Sensors Lab: A Preliminary Analysis

Heerok Banerjee*, Mathias Ooms†
Department of Mathematics and Computer Science,
University of Antwerp
Email: *heerok.banerjee@student.uantwerpen.be, †mathias.ooms@student.uantwerpen.be

***Abstract*—In this report, we have presented three preliminary analyses pertaining to low-powered Wireless Sensor Networks (WSNs). We have conducted experimental analyses which provides insights to the energy consumption, network joining time and range capacities of such WSNs. We have discussed TSCH and 6TiSCH enabled networks and how these parameters impact different networking objectives such as network lifetime, routing overhead, broadcast efficiency, neighbour discovery and end-to-end reliability.**

***Index Terms*—Wireless Sensor Networks, WSN, LP-WAN, TSCH, 6TiSCH**

I. Introduction

WSNs are spatially distributed networks of sensor devices with limited computing and sensing equipments. These sensor devices essentially constitute a wireless communication network with many primary use-cases such as monitoring, reporting, storage and analyses. However, due to its limited hardware and low power capacity WSNs should be carefully designed and fabricated with a central attention on conserving energy and prolonging the network lifetime. Such design requirements are consistently demanded across low-powered wireless sensor networks, where the network adopts to topological variations but fails to reduce congestion and routing overheads [1]. Traditional MAC layer protocols such as S-MAC, Berkley MAC (B-MAC) and its derivatives used unsynchronized duty cycling and scheduled channel pooling to minimize idle listening. While this methodology is viable across many spatially distributed networks but it is infeasible for dynamic networks such as underwater WSN and vehicular ad hoc networks [2].

Time-slotted Channel Hopping (TSCH) is a MAC-layer channel access technique that uses time synchronization to support synchronized transmission of data in a low-power networking environment. TSCH essentially governs the activity of each node by employing a time-slotted schedule with channel hopping. Channel hopping guarantees transmission reliability as the outgoing traffic is transmitted across different frequencies, hence diluting the effect of external interference, noise and multi-path fading [3]. Fig. 1 illustrates how channel hopping is implemented and since every node operate on different frequency bands, the effects of channel interference is negligible.

II. Files → in appendix

..\LSN This directory contains all the files necessary and additional that were used to execute the analyses and for clarification of theory.

– background
– problem statement
– your contributions
– structure of paper

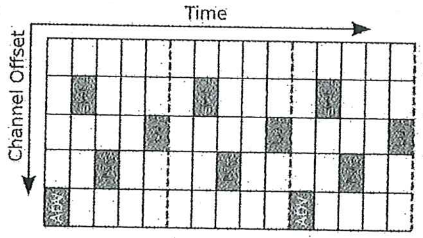


Fig. 1: Dedicated channel hopping with scheduling

..\LSN\literature This directory contains files providing theory and information used in this report.

..\LSN\assignment_1\question_* This directory contains all the files necessary to execute the analyses. A more detailed README is included here, explaining how to execute the analysis.[1]

Contiki-NG is an open-source framework for Next-Generation IoT gadgets. It centers around trustworthy low-control correspondence and standard conventions, for example, IPv6/6LoWPAN, 6TiSCH, RPL, and CoAP. **We have forked the original repository and our work is stored in this repository.**

Experiment setup.

III. ~~Preliminary Analyses~~

In this section, we will extensively discuss the three preliminary analyses that has been conducted. For each analysis, we discuss our agenda and the experimental setup in granular details. mention the number of iterations per experiment

A. Analysing the 6TiSCH energy consumption

In this analysis, we attempt to observe the energy utilization of the root node and the leaf node after network convergence. We conduct this experiment for both 6-TiSCH and only TSCH network for a duration of 10 minutes.

We use the energest module to monitor energy consumption and dump the logs in text format. Subsequently, we use python's regular expression and pandas library to perform data prepossessing, namely pattern matching followed by chunking the data and finally, computing the energy utilization as per the provided metrics [see table I, equ. 1].

The root node and the leaf node is hardcoded, in the sense that they perform the network operations such as advertisement, neighbor discovery and transmission/reception

[1] All the files should be executed under contiki-ng native environment.

post processing?

– which advertisements
– how does neighb

If I add those 4 values together, the value will be larger than the total value right?
→ explain this figure.

autonomously without any manual intervention. We can use the NG shell command line interface to manually check the status of the TSCH network using the command: tsch-status.

```
JOINED NETWORK
Timer started for 600 seconds
Time left: 600
Energest:
CPU 1s LPM  59s DEEP LPM 0s Total time 60s
Radio LISTEN 58s TRANSMIT 0s OFF  2s
Time left: 590
```

The above is the dump format of energest module. We have discussed its implementation and the obtained results in more details in section IV.

B. Analysing the TSCH joining process

In this analysis, our objective is to observe the time taken by the leaf node to join the TSCH network. Essentially, we are interested in minimizing the joining time by calibrating the device configuration.

The energest module is a bulky program which consumes a lot of primary memory. Since, Zolertia RE mote is an extremely memory-constrained device, we had to manually configure some of the device parameters. The configuration used for this entire analysis is given below:

```
#define QUEUEBUF_CONF_NUM                  4
#define NBR_TABLE_CONF_MAX_NEIGHBORS       4
#define NETSTACK_MAX_ROUTE_ENTRIES         4
#undef UIP_CONF_BUFFER_SIZE
#define UIP_CONF_BUFFER_SIZE             140
```

Essentially, we have reduced the number of neighbour enteries, buffer sizes and routing tables to preserve primary memory for bulky operations. As an anecdotal remark, it is impossible to flash energest with default configuration and so, finding the minimal device configuration is also an open problem but beyond the scope of this study.

→ what is the default value for the EB range?

C. Analysing range capabilities

In this analysis, we have attempted to analyse range capabilities of the TSCH MAC layer with respect to latency, throughput and energy consumption. Obtaining this data would be possible by assigning a fixed cell in the schedule for the leaf-node to send a packet each second. We would vary the range as follows: 1m, 5m, 10m, 50m, 100m on the included locations

Unfortunately we ran into problems while building the program. We tried to use a fixed cell in the TSCH schedule. However while we were able to create a "link" in the fixed cell, we could not make use of it. So we were not able to send packets and thus couldn't create the required data.

However we would like to share our speculations for this analysis. Increasing the range would imply an increase in latency, because the distance of the signals that need to travel are increased. The throughput would decrease, because the signals will be affected by noise. Energy consumption would increase to prevent the signals from fading. Of course these statements are just assumptions that we can't prove until we've acquired the data. → you could reference other work to back up your statements

we didn't receive any questions about this.

how? what do you mean?

IV. Results

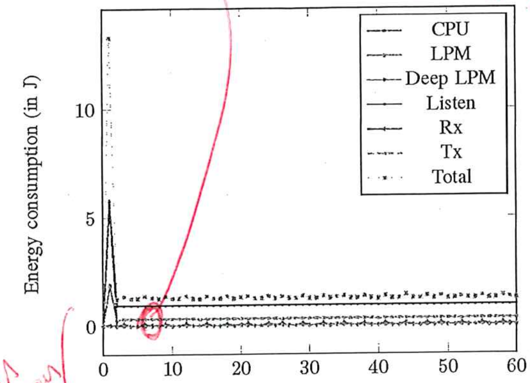
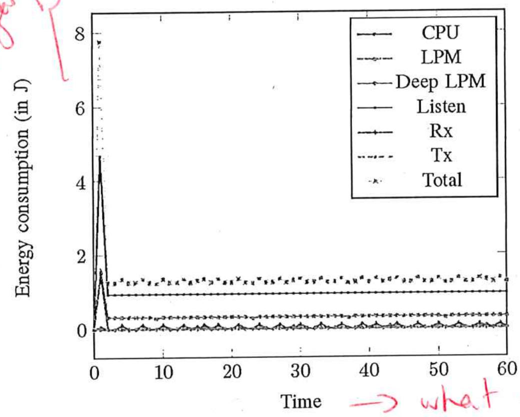


Fig. 2: Energy consumption of leaf node

is this instantaneous consumption or aggregated?

→ x-axis label? Time?



→ what time - s, minutes, hours?

Fig. 3: Energy consumption of root node

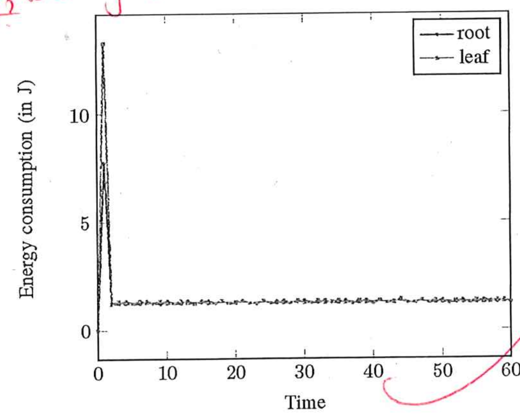


Fig. 4: Energy consumption for root vs leaf node

In this section, we have discussed the obtained results and some intuitive remarks that can be drawn from the observations. For the sake of this study, we present only empirical observations. Additionally, we also attempt to verify our speculations from the obtained results.

will hardly affect the latency ~~results~~ results
→ but the distance influences the propagation error which influences retransmission, latency, etc.

where is the comparison between these 2 modes?

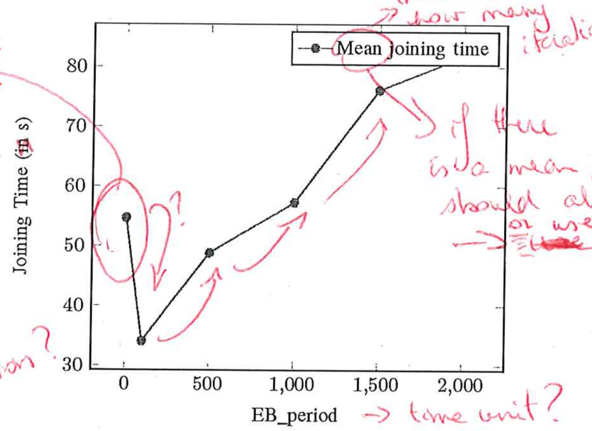
explain why!

→ advertisements at TSCH, RPL, ... which layer other messages that increase the consumption?

For analysis 1, in Fig.2, Fig. 3 and in Fig. 4, we observe a consistent power consumption after establishing the network for both full-6TiSCH-stack as well as TSCH-only. The real difference in power consumption is between the two modes. From Fig. 2 and Fig. 3, we can clearly observe that the energy consumed by the leaf node during network convergence is greater than the root node, however, they converge at a constant consumption level subsequently. So, we can conclude that using a more advanced protocol-stack comes at the cost of aggressive power consumption.

the mean of how many iterations?

No EBs, but they still join? explain.



if there is a mean, there should also be a standard dev. or use boxplot per data point

retransmission?

→ time unit?

Fig. 5: Joining time vs EB period

where is energy consumption?

For analysis 2, we have plotted the average joining time by the leaf node in Fig. 5. As observed, increasing the EB period results in slower association between root and leaf node. But, tuning the EB parameter also induces minimal network congestion and lesser probability of packet collisions. This is crucial in order to reduce the number of rebroadcasts, which would significantly reduce packet drop ratio and improve end-to-end throughput.

In the case of analysis 3, we can only assume that increasing the range would imply increased latency, decreased throughput and thus, increased power consumption.

V. Conclusion

In this report, three preliminary analyses were discussed which examined the energy efficiency and network parameters of low-powered TSCH networks and 6TiSCH networks. Based on the obtained plots, it is shown that both leaf node and root node consumed more energy during network convergence but resolved to a constant value subsequently. The joining time for the participant nodes are extensively recorded and it is shown that the mean joining time is proportional to the EB period of the base station. Additionally, higher EB periods induce a congestion-free networking environment but it is inconsistent and un-scalable since advertisements, route updates and acknowledgements are shared at larger intervals. Similarly, energy consumption is also proportional to the range, precisely the distance between the leaf and the root node. Intuitively, this can be concluded since greater distance would result in larger propagation time; which would require densely modulated signals, hence consuming more electrical energy.

you assume this as you have no data.

References

[1] Roy, A., and N. Sarma. "Energy saving in MAC layer of wireless sensor networks: a survey." National Workshop in Design and Analysis of Algorithm (NWDAA), Tezpur University, India. Vol. 96. 2010.

[2] Piran, Mohammad Jalil, G. Rama Murthy, and G. Praveen Babu. "Vehicular ad hoc and sensor networks; principles and challenges." arXiv preprint arXiv:1108.2776 (2011).

[3] Duquennoy, Simon, et al. "Tsch and 6tisch for contiki: Challenges, design and evaluation." 2017 13th International Conference on Distributed Computing in Sensor Systems (DCOSS). IEEE, 2017.

Appendix

A. Calculating Energy consumption of nodes

this can be in the experiment setup + also explain the other metrics: latency & throughput.

In this section, we will discuss the chronology of computing the energy consumption of the root node and the leaf node. As discussed in section III-A, we firstly use the energest module to dump the log file and extract the time consumed for each state (CPU, LPM and so on). Table I gives a standard convention for profiling sensor devices and for this study, we have considered the values given in the column 'Device Profiling'. The column 'CC2538 datasheet' is also an alternative template, but it is based on an empirical study and beyond the scope of this report.

Next, we compute the energy consumption for each interval as follows:

itr?

$$\forall\ states,\ S\ \ in\ \{CPU, LPM, DLPM, Listen, R_x, T_x\}$$
$$Time_{(S,itr)} = Time\_Log_{(S,itr+1)} - Time\_Log_{(S,itr)}$$
$$E_{total} = \sum_{(S,i=0)}^{n} Device\ Profiles * Time_{(S,i)} \quad (1)$$

i or itr?

→ Joule? where is Volt?

→ explain this better.

TABLE I: Standard Metrices used for computing energy utilization

State	CC2538 datasheet	Device Profiling
CPU	20 mA	15.35 mA
LPM	0.6 mA	9.59 mA
Deep LPM	0.0013 mA	2.58 mA
Listen	24 mA	28.32 mA
Rx	27 mA	30.14 mA
Tx	34 mA	31.12 mA

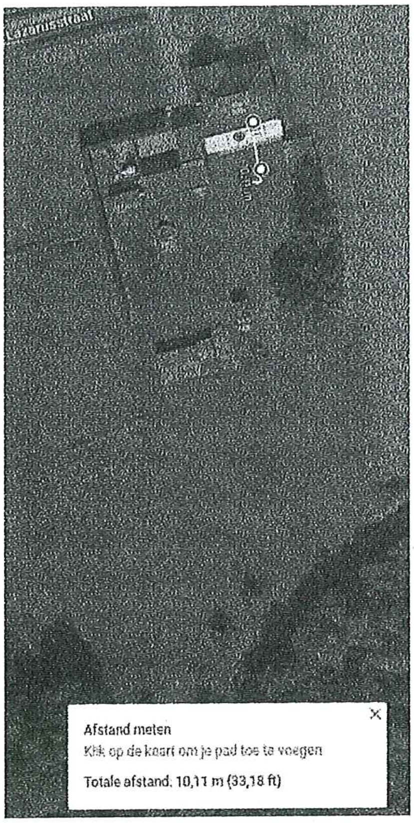


Fig. 6: Satellite photo of the locations of the nodes for 10 metres

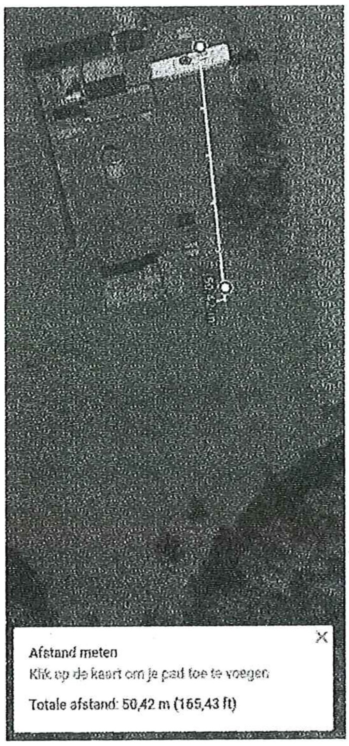


Fig. 7: Satellite photo of the locations of the nodes for 50 metres