Energy Evaluations for Wireless IPv6 Sensor Nodes

Cedric Chauvenet
Watteco Inc.
La garde, France
C.Chauvenet@watteco.com

Bernard Tourancheau
LIG, UMR 5217
Grenoble University, France
Bernard.Tourancheau@imag.fr

Denis Genon Catalot
LCIS, EA 3747
Grenoble University, France
dgc@iut-valence.fr

***Abstract*—Typical Wireless Sensor Networks (WSN) include some energy autonomous nodes and main powered router nodes. To meet a reasonable lifetime, i.e., years, the autonomous devices have to encompass strong energy constraints. Our target network architecture consists in such wireless autonomous nodes connected to a backbone based on Power Line Communication (PLC) nodes. Our given building automation application scenario runs for a 10 minutes periodic probes sampling, which reports a data frame over the Internet using IPv6. For this purpose, our WSN nodes leverage on open RPL/ContikiOS softwares and typical off the shelf electronic components. In order to optimize the node energy consumption we propose a systemic analysis that includes all the software layers and hardware components. The energy distribution among the different components and the critical software parameters are weighted against the global energy consumption. Thanks to measurements and technical data, we propose a simple model that allows to identify pitfalls and determine optimized solutions. Following our established guidelines, we believe our future WSN monitoring platform will achieve more than 10 years lifetime with the target scenario.**

Keywords*—*Wireless Sensor Networks (WSN), Energy Optimization, Energy Measurement, RPL, IPv6, PLC.

I. Introduction

WSN are called to new challenges given their energy constraints. To meet their power, price, size and deployment requirements, typical platforms fostered on a design based on a low power micro controller unit (MCU) and a low power RF transceiver. Adding a battery and probes to this communicating platform leads to a cheap autonomous WSN node. WSN's lifetime should be long enough, i.e., years, to provide a reasonable return on investment. Any costly maintenance like battery replacement annihilates the interest in WSN deployments. WSN devices will be sleeping most of the time and periodically wake up to perform their operations. This is controlled by their embedded operating software stack. Though carefully designed, these softwares should be carefully tuned with respect to the application requirements. In this work, we review the energy optimization process of a node that is compliant with the latest IETF recommendations for WSN. The target application is home and building automation.

Section II states the art of the domain and Section III presents the context of our work. In Sections IV and V the power consumption study of several MCU and transceivers are presented. Section VI shows the impact of the routing parameters. In Section VII, the global RF node energy spending is optimized. Sections VIII and IX present software optimizations for the MCU and the transceiver. Section X shows the probes consumptions that will be added to the node.

II. Related Work

Energy consumption is the most important criterion for the development of autonomous sensor network nodes. Especially when they target several decades or perpetually powered systems. As mentioned in [1], battery replacement is not an option for networks with thousands of physically embedded nodes and the paper list various and commonly used techniques to save power such as power-aware computing, energy-aware software or power management of radios.

Energy has been considered since the very beginning of WSN developments, like with the WeC node designed in 1998 [2] that concretizes the idea of communicating sensors. Then, some similar platforms have been designed, with more powerful MCU and the same radio chip until 2001, where a new generation of nodes emerged with the Mica [3], designed in Berkeley in 2002. This node was carefully designed to serve as a general purpose platform for WSN research. Work in [3] highlights the need for node sleeping most of the time with periodic wake up, and shows that a total improvement factor of 11 can be reached with this technique. From this date on, all the sensor network node designs used this duty cycling techniques to save power. Leveraging on experimental feedback, Mica2 corrected many shortcomings: the boost converter was discarded, new components were chosen. But, this new design leads to higher power consumption. Continuing the evolution, MicaZ replaced the CC1000 radio with a CC2420, an IEEE 802.15.4 2.4GHz compatible radio with 250 kbps throughput capability, and evolved O-QPSK modulation. This transceiver embedded a part of the 802.15.4 standard, limiting the software processing needed to send and receive packets by DSSS signal treatment, thus saving energy.

A major step has been reached with the design of the Telos platform [4] released in 2004. It enables experimentations with minimal power consumption, a maximum of 10 kB RAM memory for managing increasing stack fonctionnalities and increased software and hardware robustness. Telos is based on a MSP430 MCU with a CC2420 radio. The MSP430 offered significantly lower consumption, reducing the total active power down to 41 mW, with the same transmitting power consumption as MicaZ. According to [4], the power consumption of the MSP430 is 20 times lower than the MicaZ in sleep mode, and 4 times lower in active mode.

However, even if hardware greatly improved over time, software requirements changed, especially for the communication stack. The IETF proposed the adoption of IPv6 over WSN with an adaptation layer called 6LoWPAN, RFC 6282 [5] and a routing proposal named RPL, RFC 6550 [6]. 6LoWPAN offers header compression to save bytes and allows frame fragmentation to resolve MTU issues. The Routing Protocol for

ISBN: 978-1-61208-296-7

Low power and Lossy Networks, RPL, is a proactive distance vector protocol that creates Destination Oriented Directed Acyclic Graphs (DODAG). Several control messages protocols are defined in the RPL's RFC. Namely, DAG information object, DIO, and information solicitation, DIS, propagates and requests DODAG informations. Destination advertisement object, DAO, allows routers to update their tables. Also, a Neighbor Discovery mechanism, RFC 4861 [7], ensures neighbors' reachability within the topology by neighbor solicitation and advertisement, NS/NA, exchanges. Notice that node energy can be used in the metric governing this routing topology construction.

This software stack complexity impacts the key parameters governing power requirement. Work in [8] evaluates the performance of RPL and 6LoWPAN using the TinyOS stack. They showed that RPL parameters' values and the number of downwards stored routes affect the power consumption. Also they proposed an optimization for fragmented packets where only the header part is decompressed for routing decisions. [9] proposed a power consumption model that deeply describes the wireless communication. Hence, this work shows that the number of hops should be reduced to the minimum in order to saved power. To our knowledge, the impact of the RPL Contiki stack on energy consumption was not studied.

We address the power consumption issues by conducting real hardware measurements and analysis of the software impact on consumption. As suggested in [1], we split the system into several parts and conduct an analysis for each of them in the first part of the paper. However, the system should also be considered as a whole in order to insure a good integration between hardware and software components. This is the aim of the second part of our paper where we optimize the software behavior against energy.

III. Platform settings

This paper relates our research in order to define an operational and efficient WSN platform using Power Line Communication (PLC) [10], and Radio Frequency (RF) communication transceivers. These architectures gathers autonomous battery powered RF nodes, main powered PLC or RF nodes and PLC-RF routers. The PLC-RF routers serve as a backbone for RF nodes. Such an architecture restricts the routing capabilities to mains powered devices. Moreover, battery operated devices do not waste energy. The gain of these energy efficient hybrid architectures is described in [11]. At the network level, our nodes are seamlessly interconnected using the Internet IPv6 protocols [12].

In order to operate the nodes for this study, we used ContikiOS and tools [13], which provide an open source micro operating system for constrained devices.

The RF transceivers follows the IEEE 802.15.4 standard [14], which was designed for low power, short range and low throughput networks. Initially released in 2003, this standard offers a maximum throughput of 250 kbps. New revisions of this standard added other modulations scheme and a sub-GHz RF band for less attenuation transmission with lower throughput, typically 20k to 50Kb/s, to achieve a longer range and more robust communications.

The target MAC layer relies on the CSMA/CA mechanisms required by the 802.15.4 standard. Also, as in most of WSN protocol stacks, a radio duty cycling (RDC) protocol is added in order to save a significant amount of power. We use an adapted version of contikiMAC RDC [15] without the preamble sampling protocol. This protocol induces periodic radio wake up to sense the radio activity at a fairly high frequency, 4 Hz by default. This protocol may sends packet copy until the recipient acknowledge it. Notice that if the traffic required by the application is lower than 4 Hz, the periodic wake up will induces non necessary overhead. Our target temperature monitoring application typically requires one packet per node every 10 minutes send to the sink.

On top of an IPv6 communication stack, our target application uses the IPSO application framework [16] profiles designed primarily for smart energy applications. These profiles relies on the Constrained Application Protocol, CoAP [17], that enables sensing nodes to be interrogated through RESTfull primitives.

IV. MCU Power Consumption

In a node, the MCU wakes up periodically to check if an event is detected by the probe or the RF transceiver, and executes the corresponding software tasks. Otherwise, it stays in sleep mode. Table I compares the current drawn for 3 MCUs in different states. The results shows the great gap between active and low power modes. For instance, the MSP430f5438A running at 20 MHz can, in theory, achieve a 5 years lifetime over an 1Ah battery only if it stays in sleep mode more than 99,6% of the time. The MCU frequency impacts its energy consumption. In order to quantify this, we measured the voltage on a 10,1 Ω load added between a regulated DC power source of 3 V and the target architecture. In the experiments, this voltage reflects the current drawn by the MCU plus the ATRF212 transceiver in its standby mode, which is 0.2 μA in [21]. We ran the complete software stack over different frequencies without any RF communication nor connected probe. Figure 1 shows the corresponding oscilloscope traces, averaged over 128 wake up samples to include the variety of wake up profiles. We calculate the energy, E by integrating the power over time during the period of visualization following (1).

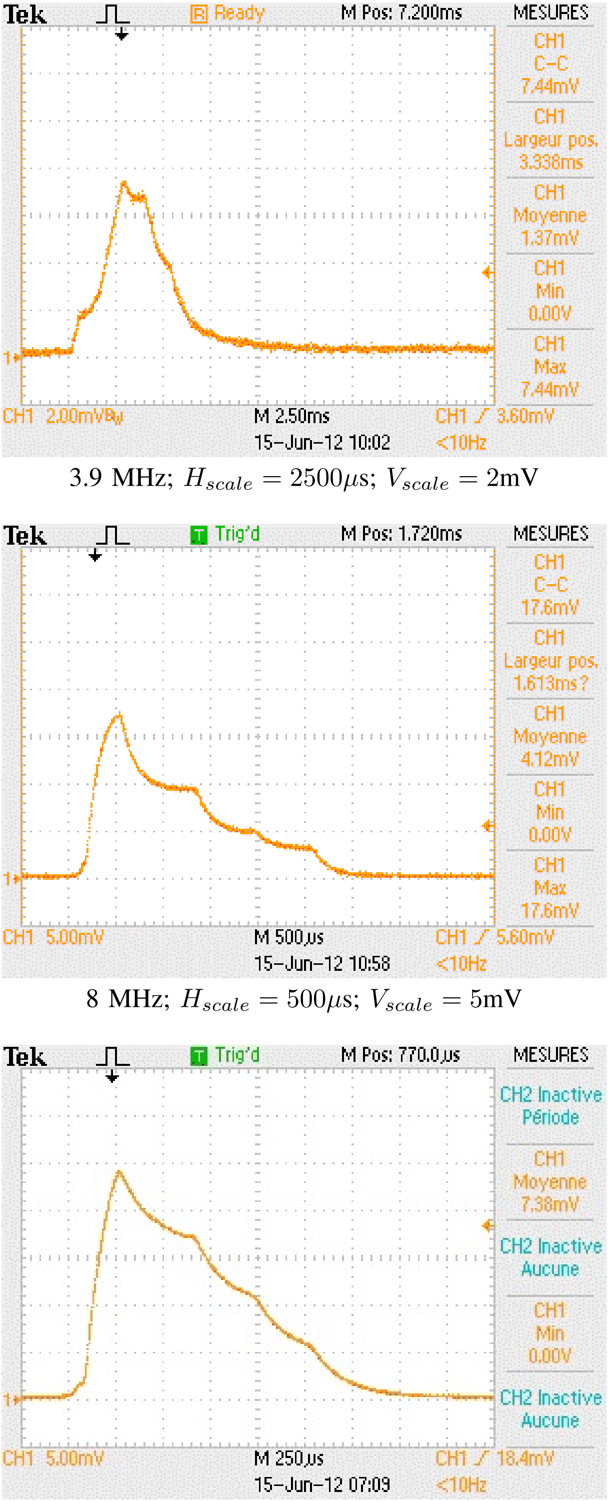
TABLE I. Micro controllers drawn current under 3V at different modes and frequencies. Experimental measurements are in bold.

Mode	MHz	MSP430 f1611 (mA)	MSP430 f5438A (mA)	SIM3C1xx (mA)
Active	4	2 [18]	**1.51**	
Active	8	4 [18]	**1.84**	
Active	16		**5.21**	
Active	20		6.37 [19]	7.8 [20]
Active	80			22 [20]
Low P	0.032	0.002 [18]	0.0021 [19]	0.0008 [20]
Sleep		0.002 [18]	**0.0012**	0.000145 [20]

$$E(J) = \frac{3}{10.1} \times \int_0^{t_{visualization}} V_{average} \qquad (1)$$

Table II summarized the corresponding computed energy.

As expected, the maximum current drawn increases with the MSP frequency while the rise up and drop time decrease,



3.9 MHz; $H_{scale} = 2500\mu$s; $V_{scale} = 2$mV

8 MHz; $H_{scale} = 500\mu$s; $V_{scale} = 5$mV

16 MHz; $H_{scale} = 250\mu$s; V_{scale} =5mV

Fig. 1. Current profile of MSP430 average wake up at different operation frequency.

TABLE II. MEASURED ENERGY PARAMETERS OF MSP430 WAKE UP PERIOD AT DIFFERENT FREQUENCIES.

Frequency of the MSP (MHz)	3.9	8	16
Rise Up Time (ms)	2.5	0.5	0.25
Maximum current (mA)	0.74	1.74	2.38
Drop Time (ms)	7.5	2.5	1.5
Time of Observation (ms)	25	5	2.5
Average Voltage (V)	1.37	4.12	7.83
Energy (μJ)	**10.2**	**6.12**	**5.48**

although none of these happens in a linear fashion. But we notice a higher gap between 3.9 and 8 MHz than between 8 and 16 MHz for these three parameters. Regarding the maximum current, there is a 1 mA gap between 3.9 and 8 MHz, and a 0.64 mA gap only between 8 MHz and 16 MHz. Notice that our measurements differ from the data sheet maximum current consumption. This can be explained for 2 reasons. Firstly, they are averages over 128 samples. Secondly, the wake up duration is not long enough to reach the maximum current consumption. The rise up and drop time are divided by a factor of 1.7 and 2 when going from 8 to 16 MHz while there are respective factors of 5 and 3 when moving from 4 to 8 MHz. The erratic performances of the MCU, while down clocked at 4 MHz, may explain this non linear behavior. Taking into account these results, the SIMC3C1 seems promising for our future platform.

V. TRANSCEIVER POWER CONSUMPTION

With our architecture, the RF nodes do not route messages. An important parameter is thus the transmission range to ensure that every RF node will reach a main powered router. The studied transceivers follow the IEEE 802.15.4 standard, which proposes several frequency bands. We focused on the 868 MHz and 2.4 Ghz ones. In order to quantify the theoretical advantage of each transceiver for our application requirements, we computed the power needed to communicate within a range d. The Friis formula states this received power as a function of the distance and power gain.

$$P_{received} = 22dB + 20 * log(d/\lambda) \qquad (2)$$

Table III presents the results. The ATRF230 has a low instantaneous power consumption and a high throughput, confirmed by its "Energy at max-range" number. However, its coverage range is small compare to the others. The ATRF212 and the CC1120 transceivers have fairly similar "Energy at max-range" consumptions, because the higher power consumption of the latter in balanced by its higher throughput capability. Though, the 2.4GHz band offers a shorter range than the two others and the maximum power of transmission authorized in this band is lower. As a result, for a RF node in our architecture, a router may be out of range in 2.4 GHz while connected in 868 MHz.

Table III also shows that the computed maximum reachable distance over 868 MHz is 27 times greater than over 2.4 GHz using the transceivers and regulation considered. The energy cost per bit in much lower is 2.4 GHz for distances shorter than the 1.5km range. Also, for the same transmitted data, the channel occupation is longer in 868 MHz and the interference range is larger. This may become an issue in dense environments. Beside energy considerations, the 868MHz band may increase the path diversity while being able to connect to

ISBN: 978-1-61208-296-7

TABLE III. CHARACTERISTICS OF RADIO TRANSCEIVERS AND CORRESPONDING ENERGY FOR DIFFERENT TRANSCEIVERS.

Module	Rate (kbps)	Freq (MHz)	Sensib (dBm)	Current Tx (mA)	Current Rx (mA)	Max Power (dBm)	Max Range (km)	Power Recv (dBm)	Energy at Max Range (μJ/bit)	Energy at 1.5 km (μJ/bit(dBm))
ATRF212	20	868	-110	25	9.2	10	25	-109.19	5.13	2.96 (-15)
CC1120	50	868	-110	45	22	14	40	-109.27	4.02	2.52 (-15)
SI4464	40	868	-110	37	13	14	40	-109.27	3.75	2.33 (10)
SI4461	40	868	-110	33	13	14	40	-109.27	3.45	2.33(10)
ATRF230	250	2400	-101	16.5	15.5	3	1.5	-100.58	0.38	0.38(3)

several routers as this allows for multi and backup routes. With our architecture and target application, we decided to use the 868 MHz band and thus the SI4461 transceiver seems a good candidate. Of course, additional parameters could be added to this model and in particular, the gain and type of antennas that could greatly impact the transmission range.

VI. ROUTING CONTROL SOFTWARE IMPACT

The literature exhibits the high power cost of RF transmissions [22], and many mechanisms have been investigated to reduce this energy budget. Some of them rely on frame size reduction such as compression [23] or data aggregation [24]. A complementary solution is to limit the number of control packets sent. For instance, RPL uses the trickle algorithm; see [25]. Several other parameters may influence the transceiver consumption, for instance throughput and packet delivery rate have direct impact on the energy consumption, as they define the duration of each transmissions and the average number of retransmission.

In our target home or building automation use case, RF nodes running over battery are periodically reporting values sensed by the probe, and downward traffic is restricted to node configuration and is not periodic. This results in highly asymmetric traffic mostly from sensors to the border router.

The RPL routing control messages practical periodicity is described in Table IV. The RF nodes mainly interact at initialization in the proactive RPL strategy. This RF node control traffic is independent from the topology. For multicast packets, a RF mode gets back to sleep just after sending whereas it waits for the acknowledgment for unicast packets. There is one exception for multicast DIS packets, where we forced the RF node's transceiver in reception mode during a duration of, by default, two seconds. This ensures that the node gets all DIO from all its potential parents. This supplementary energy cost needed to keep the transceiver active during this period is balanced by the resulting efficiency of the parent selection, which can last up to the network lifetime. Moreover, if we limit the parent selection only to the first DIO received, there is a risk that the parent communicates through a lossy link or has a high RPL rank. All this resulting in suboptimal paths, transmission retries and finally higher power consumption.

We studied our RPL's scenario traffic in order to precisely determine the communication activity. Table V shows the size and number of packets exchanges running the target monitoring application over a 24h period. The majority of the traffic is concentrated in NS/NA exchanges and data reports. The overhead induced by the routing control messages is relatively very low. Notice that NS/NA exchanges are required by our active data request mechanism, which inserts a data pending flag in the NA packets. This flag advertises that a packet intended to the RF mode is buffered by its parent. When receiving this flagged NA, the RF mode keeps its radio active until the reception of all its data packets. Moreover, these NS/NA exchanges enable a periodical check of the bidirectional connection with the selected parent. The expected transmission count, ETX, metric update accordingly to NS/NA exchanges successes and MAC retries.

TABLE IV. RPL MESSAGES PERIODICITY.

Type	Origin	Message Periodicity	Comments
NS	Router	Best parent 60s; Other p. 120s ; Other neighbor 360s	No NS to RF
NS	RF	Best p.: 3 to 600s	Trickle timer
NA	Router	Same as NS	
NA	RF	None	No NS to RF
DIO	Router	Imin 1s ; Imax 1050s	Trickle
DIO	RF	Not regular	Once attached inform neighbors about rank
DATA	Router, RF	600s	Application dependent
DAO	Router	DTSN increment 360s	For each DIO with incremented DTSN and if parent switching
DAO	RF	Lower period between DIS and DTSN increment	Each time DIO with incremented DTSN & if parent switching
DAO ACK	Router RF	Same as DAO	When correctly received
DIS	Router	Not regular	Only when node needs infos to be attached to DAG
DIS	RF	360 s	To request DIO from parent and update DAG infos
ACK	Router, RF	Same as unicast	For each unicast frame sent

TABLE V. SIZE AND NUMBER OF MESSAGES EXCHANGED BY A RF NODE OVER A 24H PERIOD.

Mesg type	Size (B)	Sent	Recv	Comments
NS	72	151	0	
NA	72	0	151	
DIO	111	0	24	ETX metric, DODAG conf, prefix Info
DATA	93	144	0	UDP 10 bytes payload
DAO	66	24	0	target, transit Info
DAO ACK	40	0	24	requested for each DAO
DIS	33	24	0	
ACK	5	199	344	
Mesg #		543	543	
Size (B)		28 892	17 128	

VII. RF NODE POWER CONSUMPTION

Unlike generic power consumption models like in [26], we built our simple energy model including our application. In our use case, a RF node executes periodic tasks: sleeping, waking up, running, sensing, transmitting and receiving. We computed the expected energy to operate the node according to the complete scenario for different hardware components and software parameters. The sleep mode current is integrated over the time of the simulation minus the active period of time. Throughputs are the same as in Table III, and the number of packet exchanges is deducted from the message periodicity presented in Table IV. We then computed the total cost over a day according to a 4 Hz wake up frequency. We consider a RF node without any probe. Probes impact is reported independently in Section X. To complete the energy model of the radio, we add the energy spent when the transceiver stays in RX mode, waiting for acknowledgments. Notice that our simple model considers that all packets are successfully received and thus there is no retry at the application or MAC layer. Table I gives the results for our reference platform with MSP430 at 16MHz and ATRF212 against the best up-to-date components.

We first investigate the power consumption of the low dropout voltage regulator, LDO, computed from its yield characteristics, with an input of 1.5V provided by a single AAA battery and the desired output voltage of 3V.

TABLE VI. ENERGY REPARTITION OVER A DAY FOR DIFFERENT RF NODES.

Node	MSP430+ATRF212				SIM3C1+SI4461	
	With LDO		Without		Without	
	(J)	%	(J)	%	(J)	%
LDO	18.38	82	-	-	-	-
Radio Tx	0.59	3	0.59	14	0.57	34
Radio Rx	0.20	1	0.20	5	0.14	8
CCA - Backoff - Wait for Ack	0.03	<1	0.03	1	0.04	2
MCU Wake Up	2.09	9	2.09	51	0.76	45
MCU + RF Sleep	1.22	5	1.22	29	0.19	11
Total	22.51	100	4.13	100	1.71	100
Expected Lifetime on battery (Yrs)	1.31		7.16		17.24	

Table VI shows that the LDO consumes 82% of the overall energy, which is an issue. The current drawn by the node is very low, between 10 and 20 μA. This falls in the range where the LDO efficiency is the worst, between 0,5 and 0,6, leading to a huge energy waste. In order to achieve a more efficient design, this element must be removed. A battery with the required voltage, typically 3V, is directly connected to the node. Moreover, as stated in [3], [4], the battery voltage cutoff should also correspond to the node components cutoff. The voltage regulation is not mandatory if all components used in a node can work over a voltage range matching the battery's one. Although, attention should be paid to the probe precision's behavior and to the clock drift against the voltage variation. In our reference platform, we selected a 3V battery with a capacity of 1000 mAh, a self discharge current below 1 % per year, and a dropout voltage of 2.0 V. We took into account the 1% self discharge of the battery in the model by subtracting 1% of the remaining energy at the beginning of each year.

The MSP430 requires a minimum voltage of 2.2V to run at 16 MHz. However, we observed that it is able to run correctly with a voltage supply as low as 1.8 V. Hence the entire energy from the battery will be used.

After removing the LDO on our reference platform, the major part of the power consumption is due to MSP430 wake up (51%). The sleep mode of the RF transceiver and the MSP430 represents the second bigger power consumption (29%). This is an interesting result, because most of the literature about energy consumption in WSN considers the RF transceiver as the biggest energy consumer [22]. This focus is only correct when looking at instantaneous power consumptions. However, when integrating over a long time, our results show that the transceiver is not the main consumer with a rather low periodic traffic. The expected lifetime with a 1000 mAh battery reaches 7.16 years. This matches the smart energy applications lifetime targeted by these nodes. The new ARM architecture seems very promising because it reduces the leakage current and wake up costs. This potentially increases the lifetime up to 17 years.

VIII. MCU SOFTWARE OPTIMISATION

Results from Table VI show that the most important gain may result from reducing the MCU consumption in wake up and sleep mode. While the sleep mode consumption depends on the hardware, we tried to reduce the cost of each wake up and to decrease the wake up frequency.

During each wake up, ContikiOS runs several periodic processes governed by timers that are not mandatory in our application scenario. For instance, we disabled several timers related to routing maintenance in RF nodes. We also increased several timers, such as neighbor checking, neighbor unreachability detection (NUD), route lifetime checking and I/O checking, because our application do not require fine grain timing constraints. We eventually adapted some RPL timers to the RF node mode of operation, such as the DIO and DIS timers management, and timers that governs the sleeping mode. In our building automation scenario, in order to keep an acceptable reactivity, a wake up frequency of 1 Hz is a good tradeoff in practice. We also activate the fast wake up function of the MSP that decrease the wake up time from 150 μs to only 5 μs. The corresponding expected lifetime increases from 7.16 years to 12.55 years.

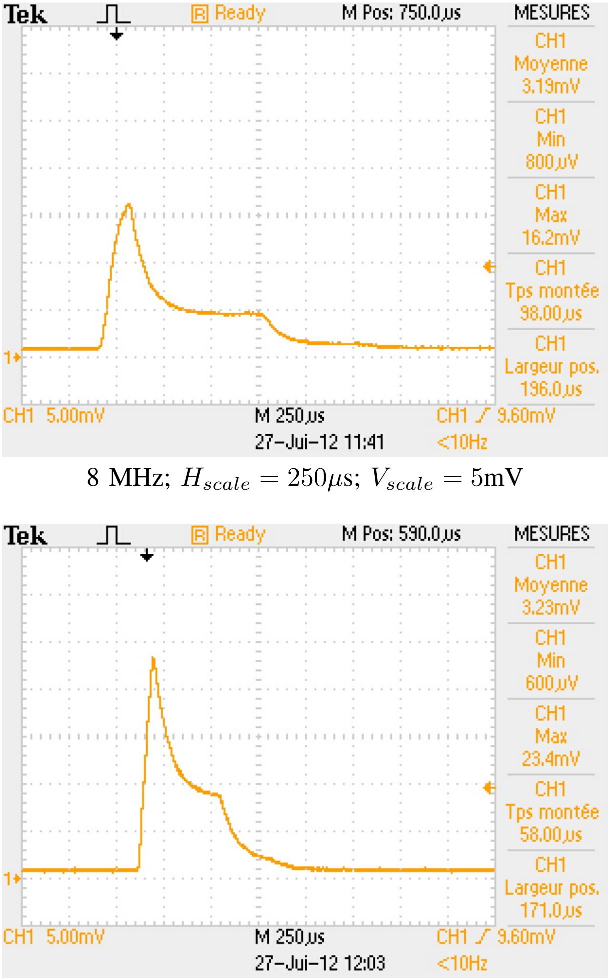
The biggest part of the energy consumption is now due to the sleep mode of the components that represents 52% of the overall energy budget while the MCU wake up is only 13%.

When measuring experimentally the power consumption of the MSP430 in sleep mode, we found 4.3 μA. This is far above the expected value of 1.2 μA mentioned in the data sheet when using LPM3 mode because we were using the internal oscillator (REFO). According to the data sheet this oscillator consume 3 μA. The intend of this oscillator is to provide a precise clock at 32,768 KHz. In our case, we do not need such a precision. We eventually use the internal low power oscillator (VLO) and the power consumption matched the expected value of 1,2 μA under 3 Volts. With this configuration the average current consumption of the platform drops to 5.9 μA, giving a computed expected lifetime of 20.23 years.

IX. COMMUNICATION SOFTWARE OPTIMIZATION

At this point, our study shows that the next optimization should focus on radio transmissions that now represents 39 % of the total energy budget.

First of all, we determined the impact of lowering the transmission power according to the transmission distance.



8 MHz; $H_{scale} = 250\mu$s; $V_{scale} = 5$mV

16 MHz; $H_{scale} = 250\mu$s; $V_{scale} = 5$mV

Fig. 2. Current average profile of MSP430 wake up at different frequency.

This is presented in Table III where we adjusted the 868MHz transceiver's power in order to compare with the maximum range of the 2.4GHz transceiver. The resulting energy gain is small and the 2.4GHz transceiver stays 2 order of magnitude better. Also, with a transceiver at maxim gain, the transmission reaches the wider possible range with the best PDR. The only drawback concerns the interferences but with our seldom communication application and a good MAC layer, this is not really an issue.

Hence, we focus on the limitation of the number of messages. When parsing the radio activity presented in Table V, it appears that 49.7 % of them are NS/NA, 13,6 % are RPL messages, and the remaining 36,7 % are data reports. Reducing the number of data messages is application dependent and thus not in the scope of our study. We study how to reduce RPL and NS/NA messages. Initially, NS/NA exchange happened every 10 minutes between a RF node and its parent. We augmented this interval, at the price of a greater latency for downward message transmissions, and more sparse connectivity checking. Though, notice that if we have a data reporting every 10 minutes, the link is regularly checked during these transmissions, and thus we may remove NS/NA exchanges, with some modifications of the neighbor discovery mechanism implemented in ContikiOS. In order to reuse our downward frame exchange mechanism, the data pending flag would also need to be put in the data frame's acknowledgments. Thus, data messages would also update the ETX metric and check the upward link, limiting the overhead to RPL messages only. Such an optimization of the NS/NA exchanges decreases the average node consumption leading to an expected lifetime of 28.5 years.

X. Probes Power Consumption

Embedded probes need some current for their functioning. Table VII presents the power consumption for classical WSN building monitoring probes running under 3V.

TABLE VII. Various probes energy and current under 3V.

Type	Vendor	Current (μA)		Duration	Energy
		active	standby	(s)	(J/yr)
Temp	TI TMP112	7	0.5	0.035	47

Temp
Humid
	Sensirion
	SHT21
		300	0.15	0.114(HP)
				0.015(LP)
					20
					15

PIR	PANASONIC EKMB1103112	1,9	1	2	95
DoorOpen	Meder KSK-1A66-1015	0.0006		NA	<1
Light	TAOS TSL2561T	240	3,2	0.25	312
CO2	AlphaSense IRC-A1	> 20000		0.46	>>1000

Embedding a CO2 probe, such as the one in Table VII, on a battery powered node is not wise since it depletes a 1000 mAh battery in only two days. On the contrary, the door/window opening probe is consuming very few energy, because its ILS bulb requires no power to operate. For other probes, there is great difference between the active and the standby consumption, hence duty cycling set up is necessary. The PIR probe cannot be fully optimized, because it automatically wakes up when it detects something, and goes back to sleep after 2 seconds. With the luminosity probe, the MCU needs to trigger a sensing window and retrieve the value acquired. As a result, the MCU can return in sleep mode during the sensing, and retrieve the luminosity value during the next wake up. The window size can be optimized as well as the periodicity of the measures. In our scenario, the sensing window period lasts 250 ms and the MCU retrieves the value during its next wake up. The temperature and temperature/humidity probes need some processing from the MCU when providing these values, increasing substantially the overall cost of the probe sensing. However, the periodicity of sensing can be large, because humidity and temperature have a low dynamic. For the SHT21, the time needed to do the measurements depends on the precision. For a full precision, 14 bits temperature, it needs a maximum time of 85 ms and for 12 bits humidity value, 29 ms. Overall, the CPU needs to be active during 114 ms. This time drops to only 15 ms for 11 bit precision temperature and 8 bit precision humidity. The TMP112 probe has a default precision of 12 bit, and an extended mode that can be activated to measure temperatures greater than +128 $^\circ$C that far exceed our requirements. According to [27], the maximum conversion time for a 12 bit temperature value is 35 ms.

Table VIII summarizes the power consumption of the probes depending on their configuration. We computed their relative part in the total energy budget, considering the node at different improvements levels following the preceding Sections. This shows that embedding a probe in a node impacts

TABLE VIII. POWER CONSUMPTION OF PROBES ACCORDING TO THE NODE SOFTWARE'S OPTIMIZATION (V1 SECTION VIII, V2 SECTION IX)

Probe	Precision / Period	lifetime v2 (yrs)	Energy (%)	lifetime v2 (yrs)	Energy (%)
Temp	12 bit / 10 s	4.69	77	5.03	82
Temp	12 bit / 60 s	12.43	39	15.13	47
TempHumid	Full / 10 s	1.64	90	1.73	94
TempHumid	Full / 60 s	6.43	60	7.81	73
TempHumid	Min / 10 s	7.34	54	9.20	68
TempHumid	Min / 60 s	13,18	18	20.61	28
PIR	/ 60 s	17.03	16	22.54	21
DoorOpen	/ 1 for 10 μs	20.23	< 1	28.50	< 1
Light	/ 10s 250ms	7.80	61	8.80	69
Light	/ 6 s 250ms	11.61	43	13.97	51

significantly the average power consumption. For instance, using the full precision mode of the SHT21 reduces by an order of magnitude the expected lifetime of the node, as compared to the minimum precision mode.

XI. CONCLUSION AND FUTURE WORK

In this paper, we conducted an extensive power consumption study for the design of a wireless sensor network platform. We assessed the energy consumption for the different elements of the node based on real power consumption measurements and data sheet numbers. We set up a simple consumption model in order to discuss the power optimization of the MCU, the radio transceiver and a range of probes. Moreover, we introduces software optimizations, related to our application scenario, in ContikiOS and the RPL network stack.

We pointed out the key parameters that govern the energy consumption. We implemented all the energy improvements for the selected components in order to estimate the best design for a WSN node. Our results estimate a more than 20 years RF node lifetime with a data reporting interval of 10 minutes. This encompasses our aims with a probe added within the power budget for a 10 years lifetime.

We plan to further power our nodes with an energy harvesting system such as a solar PV panel and a super capacitor. In the home or building automation applications such an autonomous sensing node is our next target design. Given the average power consumption of our actual nodes below 17μW, we are confident in the realization of such a design.

ACKNOWLEDGMENT

The authors would like to thank Mathieu Pouillot and Pierre-Emmanuel Goudet for their valuable help and support.

REFERENCES

[1] V. Raghunathan, C. Schurgers, S. Park, and M. Srivastava, "Energy-aware wireless microsensor networks," *Signal Processing Magazine, IEEE*, vol. 19, no. 2, pp. 40–50, 2002.

[2] J. McLurkin, "Algorithms for distributed sensor networks," Ph.D. dissertation, Department of Electrical Engineering and Computer Sciences, University of California, 1999.

[3] J. Hill and D. Culler, "Mica: A wireless platform for deeply embedded networks," *Micro, IEEE*, vol. 22, no. 6, pp. 12–24, 2002.

[4] J. Polastre, R. Szewczyk, and D. Culler, "Telos: enabling ultra-low power wireless research," in *IPSN*. IEEE, 2005, pp. 364–369.

[5] N. Kushalnagar, G. Montenegro, C. Schumacher, and A. Danfoss, "Ipv6 over low-power wireless personal area networks (6lowpans): Overview, assumptions, problem statement, and goals," RFC 4919, August 2007.

[6] A. Brandt, J. Buron, and G. Porcu, "Home automation routing requirements in low-power and lossy networks," RFC 5826, April 2010.

[7] T. Narten, E. Nordmark, W. Simpson, and H. Soliman, "Neighbor discovery for ip version 6 (ipv6)," RFC 4861, September 2007.

[8] J. Ko, S. Dawson-Haggerty, O. Gnawali, D. Culler, and A. Terzis, "Evaluating the performance of rpl and 6lowpan in tinyos," in *IPSN*. ACM, 2011.

[9] Q. Wang, M. Hempstead, and W. Yang, "A realistic power consumption model for wireless sensor network devices," in *SECON*. IEEE, 2006, pp. 286 – 295.

[10] C. Chauvenet, B. Tourancheau, D. Genon-Catalot, P.-E. Goudet, and M. Pouillot, "A communication stack over plc for multi physical layer ipv6 networking," in *Smart Grid Communications (SmartGridComm), 2010 First IEEE International Conference on*, 2010, pp. 250–255.

[11] L. B. Saad, C. Chauvenet, and B. Tourancheau, "Ipv6 (internet protocol version 6) heterogeneous networking infrastructure for energy efficient building," *Energy*, vol. 44, no. 1, pp. 447 – 457, 2012.

[12] C. Chauvenet, B. Tourancheau, D. Genon-Catalot, P.-E. Goudet, and M. Pouillot, "Interoperable ipv6 sensor networking over plc and rf media," *International Journal of Business Data Communications and Networking (IJBDCN)*, vol. 6, no. 4, pp. 1–20, 2010.

[13] A. Dunkels, B. Gronvall, and T. Voigt, "Contiki-a lightweight and flexible operating system for tiny networked sensors," in *29th Annual IEEE International Conference on Local Computer Networks*. IEEE, 2004, pp. 455–462.

[14] IEEE, "Ieee standard for local and metropolitan area networks–part 15.4: Low-rate wireless personal area networks (lr-wpans)," 2011.

[15] A. Dunkels, "The ContikiMAC Radio Duty Cycling Protocol," Swedish Institute of Computer Science, Tech. Rep. T2011:13, Dec. 2011.

[16] Z. Shelby and C. Chauvenet, "The ipso application framework," August 2012 (last checked June 2013). [Online]. Available: http://www.ipso-alliance.org/wp-content/media/draft-ipso-app-framework-04.pdf

[17] Z. Shelby, K. Hartke, C. Bornmann, and B. Franck, "Constrained application protocol (coap)," IETF draft, 2012.

[18] Texas Instruments, "Msp4301611 datasheet," 2010 (last checked June 2013). [Online]. Available: http://www.ti.com/lit/ds/symlink/msp430f1611.pdf

[19] ——, "Msp430f5438a datasheet," 2010 (last checked June 2013). [Online]. Available: http://www.ti.com/lit/ds/symlink/msp430f5438a.pdf

[20] Scilabs, "Cortexm3 datasheet," 2012 (last checked June 2013). [Online]. Available: http://www.silabs.com/products/mcu/mixed-signalmcu/Pages/SiM3C1xx.aspx

[21] Atmega, "Atrf212 datasheet," 2012 (last checked June 2013). [Online]. Available: http://www.atmel.com/Images/doc8168.pdf

[22] J.-P. Vasseur and A. Dunkels, *Interconnecting smart objects with ip: The next internet*. Morgan Kaufmann, 2010.

[23] C. Bornmann, "6lowpan generic compression of headers and header-like payloads," IETF draft, 2011.

[24] S. Cui, A. J. Goldsmith, and A. Bahai, "Energy-efficiency of mimo and cooperative mimo techniques in sensor networks," *Selected Areas in Communications, IEEE Journal on*, vol. 22, no. 6, pp. 1089–1098, 2004.

[25] P. A. Levis, N. Patel, D. Culler, and S. Shenker, "Trickle: A self regulating algorithm for code propagation and maintenance in wireless sensor networks," Computer Science Division, University of California, Tech. Rep., 2003.

[26] Q. Wang and W. Yang, "Energy consumption model for power management in wireless sensor networks," in *Sensor, Mesh and Ad Hoc Communications and Networks, 2007. SECON'07. 4th Annual IEEE Communications Society Conference on*. IEEE, 2007, pp. 142–151.

[27] Texas Instruments, "Tmp112 datasheet," 2009 (last checked June 2013). [Online]. Available: http://www.ti.com/lit/ds/symlink/tmp112.pdf