Wireless Sensing: What Simplicity has to Offer?

Asanka Sayakkara[*], Chathura Suduwella[†], Charith Shalitha[‡],
Rumesh Hapuarachchi[§], Chamath Keppitiyagama[¶] and Kasun De Zoysa[‖]
{asa[*],cps[†],chamath[¶],kasun[‖]}@ucsc.cmb.ac.lk
{acsdealwis[‡]}@gmail.com, {rehrumesh[§]}@hotmail.com
University of Colombo School of Computing,
Sri Lanka.

***Abstract*—We argue that, a wireless sensor does not have to implement a full networking stack in order to be a part of an energy efficient sensing application as long as it can deliver its sensed data to an Internet connected device over a single hop. Current hardware industry trends indicate the possibility of implementing wireless sensors for a low cost with simple capabilities. This poster highlights our preliminary work aligned with this trend and aims to open a discussion on this topic.**

I. Introduction

Wireless sensor networks (WSN) consist of small low-cost battery powered devices called *sensor motes* deployed in various domains such as precision agriculture, wildlife monitoring and many more. A sensor mote contains few on-board sensors, a low powered MCU and an 802.15.4 wireless transceiver. They run specially designed operating systems for them such as Contiki [3] and TinyOS [5]. These motes are supposed to sense their environment using the on-board sensors and report to somewhere the data can be processed. This nature of WSN applications requires the motes to be deployed in large numbers and communicate over multi-hop routing protocols such as CTP [4] and RPL [10].

Recent wireless sensing applications exhibit interesting trends as compared to their counterparts a decade ago. Instead of communicating over long distance multi-hop networks, modern WSN applications tend to rely mostly on a lower number of hops or in extreme cases single-hops to deliver data out of the network for further processing. Tracking flying foxes with light weight wireless transceivers attached to their bodies is a recent example [9], [6]. When high mobility makes it harder to manage ad-hoc communication, cellular networks have also been used straightforwardly [1]. This trend is mainly backed by the availability of cheap mechanisms to place Internet connected sink nodes without incurring a significant cost. Any resource constrained node at least can have a single hop neighbor with such capabilities. Raspberry Pi and Smartphones are promising platforms for this role in a wireless sensor network [11], [8], [2]. Many other wireless applications such as device free localizations [12] use low capable wireless nodes which do not require multi-hop communication.

II. Hardware for wireless sensors

The details given in Table I illustrate the wide variety of MCU units available in the market starting from 8bit MCUs to the latest 32bit MCUs with higher clock speeds. Some of them have already proven their capabilities in mote-class devices such as *ATmega128L* used in *MicaZ* and *MSP430F1611* used in *Tmote Sky*. The table highlights the fact that more capable MCUs are becoming available in the market that provides significantly higher processing capabilities. However in addition to the processing capabilities they provide, these modern MCUs, such as *Cortex-M* class, are increasing their energy consumption for the excess processing power they provide.

Meanwhile, people such as hobbyists and DIY enthusiasts have came up with simple designs that they use for transmitting various data between embedded devices. These applications vary from home monitoring into industrial sensing applications. Such devices with open hardware designs are easy to build and extremely low cost as compared to traditional WSN motes. However, they does not contain a powerful embedded operating systems and a *6LoWPAN* communication protocol stack which makes them unable to be accessed by the outside world over the Internet.

Fig. 1. Our simple wireless sensor which consists of a *ATtiny85* MCU together with a *nRF24L01* radio which operates in 2.4 GHz band.

TABLE I
Comparison of MCU units for low power embedded applications.

Module	RAM	ROM	Inst.	Current (uA)
ATtiny85	512B	8kB	8bit	300@1MHz
ATmega128L	4kB	128kB	8bit	5000@4MHz
MSP430G2553	512B	16kB	16bit	330@1MHz
MSP430F1611	10kB	48kB	16bit	330@1MHz
LPC1100L	4kB	16kB	32bit	840@1MHz
Cortex-M4	64kB	512kB	32bit	47810@80MHz

TABLE II
COST AND ENERGY CONSUMPTION OF 2.4GHZ RADIO MODULES.

Module	Unit cost	Tx	Rx
nRF24L01+	US$ 1.50	11.3mA	13.5mA
cc2420	US$ 5.50	17.4mA	18.8mA
CC1120	US$ 6.00	45mA	17mA

III. DO WE NEED A FULL NETWORKING STACK?

A sensor mote with a full networking stack such as *6LoWPAN*, a longer transmission range and a powerful MCU can sometimes operate for few years before completely depleting its batteries. However, the operational lifetime of a wireless sensor mote drastically decreases with the rate of packet transmissions and receptions performed. It makes the radio transceiver the most critical component of a sensor mote. A packet transmitted in such a network mainly consists of various headers specific to different layers of the communication stack in addition to the sensor data. For example when *6LoWPAN* is used, a 802.15.4 frame with the size of 127 bytes contains around 73 bytes of various headers namely MAC header of 25 bytes, IPv6 header of 40 bytes and finally a UDP header of 8 bytes [7]. Additionally, various other protocol specific control packets need to be transmitted which does not necessarily contain any useful sensor data. Under such circumstances, having a complete networking stack on an extremely resource constrained wireless sensor is questionable and deserves further studies.

With the hope of studying this situation in long term, we have started working on simple hardware designs associated with simple MAC protocols. For our basic prototype implementation of wireless sensors, we use *ATtiny85* MCUs together with *nRF24L01* radio which operates in 2.4 GHz band. These components are considered due to their lower energy consumption and cheaper unit cost (see Table II). The *nRF24L01* radio transceiver has four transmitter power amplifier levels which can be adjusted by the firmware. A considerably higher transmission power level significantly increases the transmission range of a wireless sensor as shown in Figure 2 where the transceiver achieved around 30 m range within an indoor environment.

IV. FUTURE DIRECTION

Due to the increasing availability of instant Internet connectivity anywhere and anytime over technologies such as cellular networks and Wifi, this poster describe and opens a discussion regarding the need of more simpler and battery saving architectures for wireless sensor applications with extremely simple wireless sensors and resource rich Internet connected devices. Removing resource hungry communication protocol stacks and delivering data over a single hop has a potential of becoming a promising way to build long lasting wireless sensors for various application scenarios in the post-mote era. This poster presents a work-in-progress of this investigation.

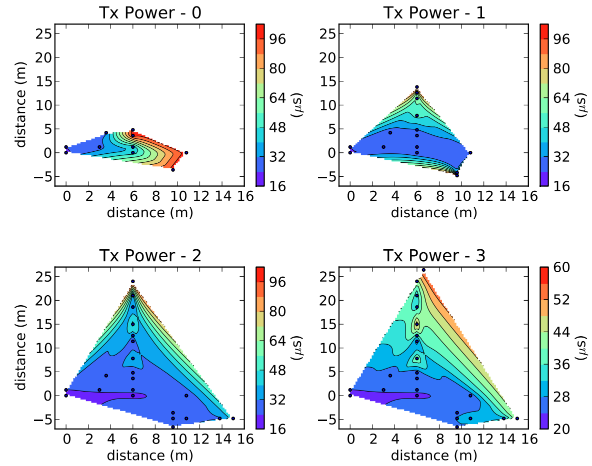

Fig. 2. Indoor transmission range of the our wireless sensor for fours (4) of its transmission power amplifier levels. This was done in a L shaped corridor where one device performs transmissions to another moving device which reply with the same content over a single hop. Color codes depicts the round-trip delay.

REFERENCES

[1] D. Anthony, W. P. Bennett, M. C. Vuran, M. B. Dwyer, S. Elbaum, A. Lacy, M. Engels, and W. Wehtje. Sensing through the continent: towards monitoring migratory birds using cellular sensor networks. In *Proceedings of the 11th international conference on Information Processing in Sensor Networks*, pages 329–340. ACM, 2012.

[2] G. Challen, S. Haseley, A. Maiti, A. Nandugudi, G. Prasad, M. Puri, and J. Wang. The mote is dead: long live the discarded smartphone! In *Proceedings of the 15th Workshop on Mobile Computing Systems and Applications*, page 5. ACM HotMobile, 2014.

[3] A. Dunkels, B. Gronvall, and T. Voigt. Contiki - a lightweight and flexible operating system for tiny networked sensors. *Local Computer Networks, Annual IEEE Conference on*, 0:455–462, 2004.

[4] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis. Collection tree protocol. In *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*, pages 1–14. ACM, 2009.

[5] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister. System architecture directions for networked sensors. *SIGPLAN Not.*, 35:93–104, November 2000.

[6] R. Jurdak, P. Sommer, B. Kusy, N. Kottege, C. Crossman, A. Mckeown, and D. Westcott. Camazotz: multimodal activity-based gps sampling. In *Proceedings of the 12th international conference on Information processing in sensor networks*, pages 67–78. ACM, 2013.

[7] N. Kushalnagar, G. Montenegro, C. Schumacher, et al. Ipv6 over low-power wireless personal area networks (6lowpans): overview, assumptions, problem statement, and goals. *RFC4919, August*, 10, 2007.

[8] Q. Li, D. Han, O. Gnawali, P. Sommer, and B. Kusy. Twonet: large-scale wireless sensor network testbed with dual-radio nodes. In *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*, page 89. ACM, 2013.

[9] P. Sommer, B. Kusy, A. McKeown, and R. Jurdak. The big night out: Experiences from tracking flying foxes with delay-tolerant wireless networking. In *Real-World Wireless Sensor Networks*, pages 15–27. Springer, 2014.

[10] R. Team et al. Rpl: Routing protocol for low power and lossy networks. *IETF ROLL WG, IETF Internet-Draft*, 4, 2009.

[11] V. Vujovic and M. Maksimovic. Raspberry pi as a wireless sensor node: Performances and constraints. In *Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2014 37th International Convention on*, pages 1013–1018. IEEE, 2014.

[12] Y. Zhao, N. Patwari, J. M. Phillips, and S. Venkatasubramanian. Radio tomographic imaging and tracking of stationary and moving people via kernel distance. In *Proceedings of the 12th international conference on Information processing in sensor networks*, pages 229–240. ACM, 2013.