A Realistic Energy Consumption Model for TSCH Networks

Xavier Vilajosana, *Member, IEEE*, Qin Wang, Fabien Chraim, Thomas Watteyne, *Member, IEEE*, Tengfei Chang, and Kristofer S. J. Pister

***Abstract*— Time slotted channel hopping (TSCH) is the highly reliable and ultra-low power medium access control technology at the heart of the IEEE802.15.4e-2012 amendment to the IEEE802.15.4-2011 standard. TSCH networks are deterministic in nature; the actions that occur at each time slot are well known. This paper presents an energy consumption model of these networks, obtained by slot-based "step-by-step" modeling and experimental validation on real devices running the OpenWSN protocol stack. This model is applied to different network scenarios to understand the potential effects of several network optimization. The model shows the impact of keep-alive and advertisement loads and discusses network configuration choices. Presented results show average current in the order of 570 μA on OpenWSN hardware and duty cycles 1% in network relays in both real and simulated networks. Leaf nodes show 0.46% duty cycle with data rates close to 10 packets per minute. In addition, the model is used to analyze the impact on energy consumption and data rate by overprovisioning slots to compensate for the lossy nature of these networks.**

***Index Terms*— IEEE802.15.4e, synchronization, wireless sensor networks, duty cycle, energy consumption, TSCH.**

I. Introduction

TIME Slotted Channel Hopping (TSCH) mesh networks are becoming central for wireless industrial deployments as they are able to achieve 99.999% reliability [1] with minimal power consumption. Standards such as WirelessHART [2], ISA100.11a [3] and IEEE802.15.4e [4] are rooted in the TSCH medium access technique. In a TSCH network, nodes are synchronized, and time is split into time slots, each typically 10*ms* long. Time slots are grouped into a slotframe which continuously repeats over time.

The network's communication schedule instructs each node what to do in each time slot: send to a particular neighbor, receive from a particular neighbor, or sleep [5], [6]. Channel diversity is obtained by specifying, for each send and receive slot, a channel offset. The same channel offset is translated into a different frequency on which to communicate at each iteration of the slotframe. The resulting channel hopping communication reduces the impact of external interference and multipath fading, thereby increasing the reliability of the network [7].

All nodes in the TSCH network are synchronized. Because communication occurs at a well-defined times within a time slot, the sender nodes know exactly when to transmit. If the sender and receiver nodes were perfectly synchronized, the receiver node would turn its radio on at exactly the instant when the transmitter starts emitting. The sender's and receiver's radio would be on only for the duration of the packet being transmitted. After the transmission of a packet, both nodes can switch their radios off to save energy or sleep a few milliseconds before repeating the same process in order to receive/transmit an acknowledgment (ACK). This simplistic scenario is the optimal solution in terms of energy consumption that can be achieved in a communication between two nodes, since it minimizes the time during which the transceivers are on.

However, as clocks between neighbor nodes in a network drift (30 *ppm* relative drift is a typical value [8]), a small "guard time" is required at the receiver end to account for relative desynchronization [9]. Acknowledgments follow a similar scheme: a guard time is introduced around the ideal reception moment.

Although the wireless medium is lossy in nature, TSCH networks are deterministic in their scheduling. The energy consumed by a node can be modeled precisely, by profiling the actions that are carried out during each slot. The aim of this article is to present an energy consumption model for TSCH networks, and to analyze the impact of control signaling and the communication schedule on this energy consumption.

Kohvakka et. al. [10] model the energy consumption of the legacy IEEE802.15.4-2006 MAC protocol operating in slotted (beaconed) mode. A similar model is later used to predict the energy consumption per received bit as a function of

X. Vilajosana is with BSAC, University of California, Berkeley, CA 94720 USA, and also with the Universitat Oberta de Catalunya, Barcelona 08035, Spain, and also with Worldsensing, Barcelona 08013, Spain (e-mail: xvilajosana@eecs.berkeley.edu).

Q. Wang is with BSAC, University of California, Berkeley, CA 94720 USA, and also with the University of Science and Technology, Beijing 100083, China (e-mail: wangqin@ies.ustb.edu.cn).

F. Chraim and K. S. J. Pister are with BSAC, University of California, Berkeley, CA 94720 USA (e-mail: chraim@eecs.berkeley.edu; pister@eecs.berkeley.edu).

T. Watteyne is with Linear Technology/Dust Networks, Hayward, CA 94544 USA (e-mail: watteyne@linear.com).

T. Chang is with the University of Science and Technology, Beijing 100083, China (e-mail: tengfei.chang@gmail.com).

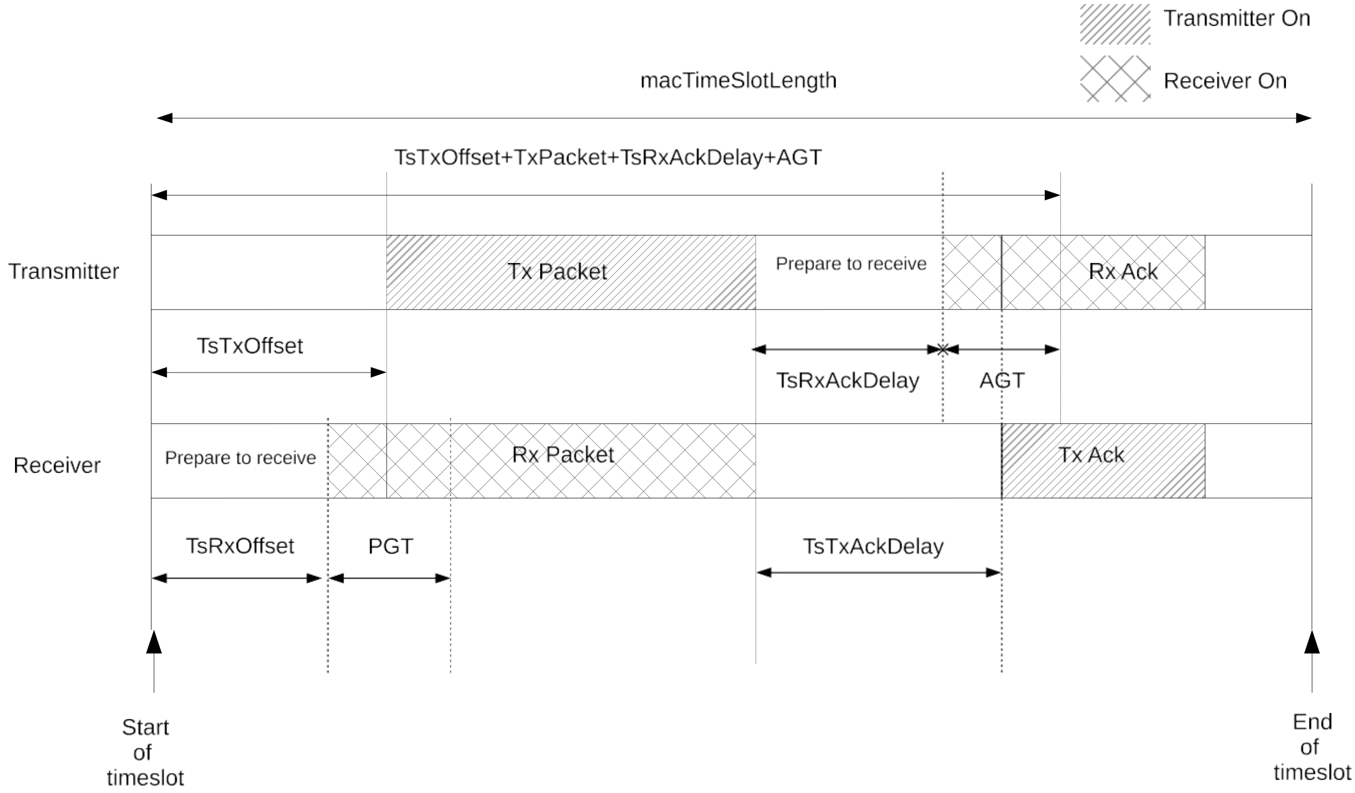


Fig. 1. Timeslot template for a transmitter (top) and receiver node (bottom). This figure shows the timing breakdown for these two types of slot, highlighting where transmission occurs [4].

traffic load and packet size [11]. Wang et. al. [12] present a general energy consumption model for WSN devices based on their hardware architecture. This model reflects the energy consumption in various functioning states, and during transitions between states of the devices leading to a very accurate energy modeling. More recently, Casilari et. al. [13] present a model for non-slotted CSMA/CA based IEEE802.15.4/ZigBee networks.

Khader and Willig [14] present an energy consumption model for WirelessHART TDMA networks, which addresses objectives different from the ones presented here. The authors focus on analyzing the WirelessHART protocol and identifying the main aspects that contribute to the energy consumption it induces. They focus is on exploiting different sleep modes on the CC2420 transceiver in order to reduce this consumption. In contrast, our goal is to develop an energy consumption model based on a fine-grained energy analysis of each type of slot in such networks, in the hope of providing a tool for consumption estimation prior to real network deployments.

II. Energy Model

As a device joins a TSCH network, it obtains information about the duration of each time slot, and the number of slots in a slotframe. In each slot, the node can either transmit, receive, or keep its radio off. A scheduling entity is responsible for building the schedule which will satisfy the bandwidth and latency needs of the different flows in the network. Throughout the lifetime of the network, this entity modifies this schedule to adapt to changes in the topology and changes of the traffic requirements. The schedule allows for a fine-grained trade-off between latency, bandwidth, redundancy and power consumption. Several scheduling approaches have been proposed, both centralized [15] and distributed [16], [17], and are currently being standardized [18]. The energy consumption of a mote is the sum of the energy consumed in each slot. To build the energy consumption model, we start by investigating the energy consumed in each type of slot, both through analysis and experimentation.

A. TSCH Slot Template

This article focuses on IEEE802.15.4e networks, but the same principle can be applied directly to other TSCH standards such as WirelessHART [2].

In IEEE802.15.4e, there are 6 different types of time slots:

- `TxDataRxAck`: A timeslot during which the node sends some data frame, and receives an acknowledgment (ACK) indicating successful reception.
- `TxData`: Similar to the previous, but no ACK is expected. This is typically used when the data packet is broadcast.
- `RxDataTxAck`: A timeslot during which the node receives some data frame, and sends back an ACK to indicate successful reception.
- `RxData`: Similar to the previous, but no ACK is exchanged.
- `Idle`: Time slot during which a node listens for data, but receives none.
- `Sleep`: Time slot during which the node's radio stays off.

TABLE I

MAPPING FROM PERIODS IN TEMPLATE TO STATES OF MOTE MODULES

Period in Template	State of motes	μP state	Radio state
StartOfTimeslot	NewSlot	Active	Sleep
TsTxOffset	TxDataOffset PostTxDataOffset TxDataPrepare PostTxDataPrepare	Active Sleep Active Sleep	Sleep Sleep ToReady Ready
TxPacket	TxDtataStart TxData PostTxData	Active Active Sleep	ToTx Tx Tx
TsRxAckDelay	TxRxAckOffset PostTxRxAckOffset	Active Sleep	Sleep Sleep
AGT	RxAckPrepare RxAckReady	Active Sleep	ToListen Listen
RxAck	RxAckStart RxAck PostRxAck	Active Sleep Active	Rx Rx Sleep
BeforeEnd	Sleep	Sleep	Sleep
EndOfTimeslot	EndSlot	Active	Sleep

Fig. 1 presents a detailed breakdown of the activity of a node in a `TxDataRxAck` slot at the transmitter, and a `RxDataTxAck` slot at the receiver [4]. The transmitter node starts by waiting for *macTsTxOffset*, during which it prepares the data to send, and configures the radio according to the frequency inferred from the schedule. The radio then turns on and transmits the packet exactly *macTsTxOffset* from the beginning of the slot. After the last byte of the packet has left the radio, the transmitter gives the receiver some time to prepare the acknowledgment packet by waiting for *macTsRxAckDelay*. If no acknowledgment is received after the *Acknowledgment Guard Time (AGT)* period, the device turns off the radio and considers the transmission failed.

On the receiver's side, the device waits for *macTsRxOffset*, then turns its radio on, listening for a packet. If after the *Packet Guard Time (PGT)*, no packet is received, the device turns its radio off for the remainder of the slot. If a valid packet is received, the node waits *macTsTxAckDelay* after the reception of the last byte, before turning its radio on and sending the acknowledgment.

B. Slots Energy Consumption Modeling

In a typical node, the two components which consume the most are the micro-controller and the radio. These components can either be two separate chips interconnected by some digital bus on a board, or grouped in a single System-on-Chip. To accurately model the energy consumption in this node, one must look at a detailed breakdown of the various states each module enters, for each type of slot. As the micro-controller and radio change state, their consumption varies. Table I shows their state during the different phases of a `TxDataRxAck` slot. The charge (in Coulombs) drawn from a battery during the execution of this slot is the sum of the charge in each of these steps.

$$Q_{Tx\ Data\ Rx\ Ack} = \int_{0}^{T\,s\,Slot\,Duration} I_{Tx\ Data\ Rx\ Ack}(t)dt \qquad (1)$$

The same analysis holds when computing the energy consumption of the other 5 slot types, and extract $Q_{Rx\ DataT\ x\ Ack}$, $Q_{Tx\ Data}$, $Q_{Rx\ Data}$, Q_{Idle}, and Q_{Sleep}.

TABLE II

CURRENT DRAWN BY THE ATMEL AT86RF231 RADIO CHIP FOR DIFFERENT STATES (THEORETICAL AND MEASURED)

GenericMode	AT86RF231 Mode	Current	Measured
Sleep	$TRX\ OFF$	0.4mA	0.49mA
ToReady	$TRX\_OFF \Rightarrow PLL\_ON$	5.6mA	N/A
Ready	PLL_ON	5.6mA	5.4mA
Tx	$BUSY\_TX$	11.6mA (0dBm)	13.7mA (0dBm)
ToListen	$TRX\_OFF \Rightarrow PLL\_ON \Rightarrow RX\_ON$	12.3mA	N/A
Listen	RX_ON	12.3mA	11.6mA
Rx	RX_ON	12.3mA	11.6mA

TABLE III

CURRENT DRAWN BY THE MSP430 AND STM32 MICROCONTROLLERS FOR DIFFERENT STATES (THEORETICAL AS DEFINED BY THE DATASHEET (DS) AND MEASURED EXPERIMENTALLY (EXP))

Generic Mode	MSP430 DS	MSP430 Exp	STM32 DS	STM32 Exp
Active	7.54mA@16Mhz N/A	7.4mA@16MHz 3.7mA@8MHz	27mA@72MHz 7.4mA@16MHz 4.6mA@8MHz	32mA@72MHz N/A N/A
Sleep	$1.1\mu A$	N/A	$14\mu A$	N/A

TABLE IV

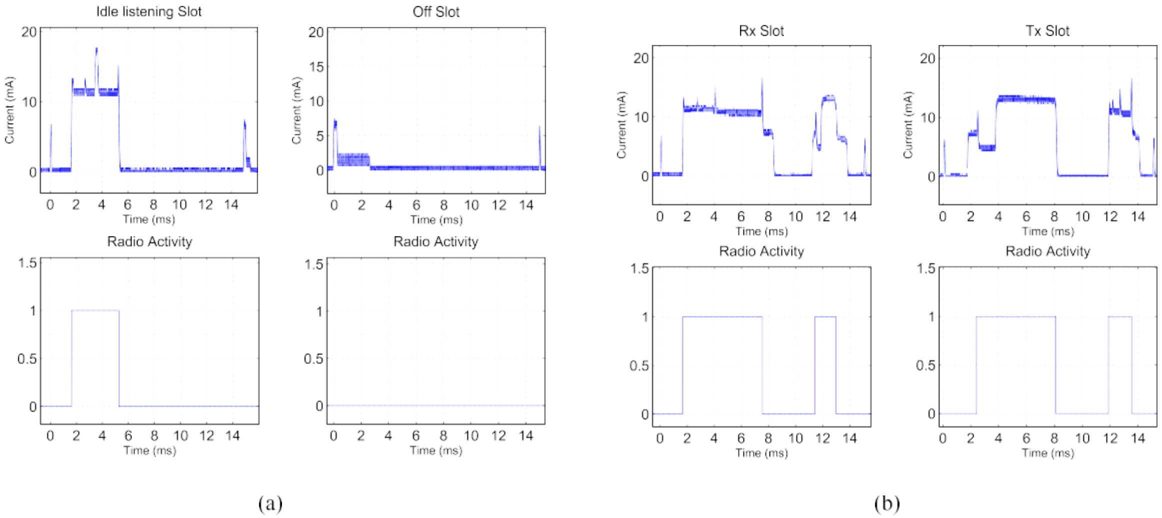
SLOT TIMING, AS IMPLEMENTED IN OPENWSN.

Timing Constant	**Value**
TsSlotDuration	$15000\mu s$
TsTxOffset	$4000\mu s$
TsRxOffset	$305\mu s$
PGT	$2600\mu s$
AGT	$1000\mu s$
TsTxAckDelay	$4000\mu s$
TsRxAckDelay	$TsTxAckDelay - AGT/2\ \mu s$

This paper focuses on two hardware platforms made with commercial off-the-shelf (COTS) components, which run the OpenWSN reference TSCH implementation [19]. These platforms are representative of spectrum of nodes available. OpenMoteSTM is the "high-end" node, which features a STMicroelectronic STM32F103RB 32-bit microcontroller at the Atmel AT86RF231 IEEE802.15.4-compliant radio. GINA [20] is the "low-end" node, with a Texas Instruments MSP430F2618 16-bit microcontroller, and the same Atmel AT86RF231 radio. Table II and Table III list the the current draw of those components in the OpenWSN implementation.

C. Experimental Verification

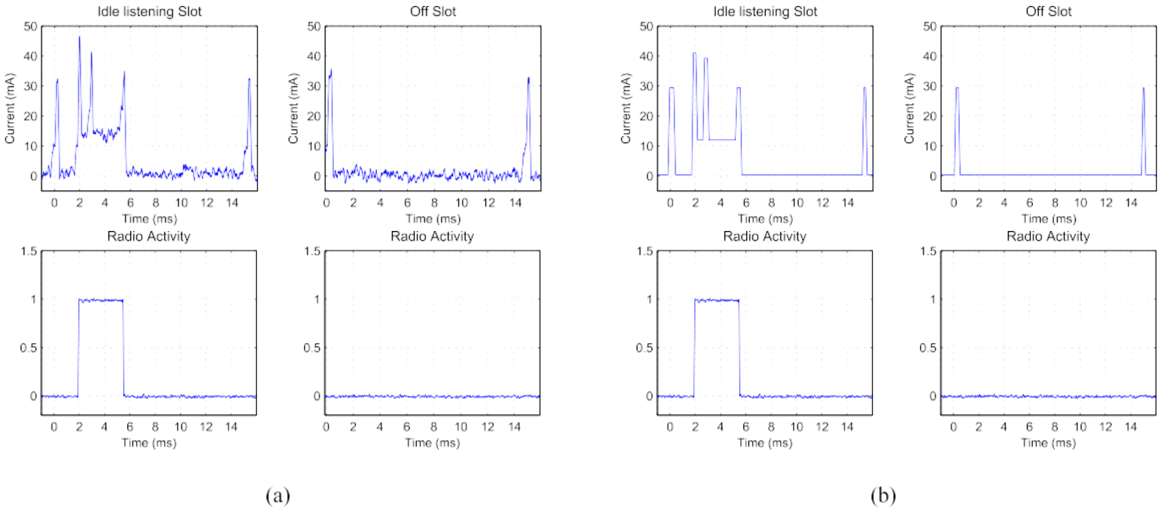
To verify the validity of the energy model, the energy consumption for three different types of slots is measured on the GINA and OpenMote-STM32 platforms, and compared to the value computed using the model. In the experiments on both GINA and OpemMote-STM32, the same slot timings are used as shown in Table IV. Fig. 2(a) shows the current drawn by a GINA mote during an idle slot. At $T_{TsT\ xof\ f\ set} - T_{RxGT}/2 - T_{Rx\ Data\ Prepare}$, the mote's microcontroller turns on to prepare the radio and turns it on at $T_{TsT\,xof\,f\,set} - T_{RxGT}/2$



(a)

(b)

Fig. 2. Measured current draw on a GINA mote. (a) Idle listen and off slots. (b) Transmission and reception slots.



(a)

(b)

Fig. 3. Current draw during idle listening and off slots, on an OpenMote-STM32. (a) Measured. (b) Computed using the model presented in this article.

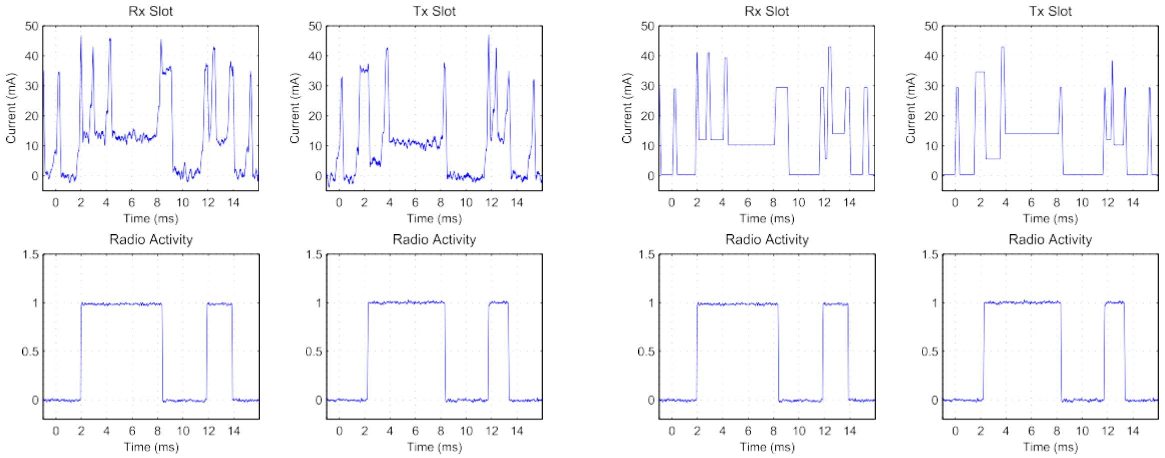
for reception. After T_{RxGT} the radio is switched off as no packet is received.

Fig. 2(b) shows the current drawn by a GINA mote during a transmission slot. The micro-controller is turned on at $T_{TsTxoffset} - T_{TxDataPrepare}$, allowing enough time to prepare the data. At that time, the radio is turned on and the packet is loaded into the radio. The first current spike in the figure at $0.047ms$ is measured when the radio is being turned on. The bytes of the packet are then loaded into the radio's transmit buffer. At $T_{TsTxoffset}$, the packet is sent. Once the radio is done transmitting, the radio is switched off for the $T_{TsRxAckDelay}$ period. A little before the ACK is expected, the radio is turned on again to listen. After the reception of the ACK, the radio is switched off.

Similarly, Fig. 2(b) presents the current drawn by a GINA mote during a reception slot. The mote sleeps until $T_{TsTxoffset} - T_{RxGT} - T_{RxDataPrepare}$. The micro-controller then switches on to configure the radio to the right frequency. At $T_{TsTxoffset} - T_{RxGT}/2$, the radio starts listening. After listening and receiving for a total of $4.6ms$, the packet is completely received and the micro-controller and radio are turned off during the $T_{TsTxAckDelay} - T_{TxAckPrepare}$ period. The ACK is then loaded into the radio and transmitted at $T_{TsTxAckDelay}$, after which the micro-controller and radio are switched off.

The energy consumption for the three remaining types of slots is presented in Figs. 3(a) and 4(a).

These experimental results are compared to the results calculated by implementing the energy model in Matlab. The values corresponding to GINA and OpenMote-STM32 have been selected from Table II and Table III, and introduced in the calculation. The calculated results are shown in Fig. 3(b) and Fig. 4(b). The comparison between measured and calculated values shows a good match between the model and the experimental values. In addition, Table V shows the charge (in μC) drawn by the device, both by experiment and through model-based calculation. The difference between the measurement and the calculation comes from multiple sources, e.g. the measurement error (equipment imprecision when measuring currents of the order of μA), little differences on timing within the slot and the difference between typical current consumption values reported by the manufacturer as



(a)

(b)

Fig. 4. Current draw during transmission and reception slots, on an OpenMote-STM32. (a) Measured. (b) Computed using the model presented in this article.

compared to the ones actually consumed during execution (see Table II and Table III).

D. Slot-Frame Energy Consumption Modeling

We can use the energy model of the different slots to determine how much energy is used during each slotframe, and compute the expected battery lifetime of the network. Equations (2) through (7) define how much energy is consumed in a slotframe for the different types of slots. Subscripts *a* refers to available slots while subscript *u* refers to used slots. Consider PDR as Packet Delivery Ratio defined as the number of packets being acknowledged divided by the number of packets being sent by a node. In the following equations, *N* stands for *number of* and the subscripts define the type of slot being considered (e.g $N_{aRxT\ x}$ stands for *the number of available slots where the node will be listening for a packet and will send and acknowledgment (ACK).*)

$$Q_{FIdle} = \frac{((N_{aRxTx} - N_{uRxTx}) + N_{aRx} - N_{uRx})) \times Q_{idle}}{PDR} \quad (2)$$

Eq. 2 defines the contribution of idle slots to the total charge drawn in a slotframe. The number of slots that are in the idle state are those that have been configured on the schedule as *RxDataTxAck* or *RxData*, but during which no data is received.

$$\begin{aligned} Q_{FSleep} &= (N_{sleep} + (N_{aTxRx} - N_{uTxRx}) \\ &+ (N_{aTx} - N_{uTx})) \times Q_{sleep} \end{aligned} \quad (3)$$

Eq. 3 defines the contribution of *Sleep* slots. As in (2), the *TxDataRxAck* slots ($N_{aT\ x\ Rx}$) that are not used $N_{uT\ x\ Rx}$ are in *Sleep* state. In this case, during a *TxDataRxAck* slot, the transmitter has no data to send and the radio is therefore never turned on.

$$Q_{FTxRx} = \frac{N_{uTxRx} \times (\frac{NBSent}{Max\ Pkt\ Sz} \times Q_{Tx} + (Q_{TxRx} - Q_{Tx}))}{PDR} \quad (4)$$

Eq. (4) defines the contribution of the N_{uTxRx} *TxDataRxAck* slots which are used. In that case two considerations are taken, the first one involves the number of bytes being sent, $NBSent$ which is considered with respect to the maximum packet size as our energy consumption measurements have been done with *Max Pkt Sz*[1] packets. In addition the Probability Delivery Ratio (PDR) expected by the network is considered. Analogously, Eq. (5) describes the energy consumed in slots that are of type *TxData*.

$$Q = N_{FTx} \times (\frac{NBSent}{Max\ Pkt\ Sz} \times Q_{Tx})PDR \quad (5)$$

Eq. (6) computes the contribution of *RxDataTxAck* slots. The number of used slots is defined as $N_{uRxT\ x}$, in addition the number of bytes being sent is also considered.

$$Q_{FRxTx} = N_{uRxTx} \times (\frac{NBSent}{Max\ Pkt\ Sz} \times Q_{Rx} + (Q_{RxTx} - Q_{Rx})) \quad (6)$$

Eq. (7) computes the contribution of *RxData* slots.

$$Q_{FRx} = N_{uRx} \times \frac{NBSent}{Max\ Pkt\ Sz} \times Q_{Rx} \quad (7)$$

Eq. (8) defines the total charge drawn during a slotframe, and is the sum of the contributions by the different types of slots.

$$\begin{aligned} Q_{slotframe} = &\ Q_{FIdle} + Q_{FSleep} \\ &+ Q_{FTxRx} + Q_{FTx} + Q_{FRxTx} + Q_{FRx} \end{aligned} \quad (8)$$

Finally, (9) defines the battery lifetime of a node, in days, assuming it runs from a 3.6V power supply.

$$lf = \frac{B_{capacity} \times 3.6}{Q_{slotframe}} \times \frac{Length_{slot} \times Length_{slotframe}}{3600 \times 24} \quad (9)$$

E. Relay and Leaf Node evaluation

In order to support our realistic energy consumption model a small network composed of five GINA motes running the OpenWSN [19] protocol stack has been built. A topology is forced so that one of the motes is a relay to the base station and the rest of the motes are configured to be leaf nodes as depicted by Fig. 5. Leaf motes are configured to send a

[1] 127 bytes in IEEE802.15.4e.

TABLE V

Measured and Simulated Charge Drawn for Each Types of Slot, in μC

	Measured		Simulated	
State	GINA	OM-STM32	GINA	OM-STM32
Idle	47.9	101.1	54.1	85.2
Sleep	4.9	37.8	8.2	9.2
TxDataRxAck	92.6	161.9	103.3	151.2
TxData	69.6	119.2	76.7	123.1
RxDataTxAck	96.3	217.0	105.2	175.9
RxData	72.1	154.8	78.0	125.0

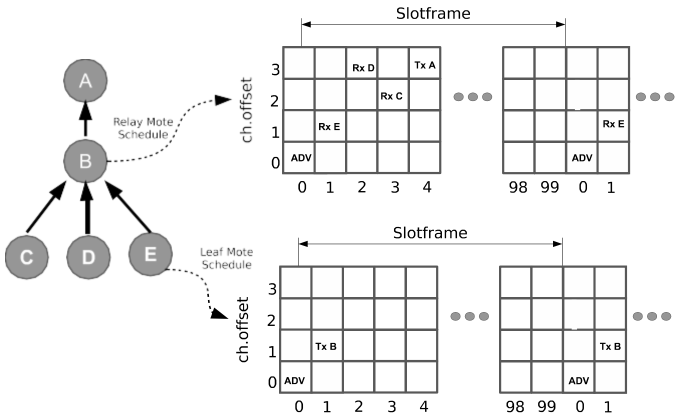


Fig. 5. Experimental setting diagram. The schedule of Mote B is configured so it can relay information of the leaf nodes. Leaf nodes are configured to be able to send a packet once every 4 slotframes. Empty slots are Sleep slots. Slotframe is composed of 100 slots of 15ms each.

packet every 4 seconds leading to at most four packets in a slotframe at the relay mote. The setup has been built in our laboratory, all motes have been installed in a well known position and with a clear line of sight between them. Static schedules have been pre-configured at each node according to Fig 5, no overprovisioning have been configured although the data rate in leaf nodes has been configured to be 1 packet every 4 seconds, considerably lower than the supported data rate. Note that Tx slots that are not used are considered to be Sleep slots having the minimal energy consumption. No other IEEE802.15.4 devices were operating in the laboratory. Channel hopping has been used to mitigate the effect of multipath and external interference [7] and we have considered the expected channel errors to be negligible.

The experiment measures the charge drawn by both relay and leaf motes, as well as the radio duty cycle during a complete slotframe. The results are presented in Table VI. Simulation and experimental results match with an average current consumption of 581.9 μ A for the relay mote in the experimental setup and 569.8 μ A for the simulated setup. The leaf mote consumes less energy (455.0 μ A and 415.4 μ A respectively) due to less active slots in its schedule. The radio duty-cycle is computed in both cases (experimental and simulated) showing a clear match between our model and the experimental setup.

III. TSCH Optimizations

The energy consumption model proposed in this paper can be used to guide TSCH network configuration. In this section, we present two example cases. The GINA platform is assumed in this evaluation.

TABLE VI

Charge by Slotframe. Slotframes Are Composed of 100 Slots of 15ms

	Measured		Simulated	
Node Type	Relay	Leaf	Relay	Leaf
Slotframe Charge Drawn (μC)	872.8	682.5	854.7	623.1
Energy Consumed (μA)	581.9	455.0	569.8	415.4
Radio Duty-Cycle in %	1.14	0.47	1.12	0.47

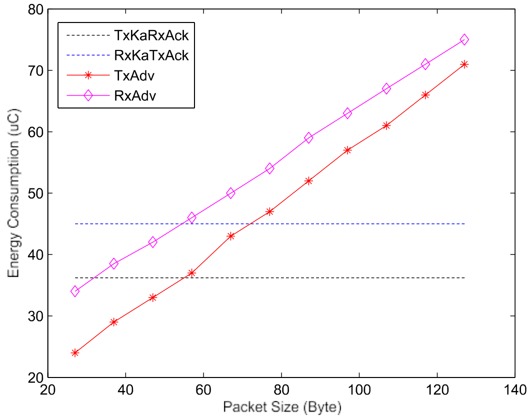


Fig. 6. Energy consumption by Keep-alive and Advertisement for GINA.

A. Synchronization Policy

IEEE802.15.4e defines two schemes to maintain one-hop synchronization. One is advertisement (ADV)-based, and another one is keep-alive (KA)-based. By applying the energy consumption model defined in Section II, we can compare these approaches to understand the impact of each method in terms of energy consumption.

Considering a guard time of 2600μs and a clock drift of ± 30 ppm, (i.e. the clock difference between two nodes will be 60 ppm in worst case), in order to maintain synchronization, one of the two above actions need to occur at least every 43s. Leaving some margin, we consider 40s as the interval between two synchronization events. In the KA-based synchronization, a keep-alive frame and an ACK frame are both used, each containing 12 bytes. For ADV -based synchronization, only an Enhanced Beacon frame (EB) is needed. Usually, an EB contains time and channel information for synchronization, and initial link and slotframe information for new nodes to join the network. The length of the EB is therefore configurable (26 to 127 bytes), depending on the number of links and options encoded as Information Elements (IE) in the advertisement.

Fig. 6 shows the energy consumed on both the transmitter and receiver nodes by using either KA-based or ADV -based synchronization. Keep-alive packets have a fixed length, while the length of EB packets can vary according to the number of links being announced. We assume that the interval between two EBs (which is defined as a trade-off between energy consumption and network joining requirements) is longer than the maximum interval between two synchronization actions (i.e 40 seconds in our example). In order to minimize the energy

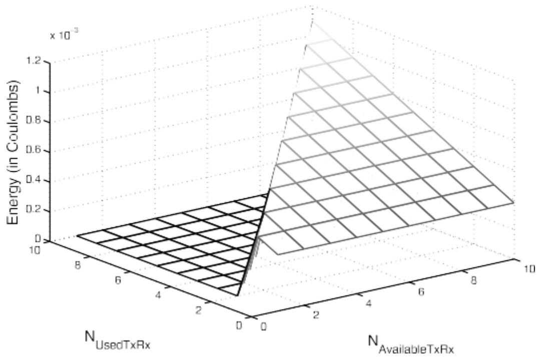


Fig. 7. Impact of overprovisioning on the transmitter side in a slotframe.

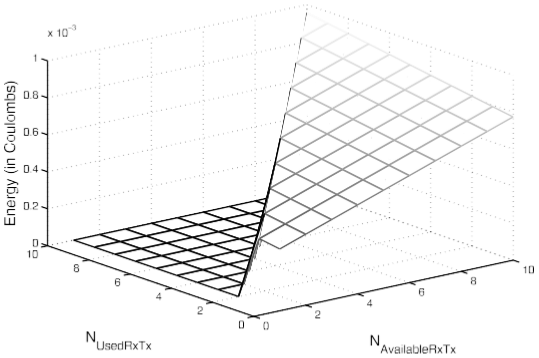


Fig. 8. Impact of overprovisioning on the receiver side in a slotframe.

consumed during synchronization, it is required that EBs have a length shorter than 57 bytes (this is particular to the GINA platform as other devices will show other energy consumption numbers). Therefore, in the case where *EB*s cannot be shorter, keep-alive based synchronization is preferable.

Besides the impact on energy consumption discussed above, other factors have to be taken into consideration while scheduling *EB* and *KA* messages. For example, *EB*-based synchronization may have a big advantage for a node which has many children; and *K A*-based synchronization can lead to power advantages, when using more advanced synchronization mechanisms such as adaptive frame-based synchronization as described by Stanislowski *et. al.* [21].

B. The Cost of Overprovisioning

Overprovisioning happens when the scheduling entity allocates more links to account for the losses due to the wireless transmission, in order to meet the QoS requirements. Overprovisioning consists in scheduling extra slots for possible re-transmissions. The lower the packet delivery ratio of a link between two neighbors, the more overprovisioned slots are needed. This is present in the equations in Section II-D. It is necessary to say that the queue length of node and the latency in a track are very strong functions of PDR and overprovisioning, as PDR is dependent on the environment, queue length and level of overprovision become the two main factors to vary in order to dimension the network. Overprovisioning impacts the energy consumption of the network while queue length impacts the memory requirements.

With regards to overprovisioning and from the transmitter's side, having *TxDataRxAck* or *TxData* slots in the schedule and not using them is equivalent to considering these slots to be sleep slots. When the packet delivery ratio of a link is better than expected, less re-transmissions happen, and some overprovisioned slots are unused. From the transmitter's point of view then, unused overprovisioned slots do not incur any extra energy consumption, since the radio stays off.

Fig. 7 presents the results of a simulation considering a slotframe with different available TxRx slots and different levels of usage of these slots. When a slot is used, the energy consumed by the slot frame is the same, regardless of the number of available TxRx slots, this is due to the fact that non-used TxRx slots are equivalent to sleep slots. Note also that the presented results are centered on the schedule configuration, i.e used slots cannot be more than available slots although logically the data-rate requirement in a node can be higher than the provided and in that case the consumption will be aligned with the maximum available data rate.

Overprovisioning impacts the energy consumption of the receiver side. That is, having *RxDataTxAck* or *RxData* slots on the schedule and not using them has an impact on the energy consumption of the network. The reason for that is that nodes in *RxDataTxAck* or *RxData* slot always listen whether a packet is received or not. In case of not getting a packet, the *RxDataTxAck* or *RxData* slot is equivalent to an idle listen slot because nodes only listen during the guard time period.

Fig. 8 presents the result of the simulation considering a slotframe with 10 configured RxTx slots and with different levels of usage. As it can be seen, for example for 1 used *RxDataTxAck* slot, the energy consumption increases depending on the number of available *RxDataTxAck* slots. The amount of that increment is in fact the difference of the energy consumption between an sleep slot and an idle listen slot.

IV. Conclusion

This article presents an energy consumption model for TSCH-based networks, and is applicable to industrial wireless standards such as WirelessHART, ISA100.11a and IEEE802.15.4e. This article highlights the contribution of each type of slot to the total energy consumed. The analytical model is verified by comparing the calculated energy consumption with experimental measurements on different hardware platforms. We use the derived equations to discuss different synchronization policies and the use of overprovisioning.

References

[1] L. Doherty, W. Lindsay, and J. Simon, "Channel-specific wireless sensor network path data," in *Proc. 16th IEEE ICCCN*, Aug. 2007, pp. 89–94.

[2] *WirelessHART Specification 75: TDMA Data-Link Layer*, HART, Oxford, U.K., 2008.

[3] *Wireless Systems for Industrial Automation: Process Control and Related Applications*, Standard ISA-100.11a-2011, May 2011.

[4] *IEEE Standard for Local and Metropolitan Area Networks—Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) Amendment 1: MAC Sublayer*, IEEE Standard 802.15.4e-2012, Apr. 2012.

[5] E. Toscano and L. L. Bello, "Multichannel superframe scheduling for IEEE802.15.4 industrial wireless sensor networks," *IEEE Trans. Ind. Informat.*, vol. 8, no. 2, pp. 337–350, Feb. 2012.

[6] N. Burri, P. V. Rickenbach, and R. Wattenhofer, "Dozer: Ultra-low power data gathering in sensor networks," in *Proc. 6th IPSN*, Apr. 2007, pp. 450–459.

[7] T. Watteyne, A. Mehta, and K. S. J. Pister, “Reliability through frequency diversity: Why channel hopping makes sense,” in *Proc. PE-WASUN*, Oct. 2009, pp. 116–123.

[8] T. Schmid, P. Dutta, and M. B. Srivastava, “High-resolution, low-power time synchronization an oxymoron no more,” in *Proc. 9th ACM/IEEE Int. Conf. Inf. Process. Sensor Netw.*, Apr. 2010, pp. 151–161.

[9] K. S. J. Pister and L. Doherty, “TSMP: Time synchronized mesh protocol,” in *Proc. IASTED Int. Symp. DSN*, Nov. 2008, pp. 1–8.

[10] M. Kohvakka, M. Kuorilehto, M. Hännikäinen, and T. D. Hämäläinen, “Performance analysis of IEEE 802.15.4 and ZigBee for large-scale wireless sensor network applications,” in *Proc. 3rd ACM Int. Workshop Perform. Eval. Wireless Ad Hoc, Sensor Ubiquitous Netw.*, Oct. 2006, pp. 48–57.

[11] S. Pollin, M. Ergen, S. Ergen, B. Bougard, L. Der Perre, I. Moerman, *et al.*, “Performance analysis of slotted carrier sense IEEE 802.15.4 medium access layer,” *IEEE Trans. Wireless Commun.*, vol. 7, no. 9, pp. 3359–3371, Sep. 2008.

[12] Q. Wang and W. Yang, “Energy consumption model for power management in wireless sensor networks,” in *Proc. 4th Annu. IEEE Commun. Soc. Conf. Sensor, Mesh Ad Hoc Commun. Netw.*, Jun. 2007, pp. 142–151.

[13] E. Casilari, J. Cano-Garcia, and G. Campos-Garrido, “Modeling of current consumption in 802.15.4/ZigBee sensor motes,” *Sensors*, vol. 10, no. 10, pp. 5443–5468, May 2010.

[14] O. Khader and A. Willig, “An energy consumption analysis of the Wireless HART TDMA protocol,” *Comput. Commun.*, vol. 36, no. 7, pp. 804–816, Apr. 2013.

[15] M. Palattella, N. Accettura, M. Dohler, L. Grieco, and G. Boggia, “Traffic aware scheduling algorithm for reliable low-power multi-hop IEEE 802.15.4e networks,” in *Proc. IEEE 23rd Int. Symp. PIMRC*, Sep. 2012, pp. 327–332.

[16] A. Morell, X. Vilajosana, J. L. Vicario, and T. Watteyne, “Label switching over IEEE802.15.4e networks,” *Trans. Emerging Telecommun. Tech.*, vol. 24, no. 5, pp. 458–475, Aug. 2013.

[17] N. Accettura, M. R. Palattella, G. Boggia, L. A. Grieco, and M. Dohler, “DeTAS: A decentralized traffic aware scheduling technique enabling IoT-compliant multi-hop low-power and lossy networks,” in *Proc. IEEE Workshop Int. Things, Smart Objects Services*, Jun. 2013, pp. 1–3.

[18] P. Thubert, T. Watteyne, M. R. Palattella, X. Vilajosana, and Q. Wang, “IETF 6TSCH: Combining IPv6 connectivity with industrial performance,” in *Proc. 7th Int. Conf. IMIS*, Jul. 2013, pp. 541–546.

[19] T. Watteyne, X. Vilajosana, B. Kerkez, F. Chraim, K. Weekly, Q. Wang, *et al.*, “OpenWSN: A standards-based low-power wireless development environment,” *Trans. Emerging Telecommun. Technol.*, vol. 23, no. 5, pp. 480–493, Aug. 2012.

[20] A. Mehta and K. Pister, “WARPWING: A complete open-source control platform for miniature robots,” in *Proc. IEEE/RSJ IROS*, Oct. 2010, pp. 5169–5174.

[21] D. Stanislowski, X. Vilajosana, Q. Wang, T. Watteyne, and K. Pister, “Adaptive synchronization in IEEE802.15.4e networks,” *IEEE Trans. Ind. Informat.*, vol. 9, no. 10, pp. 600–608, Jun. 2013.