_RDC, X-MAC and LPP. Measure the energy consumption for each configuration. Which RDC consumes the highest energy and which one is the lowest?

Task 3
Please edit the main source file in the folder **/home/user/contiki-2.7/examples/powertrace** with the purpose of using unicast transmission. Choose one node and compare its total energy consumption for the different RDC driver configurations (Null_RDC, LPP, XMAC, CX-MAC, ContikiMac)

Function powertrace_start (clock_time_t period) start a process “power_trace_process”. The “powertrace_process” includes a single thread which is invoked by time event. In this case, after the clock_time_t period, the thread is invoked. At this time, it will call the function “powertrace_print” to print out clock_time, sending address, receiving address, sequence number, etc.

The source code: example-powertrace.c is just an example which shows how power can be traced. In our exercises, we do not use two threads but only focus on a single thread. Hence, your task is to comment the powertrace_start function in the example-powertrace.c and edit the code inside infinitive loop while (1) to calculate energy and print it out. The final source code will be similar to a combination of code in tip1 and tip2.

Tip1:

cpu = energest_type_time(ENERGEST_TYPE_CPU);

lpm = energest_type_time(ENERGEST_TYPE_LPM);

transmit = energest_type_time(ENERGEST_TYPE_TRANSMIT);

listen = energest_type_time(ENERGEST_TYPE_LISTEN);

time = cpu + lpm

radio = transmit + listen.

Tip2:

```
#include ”sys/energest.h”
...
static unsigned long rx_start_duration;
..
// in a while loop
while(1) {
...
}
...
```

Tip3:

RDC driver configuration	**Value in Contiki**
ContikiMAC	contikimac_driver
Null_RDC	nullrdc_driver
LPP	lpp_driver
X-MAC	xmac_driver
CX-MAC	cxmac_driver