Tutorial 2: Network Performance in Cooja

AESE Group

January 22, 2018

1. Introduction

Sensor networks typically consist of potentially thousands of nodes which are characterized as low-cost, low-power, and resource constrained devices. Despite their limitations, these must be able to provide a reliable service. Different optimization methods are used to improve the performance of sensor networks and to extend their lifetime. Also, these must be monitored and their performance effectively measured, to guarantee that the network fulfils initial expectations.

Multiple factors determine the performance of sensor networks. To ensure the network reliability, data should be delivered to a destination without errors and in a defined time. How much packet loss or time delay can be tolerated are both application dependent metrics. For example, the sensor network used to monitor a patient's health is delay and loss intolerant. Also, it is important to have motes that are efficient when using the available energy. Measuring their power usage is crucial as they are mostly using tiny batteries which could be very difficult or impossible to replace. Additionally, there are many different metrics, such as network lifetime, network overhead, throughput, response time, sampling frequency, that can be considered when evaluating the performance of the network.

This tutorial will give you an insight concerning the performance of the network. We will use packet delivery ratio, power usage and dissemination time to measure the performance of different networks topologies. These metrics in the context of the Contiki OS are described next.

1.1 Packet Delivery Ratio (PDR)

The PDR is a ratio between the total number of packets successfully received at the server side and the number of packets sent by the motes. In the ideal case, when all packets are successfully received, PDR equals 1 (or 100%). If no packets have been received, PDR equals 0 (or 0%). This metric has been addressed in Tutorial 1.

1.2 Power Usage via Powertrace

To calculate the average power consumption for each mote in the network, Contiki Powertrace tool [1] can be used. The tool gives the information of the total time mote spends on transmitting (Tx), listening (Rx), low-power mode (LPM) and radio-off mode(CPU). Then, the mote's power consumption is given by

$$P(mW) = \frac{(T_x \times 19.5mA + R_x \times 21.8mA + CPU \times 1.8mA + LPM \times 0.0545mA) \times 3V}{32768 \times 10s}$$

It is assumed that the measurements are taken every 10 seconds. To include Powertrace in your applications, you need to add the following lines to your code.

- First, in your Makefile (e.g. Makefile in /home/user/contiki/examples/ipv6/rpl-udp folder) include the application:

```
APPS+=powertrace
```

- Then in your main .c file (e.g. in udp-client.c) include the following header:

```
#include "powertrace.h"
```

- And add this line to your main process in .c file (e.g. in udp-client.c):

```
powertrace_start(CLOCK_SECOND*10);
```

- Change the line "`#define DEBUG DEBUG_FULL`" .c file (e.g. in udp-client.c) to:

```
#define DEBUG DEBUG_NONE
```

The Powertrace data will be available on the mote output window in this form:

```
00:10.522  ID:11   1286 P 0.18 0 4601 324085 0 2325 0 2028 4601 324085 0 2325 0 2028 (radio 0.70% / 0.70% tx 0.00% / 0.00% listen 0.70% / 0.70%)
00:10.563  ID:2    1285 P 0.18 0 14277 314153 5572 2865 0 2343 14277 314153 5572 2865 0 2343 (radio 2.56% / 2.56% tx 1.69% / 1.69% listen 0.87% / 0.87%)
00:10.577  ID:6    1285 P 0.18 0 9353 319080 2588 2642 0 2258 9353 319080 2588 2642 0 2258 (radio 1.59% / 1.59% tx 0.78% / 0.78% listen 0.80% / 0.80%)
00:10.672  ID:4    1285 P 0.18 0 5287 323148 0 3225 0 2862 5287 323148 0 3225 0 2862 (radio 0.98% / 0.98% tx 0.00% / 0.00% listen 0.98% / 0.98%)
00:10.725  ID:7    1285 P 0.18 0 4609 323820 0 2308 0 2028 4609 323820 0 2308 0 2028 (radio 0.70% / 0.70% tx 0.00% / 0.00% listen 0.70% / 0.70%)
```

The values are described in Table 1 (starting with the entry after the ID number). Please note that the percentage values are omitted from the table.

Table 1. The structure of Powertrace messages

Parameter Name	Parameter Description
clock_time()	clock time
rimeaddr	rime address
seqno	sequence number
all_cpu	accumulated CPU energy consumption
all_lpm	accumulated Low Power Mode energy consumption
all_transmit	accumulated transmission energy consumption
all_listen	accumulated listen energy consumption
all_idle_transmit	accumulated idle transmission energy consumption
all_idle_listen	accumulated idle listen energy consumption
CPU	CPU energy consumption for this cycle
LPM	LPM energy consumption for this cycle
Tx	transmission energy consumption for this cycle
Rx	listen energy consumption for this cycle
Idle_transmit	idle transmission energy consumption for this cycle
Idle_listen	idle listen energy consumption for this cycle

To calculate the power usage, you should use the entries in columns 11-14.

1.3 Dissemination Time

Dissemination time represents the length of time needed for a packet to travel between the source (mote) and the destination (server). It is usually given as the average time of all packets for an individual mote and it is measured in ms.

2. Lab Task

1. Set up a WSN which consists of 10 motes by following these steps:
   - Radio medium: Unit Disk Graph Medium (UDMG): Distance Loss
   - Type of motes:
     - **One** Sky Mote running udp-server.c (from /home/user/contiki/examples/ipv6/rpl-udp folder) **randomly** positioned
     - **Nine** Sky Motes running udp-client.c (from /home/user/contiki/examples/ipv6/rpl-udp folder) **randomly** positioned
   - Simulation area: 150m x 150m
   - Transmission (TX) range: 50m
   - Interference (INT) range: 70m

   All other parameters should be kept as default values. Make sure that you do not have isolated motes.

2. Change the sending rate to 6 packets per minute.
3. Run the simulation for 20 minutes with no speed limits and analyse the *power usage* and the *dissemination time* per mote (discard first 5 minutes). The Powertrace measurements are taken every 10 seconds.
   PLEASE NOTE: To speed up the simulation, remove all unnecessary printfs in both files, udp-server.c and udp-client.c, and turn off Radio Traffic in the Network window.
4. Change the topology of the network to the LINEAR topology. In the linear topology motes are deployed in a line. All motes are equally spaced and connected with maximum two other motes. The *server mote* should be placed at the end of the line, while the simulation area should be increased to accommodate all ten motes.
5. Compare the performance of random and linear topology using the same metrics as in Task 3.

3. References

[1] A. Dunkels, J. Eriksson, N. Finne, and N. Tsiftes, Powertrace: Network Level Power Profiling for Low - Power Wireless Networks, [Online], Available: http://soda.swedish-ict.se/4112/, 2011.