TSCH: A look at reliability and energy requirements

Gabriele Paris
Dept physic
University of Antwerp
Antwerp, Belgium
gabriele.paris@student.uantwerpen.be

Pieter Hendriks
Dept computer science
University of Antwerp
Antwerp, Belgium
pieter.hendriks@student.uantwerpen.be

Abstract—In this paper, we assess the energy performance of TSCH and 6TiSCH using the Contiki-NG operating system on Zolertia RE-mote (Rev. B) nodes.
These boards are part of the IoT hardware family, and they allow for the creation of a network minimizing energy consumption. We perform different experiments aiming to evaluate various parameters such as energy consumption or latency of the boards in a two nodes network.
Different measurements are performed on the boards under different conditions such as only the MAC layer enabled compared full-stack, during the joining process, and at various distances. For each experiment, we report and explain how the introduced variations in the network affect the measured values.

Index Terms—6TiSCH, Contiki-NG, Energest, IEEE 802.15.4, Power Consumption, Range, Time Slotted Channel Hopping, 6TiSCH, TSCH, Zolertia

I. INTRODUCTION

The IETF IPv6 over the TSCH mode of IEEE802.15.4e (6TiSCH) working group has standardized a set of protocols to enable low power industrial-grade IPv6 networks. 6TiSCH proposes a protocol stack rooted in the Time Slotted Channel Hopping (TSCH) mode of the IEEE802.15.4-2015 standard, a scheme aiming to guarantee network reliability by keeping nodes time-synchronised at the MAC layer. The latter is accomplished by scheduling, therefore nodes must remain time synchronised throughout the network deployment's life-time. To this end, nodes periodically exchange Enhanced Beacon (EB) packets[8]. It supports multi-hop topologies with the IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL) routing protocol, and is IPv6-ready through 6LoWPAN[8].

Contiki-NG is an operating system for resource-constrained devices in the Internet of Things. Contiki-NG contains an RFC-compliant, low-power IPv6 communication stack, enabling Internet connectivity. The system runs on a variety of platforms based on energy-efficient architectures such as the ARM Cortex-M3/M4 and the Texas Instruments MSP430. The code footprint is on the order of a 100 kB, and the memory usage can be configured to be as low as 10 kB.

This paper serves as an initial look at the performance of TSCH on the Contiki-NG operating system. We study the following cases:

- Energy consumption of TSCH network stack vs 6TiSCH network stack
- Time taken for and energy used during node association with an established TSCH network
- Measure the impact of distance between nodes on energy use and network reliability

The hardware in use are Zolertia RE-Mote boards, designed jointly with universities and industry partners in RERUM European project, to ease the development of private and secure applications for IoT and Smart City applications.
The RE-Mote packs several onboard resources, like an accurate time keeper real time clock (RTC), external watchdog timer (WDT) to force an embedded microprocessor or microcontroller to reset in case of loop, Micro-SD, RF switch, and a Shutdown mode to reduce its power consumption down to 150nA[3].

The paper is subdivided into three main sections: in the first one, we will compare the energy consumption of the leaf and root node (LLC) at first using up to the MAC TSCH layer, and then using the full stack.
In the second section, we will analyze the TSCH joining process and relative energy consumption. Further, in the final section, we will assess the performance of the TSCH mode over various distances.

II. RELATED WORK

A. Accurate energy consumption using Energest

In Section III of this paper, we focus on the analysis of the energy consumption between leaf and root node. This analysis can be done by knowing different parameters like current, voltage, and time. Voltage is always fixed at $3.3V$ due to the power supply and voltage regulator on the board, current should instead be correctly provided in the board datasheet, previous works have shown that real current consumption values are different from the datasheet provided ones[7]. In this paper, we are going to use such current measurements to determine the energy consumption.

B. Throughput Analysis

S. Lee et al present an analysis of the throughput performance of TSCH networks. They show which parameters of the TSCH networks (node count, hop count) influence the throughput metric in what way. They concluded that throughput capability of the network increases as the node count increases. The throughput capability drops when the hop count increases. Our research focuses on different performance characteristics of the TSCH implementation (energy efficiency

rather than throughput), while varying different variables and can be useful alongside these results[6].

C. Join Process Performance

D. De Guglielmo et al analyse the time it takes for nodes to associate with an existing TSCH network. Their work focuses on the time taken for nodes to associate to the network and finds that the amount of channels in the hopping sequence is the most important factor[2]. Our research additionally considers the energy states nodes are in during the join process.

III. ANALYSING THE 6TiSCH ENERGY CONSUMPTION

A. Introduction

In the first analysis, we compared the energy consumption during a certain time period of the entire 6TiSCH stack to when only enabling the TSCH MAC layer (without link-layer security) after network convergence. The full stack can be seen as in Figure 1.

For both analyses, we report on the consumption of the root and the leaf node separately. Following we remark differences in energy consumption between the root and the leaf node in the two different scenarios. The TSCH mode for medium access control (MAC) included in the standard IEEE 802.15.4 has been designed as the multichannel MAC protocol for Low-power and lossy networks (LLNs), a key component of the IoT. Its flexibility makes the TSCH mode a very promising candidate for the future of the MAC layer in LLNs. As such, its performance under different conditions must be assessed, so that accurate guidelines for its application can be drawn[4].
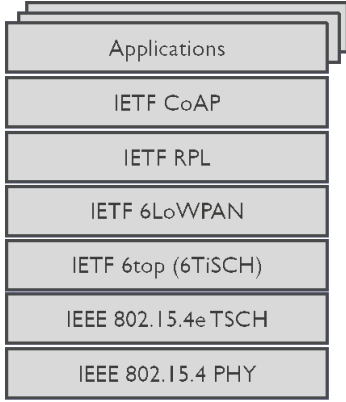

Fig. 1. 6TiSCH Protocol Stack[5].

the physical setup of the experiment consists of two Zolertia Remote RevB boards placed at a fixed distance of approximately 20 cm as shown in Figure 2. There is nothing between the boards hindering communication.

The basic setup for the following analysis is a root (coordinator) node and a leaf node that exchange the same amount of data packets. Both nodes are configured to measure the energy consumption by the Energest[1] module available in
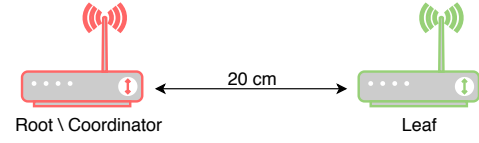

Fig. 2. Network topology for 6TiSCH energy consumption analysis.

Contiki-ng[2]. The Energest module can be used to implement lightweight, software-based energy estimation approach for resource-constrained IoT devices. By tracking the time various hardware states such as the radio is turned on, and by knowing the power consumption of the state, it is possible to estimate the energy consumption[3]. The energy consumption is measured referring to the following formula:

$$E_{tot} = \sum_{s \in state}^{N_s} E_s = \sum_{s \in state}^{N_s} I_s \cdot V_{cc} \cdot t \tag{1}$$

Where V_{cc} is the supply voltage, fixed as a constant at the value $3.3V$, I_s is provided by the Table Table I column *Device profiling*[7] chosen because more accurate than the one provided by the CC2538 datasheet, and t is measured using Energest.

All measurements are performed in a span of 15 minutes each.

State	CC2538 datasheet	Device profiling
CPU	20 mA	15.35mA
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

TABLE I
COMPARISON BETWEEN VALUES FROM THE DATASHEET AND N6705B DC POWER ANALYZER RADIO.[7]

B. Only TSCH MAC layer

In the first part of the experiment, we used only the stack till the MAC layer (TSCH) as shown in Figure 1. In TSCH networks, every node follows a time-synchronized schedule. This schedule instructs every node about exactly what to do and avoids wasting valuable energy. The TSCH schedule is divided into time slots. The duration of a time slot is typically 10 ms or 15 ms and sufficient to transmit a packet of the maximum size of 127 bytes, immediately followed by an optional acknowledgment frame indicating that the packet was successfully received. Multiple time slots are grouped into a slot frame, and the size of a slot frame defines the width of the schedule. These slot frames repeat continuously over time. TSCH also allows one to use multiple frequencies, leading to a two-dimensional matrix of cells. The number of available frequencies actually determines the height of the schedule[1].

---

[1]https://github.com/contiki-ng/contiki-ng/wiki/Documentation:-Energest

[2]https://github.com/contiki-ng

We proceeded as explained in the setup to measure the energy consumption between leaf and root node.

Results, measured after network convergence, are reported in Table II.

Node	CPU	LPM	Deep	Tx	Rx
Leaf	3.330 mJ	314.385 mJ	85.143 mJ	1.132 mJ	993.527 mJ
Root	3.330 mJ	314.385 mJ	85.143 mJ	1.132 mJ	993.527 mJ

TABLE II

AVERAGE POWER CONSUMPTION COMPARISON BETWEEN ROOT AND LEAF NODE PER STATE, TSCH MAC ONLY.

What we can see is that data from root and leaf nodes show no variation, and the energy consumed from both of them is the same.

In a TSCH network The root needs to control the way the network is formed, including how new nodes join and how already joined nodes advertise the presence of the network, this is all encoded in the TSCH RFC[3].

The root node during network convergence needs to:

- Define the Information Elements included in the Enhanced Beacons, advertising the presence of the network;
- for a new node define rules to process and filter received EBs;
- Define the joining procedure. This might include a mechanism to assign a unique 16-bit address to a node and the management of initial keying material;
- Define a mechanism to secure the joining process and the subsequent optional process of scheduling more communication cells.

After the network has been formed it must be mantained, this implies that the root node must:

- Manage each node's time source neighbor;
- Define a mechanism for a node to update the join priority it announces in its EB;
- Schedule transmissions of EBs to advertise the presence of the network.

We can therefore say that, atleast for two node comunicating for such time, after network convergence, the workload on the root is not enought to show a difference in energy consumption during the stated time spam.

C. Full stack

The next experiment is as cited in Section III-A related to the energy consumption once the full 6TiSCH stack has been enabled (except for the security layer).

The setup of this experiment is the same to the previous one. Two boards 20cm apart from each other (2) are running the same source code.

In this scenario, we have a coordinator node and a leaf node. No messages are exchanged between the two if not for standard 6TiSCH service messages.

We proceeded as explained in the setup to measure the energy consumption between leaf and root node.

Results are reported in Table III.

[3] https://tools.ietf.org/html/rfc7554

Node	CPU	LPM	Deep	Tx	Rx
Leaf	5.00 mJ	313.341 mJ	85.143 mJ	80.121 mJ	98.362 mJ
Root	3.890 mJ	310.560 mJ	84.209 mJ	0 mJ	33.879 mJ

TABLE III

AVERAGE POWER CONSUMPTION PER STATE BETWEEN ROOT AND LEAF NODE, FULL 6TiSCH STACK.

D. Final comparison

As a final comparison, we compared the total energy usage between root and leaf nodes in the two configurations. What we measured has been reported as Table IV.

Node	MAC only	Full stack
Leaf	1.398 J	0.582 J
Root	1.398 J	0.433 J

TABLE IV

ENERGY COMPARISON BETWEEN LEAF AND ROOT NODE IN MAC ONLY AND FULL STACK CONFIGURATIONS.

IV. ENERGY CONSUMPTION DURING THE TSCH JOIN PROCESS

V. IMPACT OF DISTANCE ON ENERGY CONSUMPTION AND NETWORK RELIABILITY

A. Configuration

For the most part, the default configuration of Contiki-NG is used. Any non-default values we've used, as well as other important details, are discussed here.

1) Network topology: We study a two-node TSCH network. Both nodes are Zolertia RE-mote (Rev. B) boards. One will act as the TSCH coordinator, the other as a leaf node. In all tests performed in this section, the traffic is sent from the leaf node to the root node.

The nodes always remain within line of sight, at equal heights (elevated 1 meter above the ground). The only physical parameter being varied (other than those outside our control) is the distance between them. For our range analysis, we consider four distances: 1 meter, 10 meters, 50 meters and 100 meters. The node positions at each distance were marked, to ensure they were positioned as reliably as possible.

2) TX power: For each distance in the experiment, we also consider the impact of the TX power parameter. This value sets the intensity of the signal transmission. A larger TX power value leads to increased energy consumption[1]. The amount of extra energy consumed by varying this value is left outside the scope of this study, because our study is limited to observing time spent in each energy state. The associated current draw for each energy state (used to compute final energy consumption values) is obtained through spec sheets or device profiling results (see Table I). We consider 3 distinct values for this parameter: 7 dBm, 0 dBm and -7 dBm.

3) Traffic characteristics: The traffic is uniform, 1 packet is sent per second. The packets contain 64 bytes of data. The packet size was chosen because 64 bytes is sufficiently large to contain sensor measurement data while remaining small enough to be transmitted over the TSCH layer without

fragmentation. Each experiment will send 60 packets so that we may average the performance for each parameter set over that data.

4) TSCH schedule: We use a very simple schedule for our experiments. There is a single advertising-only cell at (0, 0) and a cell used to transmit our data packets at (1, 1). The (0, 0) cell is used for beacons, so that the leaf node can join the network. It is not used for any of our measurements. All data we collect is sent on the (1, 1) cell. That cell is configured as a transmit-only (to the coordinator) cell in the leaf node and a receive-only (from the leaf) cell in the coordinator node. This setup is illustrated in Figure 3. The size of the schedule is left as its default value, that is, 7 slots per frame, 4 channels.
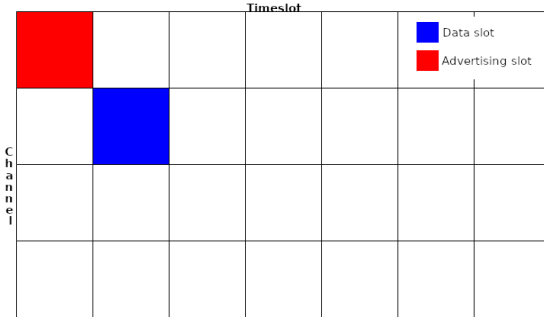

Fig. 3. TSCH Schedule as used in the range analysis experiments

5) Network stack: For these experiments, most of the default Contiki-NG network stack has been disabled. We're interested in the performance of TSCH, so our experiments interact directly with the MAC layer. To facilitate this, Contiki-NG is built with the following Makefile variables set:

- MAKE_MAC=MAKE_MAC_TSCH
- MAKE_NET=MAKE_NET_NULLNET
- MAKE_ROUTING=MAKE_ROUTING_NULLROUTING

MAKE_MAC_TSCH configures the build system to build the TSCH MAC layer. MAKE_NET_NULLNET and MAKE_ROUTING_NULLROUTING essentially disable their respective functionalities. Placeholder functions (no side-effects, always return success) are defined in Contiki-NG, but these are not used by our implementation.

6) Preprocessor definitions: The following values are defined in the preprocessor (some through Makefile, others through header definitions) in order to configure the behavior we want. Each parameter is briefly explained below.

- ENERGEST_CONF_ON=1
  Enable the energest package to allow analysis of power use.
- TSCH_SCHEDULE_CONF_WITH_6TISCH_MINIMAL=0
  Disable creation of a default TSCH schedule. Instead we create our own, as described in Section V-A4.
- LLSEC802154_CONF_ENABLED=0
  Disable Link Layer security.
- TSCH_CONF_JOIN_SECURED_ONLY=0
  Allow TSCH to join unsecured networks.

B. Results

In our current test scenario, there is no meaningful difference between the various power levels when it comes to
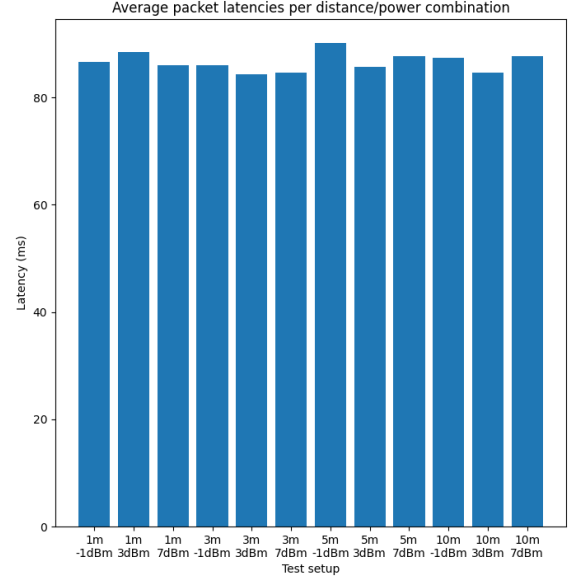

Fig. 4. The average packet latencies for each distance/power combination
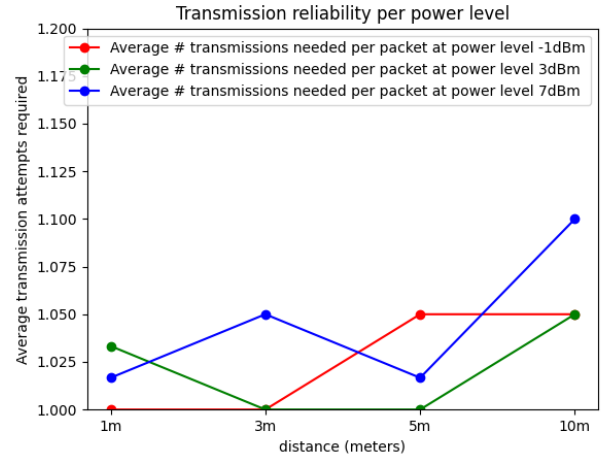

Fig. 5. Overview of amount of transmission attempts are made on average to send a single packet

transmission reliability. The amount of transmission attempts that must be made per packet are essentially identical (and almost equal to 1) for all tested scenario's.

Another observation we can make, which is influenced by the transmission reliability, is the packet latency. In Figure 4, the latencies are shown, averaged over the experiment run, per tested scenario.

The latencies shown in Figure 4 are grouped very closely together, just over 80 milliseconds. Since the default TSCH slot length is 10 milliseconds, and the slot frame has 7 slots, it takes just over the duration of a slotframe for a packet to go from being queued in the leaf node to arriving at the root (give

or take clock drift since the last TSCH time synchronization). This seems a little high, given the limited amount of traffic being generated.
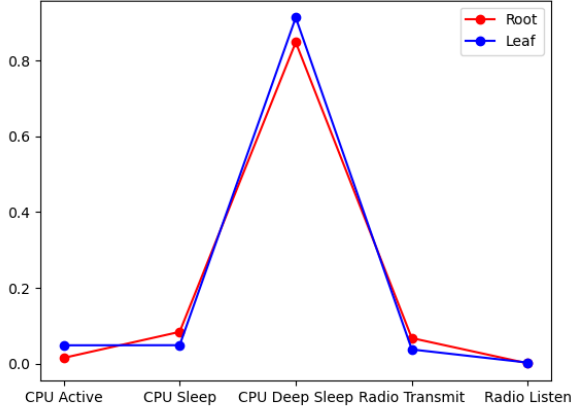

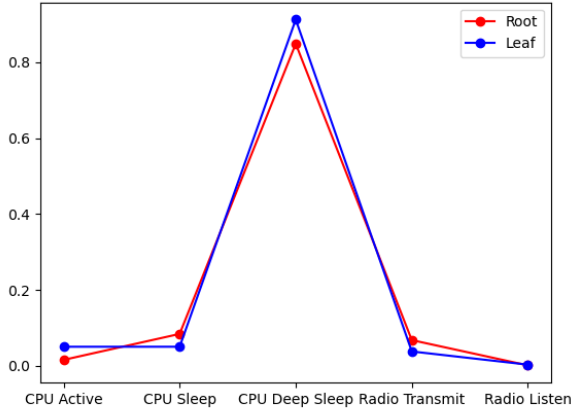Fig. 6. The power states for the nodes when 10 meters apart

Fig. 7. The power states for the nodes when 1 meter apart

The power results, as the latency and delivery results, are incredibly similar. This is trivially clear when comparing Figure 7 and Figure 6 - they are nearly identical.

Using the results from profiling the device (as in Table I), and the Energest results we've obtained, we can compute the amount of power that is needed for each device. The leaf, overall, used approximately 3.4 J. The root used approximately 4.1 J.

With a total data size of 360 kilobits (12 tests * 60 packets * 64 bytes per packet * 8 bits per byte / 1024 bits per kilobit), that corresponds to roughly $9.6\frac{mJ}{kbit}$ for the leaf and $11.5\frac{mJ}{kbit}$ for the root.

C. Conclusion

The tests we ran do not offer a clear insight to the performance of TSCH in various scenarios. The different scenarios appear too close together to distinguish meaningful differences. Due to the inherent noisiness of the wireless medium, the tests should be expanded to be able to clearly spot a pattern. The duration should be extended and the difference between the values under test should increase. This way, if there is a pattern, it is less likely that a difference in noise in the medium will cause the test results to be essentially indistinguishable.

VI. FUTURE WORK

1) Varied packet size: In this study, we've only looked at the performance characteristics of the network at various distances for packets carrying 64 bytes of data payload. An interesting follow-up might be the analysis of how the results change when the packet size is varied as well. A sensor network doing less complex measurements could definitely get by with smaller packets, for example. Smaller packets lead to shorter transmission times. This may reduce the error rate during transmission and thus could potentially achieve the same reliability at a lower TX power.

2) Individually varied TX power: We've only considered the case where TX power is varied on both nodes. A potentially interesting follow-up to this is to consider the impact varying the parameter on one of the nodes has. This would clarify if, in order to ensure satisfactory network performance, TX power must be increased for all nodes or only for some. Our results have shown that larger TX power results in fewer retransmission (at the cost of increased power use per transmission[1]), but we've not studied the trade-off in more detail. It is conceivable to have the root node be connected to a power source, while other nodes are running on battery, for example. So an interesting study would be to only increase TX power on the root node and see if (and to what degree) that improves performance the reliability, since its energy consumption is less relevant.