

Smart Decision Fog Computing Layer in Energy-Efficient Multi-hop Temperature Monitoring System using Wireless Sensor Network

Krzysztof Daniluk

Faculty of Electronics and Information Technology,
 Warsaw University of Technology System Control Division,
 Complex Systems Group, Warsaw, Poland K.Daniluk@elka.pw.edu.pl

Abstract—Smart decision layer, as Fog Computing layer in WSN is presented and discussed. Working WSN sleep state mode analysis is presented in multi-hop temperature monitoring system using wireless sensor network in environmental monitoring scenario.

The monitoring system consists of wireless sensor nodes, called motes. In laboratory experiment are used SunSPOT motes, invented by Sun Microsystems, now manufactured by Oracle. Temperature analyzing application is presented, forwarding measured data in a multi-hop way to the network host. Desktop computer plays here a role of temperature measurements database. Link-Quality Routing Protocol (LQRP) is used, which is based on Ad-hoc On-Demand Distance Vector (AODV), which is reactive routing protocol.

All gathered measurements, all data are sent to the base-station node, which is a special WSN node, directly connected via USB to the computer. Experiments were conducted taking into consideration especially SunSPOTs' processor board parameters - sleep state modes. Their processor board is turned into different sleep modes for different periods of time. A comparison of sleep mode time periods in total time of running an application is discussed. Periods of sleep states are different depending on how long is turned on the utility thread sleep.

The experiment's results show, that there is a need to build smart decision layer as fog computing layer for WSN. It should be done in order to manage in a better way environmental monitoring with usage of different kind of sensors, especially to better manage transferring big portions of data from sensors to the Cloud data storage. Sensors, which build WSN, should be distributed over specific areas, as well as fog computing devices should be able to communicate with sensors in order to transfer only relevant data to the Cloud.

Index Terms—Wireless Sensor Networks, Internet of Things, Fog Computing, Intelligent layer

I. INTRODUCTION

Experimental version of environmental monitoring system is discussed and presented as working in real-life

temperature monitoring-oriented scenario using Wireless Sensor Networks (WSNs). The idea is to present application, measuring and sending gathered data in a multi-hop manner to the base-station node, connected to the computer. As a result of this paper is presented Fog Computing layer, enabling in more energy-efficient way to manage of data from WSN to the Cloud data storage. In chapter II Wireless Sensor Networks are introduced. In chapter III laboratory equipment is described. In chapter IV SunSPOT processor board energy states, experiment and results are presented. Chapter V presents, as a solution to solve experiment's problems, Fog Computing in Wireless Sensor Networks. Chapter VI summarizes the article.

II. WIRELESS SENSOR NETWORK

Wireless Sensor Network is a distributed system [1] with hundreds of small-size embedded devices, which are deployed densely over a specific area. Each sensor node participates in transferring data to other sensor nodes or base-stations, which are within its range. WSNs have been identified [2] as one of the most important technologies of this century. They have CPU power, radio transceiver and sensing capabilities, i.e. plenty of sensor devices can be deployed in a sensing area [5]. Such instruments can be deployed in a sensing area, where traditional networks are not able to be used. The problem is, that wireless sensor nodes are often small battery-fed devices, which underlines the fact, that their power source is very limited [2, 3]. An important issue is that the quality of wireless transmission depends on numerous external factors [6, 7, 8], like weather conditions or landform features.

In fig. 1 we see an example of wireless sensor network, with free-range motes/intermediate motes (without the black cover) and base-station on the right-hand side.

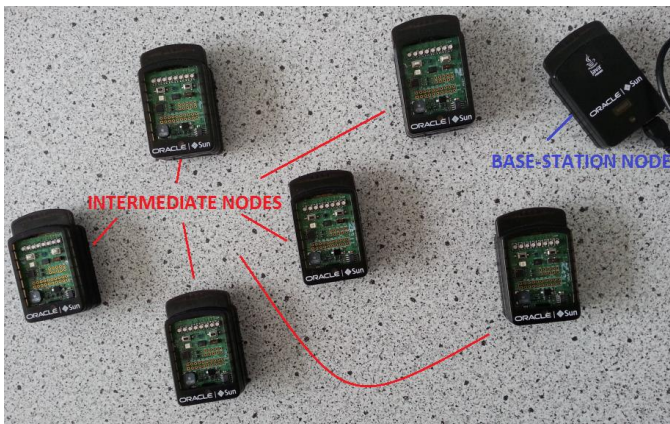


Fig. 1 Wireless sensor network structure

III. LABORATORY EQUIPMENT – HARWARDE & SOFTWARE

The laboratory equipment consists of 18 wireless sensor nodes: 12 free-range motes and 6 base-stations. Wireless sensor nodes are called SunSPOTs, they are originally invented by Sun Microsystems, now are manufactured by Oracle. The main difference between free-range mote and base-station is the fact, that free-range mote has sensor board and battery. Base-station can receive radio transmission only, when is plugged by USB to computer, no sensing capability is provided.

Figure 2 presents technical specification, of used in experiment, wireless sensor node - SunSPOT

- ▶ IEEE 802.15.4 standard for networking
- ▶ Using Squawk Java Virtual Machine
- ▶ Processor board: ARM architecture 32 bit CPU with ARM926EJ-S core running at 400MHz
- ▶ 1Mbytes SRAM memory
- ▶ 8Mbytes Flash Memory
- ▶ 2.4 GHz IEEE 802.15.4 radio with integrated antenna
- ▶ 3-axis accelerometer
- ▶ Sensors: temperature & light
- ▶ 8 tri-color LEDs, analog & digital inputs, 2 switches
- ▶ 720mAh lithium-ion battery
- ▶ Deep sleep mode: 33 uA, i.e. 909 days



Fig. 2 SunSPOT technical data

In fig. 3 we can see overview of SunSPOT's hardware, described in fig. 2



Fig. 3 Sunspot's hardware

IV. EXPERIMENT

In this part are presented following topics: A – Routing table for used multi-hop routing protocol, B – SunSPOT's processor board energy states, C – The results of experiment.

A. Table of routing

In the experiment LQRP routing protocol is used. This kind of protocol, offered by SunSPOT API, is based on AODV multi-hop and reactive routing protocol.

Fig. 4 presents an example of routing table, where is shown destination address, next hop address and number of hops to achieve the destination.

Destination	Next Hop	Hops
0014.4F01.0000.7C68	0014.4F01.0000.7CAC	3
0014.4F01.0000.7F92	0014.4F01.0000.7CAC	3
0014.4F01.0000.7672	0014.4F01.0000.7CAC	2
0014.4F01.0000.7631	0014.4F01.0000.7CAC	4
0014.4F01.0000.7CAC	0014.4F01.0000.7CAC	1

Fig. 4 Example of SunSPOT's routing table

B. SunSPOT's processor board energy states

Table 1 presents SunSPOT's processor board energy states [4]. This helps to understand how WSN nodes may behave, which kind of energy state may choose, depending on the situation, whether radio and sensor board are used or not. In the table 1 we can see different energy states and their corresponding radio and sensor board status, and finally current draw in particular energy states.

ENERGY STATE	RADIO status	SENSOR BOARD status	CURRENT DRAW
Deep sleep	0	0	33 μ A
Shallow sleep	0	0	24 mA
Shallow sleep	1	0	40 mA
Awake, active calculating	0	0	80 mA
Awake, active calculating	1	0	98 mA
Shallow sleep	0	1	31 mA
Shallow sleep	1	1	46 mA
Awake, active calculating	0	1	86 mA
Awake, active calculating	1	1	104 mA

Table 1 SunSPOT's Processor board energy states [4]
(0 = OFF, 1 = ON)

C. Results of experiments

In this section we can see experiment's results.

There is defined term **MBST – Main Board Sleep Time** calculated in milliseconds. When sleeping thread is run, it stops execution the application. The aim of the application is to forward temperature readings from one mote to another (based on routing table) in a multi-hop way. There are defined 2 types of WSN nodes: intermediate (forwarding temperature measurements to the next hop node, based on routing table) and base-station node – which collects all data from the WSN. Points of execution of the program are following:

- Inside the intermediate node, step 1 – looking for a multihop route to the base-station and opening the socket connection

- Inside the intermediate node, step 2 – sending the temperature measurement to base-station node based on routing table next-hop parameter
- Base-station node is not analyzed from the processor board energy states point of view, because is assumed it is all the time connected to the computer via USB cable

Energy state analysis covers intermediate WSN nodes.

Table 2 presents results of experiment.

	Total time of experiment (seconds)		
	10	30	60
MBST (milliseconds)	(Shallow sleep time / total time) %		
1	42,84%	46,64%	48,2%
2	43,61%	48,22%	50,07%
3	45,25%	48,51%	50,29%
10	45,83%	49,35%	56,57%
100	59,19%	64,55%	68,92%
1000	74,75%	86,67%	90,69%

Table 2 Experiment's results

Application, deployed on each intermediate node, is monitoring and sending by multi-hop temperature readings. **Main sleeping thread turns into sleep state for time between 1 and 1000 milliseconds.** This has influence for introducing shallow sleep time energy state, which has percentage part in total time of execution of an application.

Shallow sleep state mode of SunSPOT's processor board, is turned on, when main sleeping thread lasts less than 3000 milliseconds. After such period of idle time, deep sleep is turned on automatically, which has influence in waking up procedure all processor board elements. **From point of view of procedure of fast waking-up time, flexible change between sleep and wake-up supports shallow sleep, which seems to be the most accurate, comparing to deep sleep mode [4].**

Table 2 presents results of experiment in 3 different scenarios, when total time of experiment lasts 10, 30 or 60 seconds. In each scenario, there is run main sleeping thread, called MBST – Main Board Sleep Time, which duration is 1,2,3,10, 100 or 1000 milliseconds. For each use case is presented percentage contribution of shallow sleep time (with radio and sensor board turned on or turned off, but with no active calculations provided) in total time of running temperature analyzing application. In shallow sleep time, current draw, varies between 24mA to 46mA, depending on the fact, whether radio/sensor board is turned on or off.

D. Conclusions from the experiment

Experiment's results, shown in table 2, underline the fact, that it is hard to combine very frequent data readings (short main sleeping threads, i.e. short duration of MBST) with long lasting shallow sleep time of the whole working system.

If shallow sleep time is often, we can say, that such system is energy-efficient, i.e. is often in a sleep state mode and is able to wake up quickly. For vineyard monitoring scenario it may be crucial, for different weather conditions, to be ready for very changeable, in frequency of gathering, data readings. This happens also for data readings taken very rarely, if no major changes appear in weather conditions.

In order to work in that way, i.e. flexible changing frequency of gathering data from sensors, dependable on weather conditions, as well as better managing big data taken from WSN, Fog Computing layer is introduced to use in WSN.

V. FOG COMPUTING IN WSN

A. Fog Computing in WSN

In this chapter is proposed Fog Computing layer for WSNs in order to manipulate in more efficient way [9, 10] with different energy states in SunSPOTs (shallow sleep, deep sleep, awake).

Fog Computing layer should be understood as a bridge between the Cloud data center (where WSN data readings may be stored) and the Internet of Things, here WSNs.

Fog is a distributed computing architecture, a paradigm extending Cloud Computing to the edge of the network. Fog, like Cloud, provides data, computation, storage and application services to end-users. The main characteristics for the Fog is its proximity to end-users, used especially to reduce service latency, enabling real time big data analytics.

Fog devices are heterogeneous devices like access points, edge routers and switches, end-user devices. The Fog platform supports real-time analytics processes, filters the data and pushes to the Cloud. There is waste of network resources to transmit plenty of gigabytes of data to the cloud and also in doing analytics in that manner. Better way is to analyze data locally and then decide what should be passed to the Cloud and what requires immediate reaction.

There is presented architecture for Fog Computing layer, with gathering data from WSN, analyzing and forwarding only the necessary data for the end user. Fog Computing devices, in their analytical work, should cover such parameters like: packet size, signal strength, volume traffic controller, radio controller, sensor board controller & master clock controller of each sensor from the WSN.

The aim of Fog Computing smart decision layer in WSN is to control shallow and deep sleep mode in more flexible way [19, 20], as well as to better manage Big Data, i.e. doing the whole analytical work.

Nowadays, whole analytical work is done in WSN by each sensor separately. In this architecture, whole analytical work

will be done by Fog Computing devices, which collect sensor readings and give them feedback with information to change e.g. their frequency of the readings.

Fog Computing devices may also act, as control devices of Wireless Sensor Networks, i.e. they may store in their databases the battery level of each sensor, calculate the battery level of each sensor based on transmission analysis, without asking each sensor about its battery level, i.e. without asking about additional data from sensors, which may be calculated by Fog Computing layer. **The next step for further experiments is to start to build Fog Computing layer for WSN in order to enhance the energy-efficient work [13, 14, 15] of wireless sensor network nodes not only based on turning on/off the main sleep scheduler, like done in the experiment, but also with manipulating the work [12] of master clock controller, changing the signal strength, introducing volume traffic controller in Fog Computing layer.**

B. Future work

Next experiment will introduce decreasing the computational power, controlling the traffic volume and radio signal strength. It is assumed [11], that these new parameters will extend the length of shallow sleep duration, so will introduce new energy-efficient way [16, 17, 18] for resource scheduling in Wireless Sensor Networks. Table 3 presents concept for new parameters in Fog Computing layer in WSN.

WSN Fog Computing layer (Resource scheduling of each sensor from WSN)	
PARAMETERS	OBJECTIVE FUNCTION
1) Packet size	Shortest data frame with deep sleep indication flag
2) Master Clock Controller	Decreasing the computational power after load level analyzing
3) Signal strength (transmission power control)	Decreasing the radio power, if received signal strength indicator (RSSI) give assumption it is safe and possible
4) Radio Controller	Radio on/off, radio channel
5) Sensor Board Controller	Sensors on/off connected
6) Volume traffic controller	Radio Controller + Sensor Board Controller

Table 3 Concept for new parameters in WSN Fog Computing layer

Fig. 5 presents new concept of Fog Computing layer in WSN, which will be introduced in experiments analyzing duration of shallow-sleep time in total time of working application in WSN.

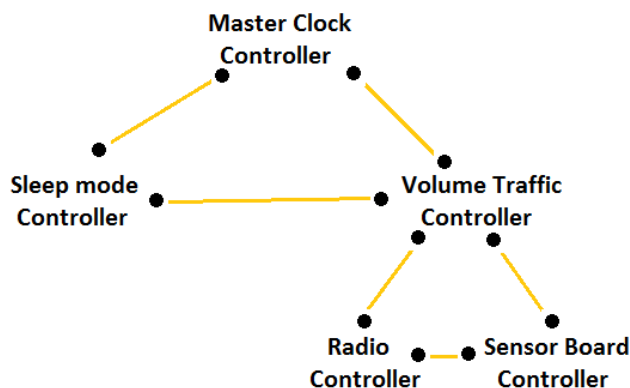


Fig. 6 Concept of Fog Computing layer in WSN

Below are proposed main parts of Fog Computing layer in WSN:

- **Master Clock Controller**
Analysis and changing master clock frequency
- **Sleep Mode Controller**
Analysis and changing sleep mode (deep, shallow, active)
- **Volume Traffic Controller**
Analysis how much data where transferred in amount of time in the past – analysis for the future, expected traffic
- **Radio Controller**
Analysis and turning on/off the radio. Changing radio power (RSSI)
- **Sensor Board Controller**
Analysis and turning on/off selected sensor (temperature, light, accelerometer)

VI. CONCLUSION

Fog Computing layer in WSN is presented as a solution for problems presented in the experiment. Such architecture layer should be implemented in order to better manage vineyard monitoring transmissions for different kind of sensors and especially to better manage transfer of Big Data from sensors to the Cloud data storage. Fog Computing devices should be able to communicate with sensors in order to transfer only relevant data to the Cloud. Concept of Fog Computing layer in WSN is presented and as a future work is going to be implemented in the next experiment. Presented concept supports energy-efficiency in Wireless Sensor Networks.

REFERENCES

- [1] M. C. Vuran, I. F. Akyildiz, *Wireless Sensor Networks*. Wiley, 2010.
- [2] A. Tiwari, P. Ballal, and F. L. Lewis, Energy-efficient wireless sensor network design and implementation for condition-based maintenance, *ACM Trans. Sensor Networks (TOSN)*, vol. 3, no. 1, pp. 1–23, 2007.
- [3] S. R. Gandham, M. Dawande, R. Prakash, and S. Venkatesan, S., Energy efficient schemes for wireless sensor networks with multiple mobile base stations, in *Proc. IEEE Global Telecom. Conf. GLOBE-COM'03*, San Francisco, USA, 2003, vol. 1, pp. 377–381.
- [4] <https://java.net/projects/spots-core-libraries/sources/svn/content/trunk/sdkresources/resources/doc/SunSP-OT-Programmers-Manual.pdf> (accessed: May 2015)
- [5] P. Baronti, P. Pillai, V. W. C. Chook, S. Chessa, A. Gotta, and Y. Fun Hu, “Wireless sensor networks: a survey on the state of the art and the 802.15.4 and ZigBee standards”, *Comp. Commun.*, vol. 30, no. 7, pp. 1655–1695, 2007.
- [6] ZigBee Alliance, “ZigBee Specification v1.0”, New York, USA, 2005.
- [7] W. R. Heinzelman, A. Chandrakasan, H. Balakrishnan, “Energy-efficient communication protocol for wireless microsensor networks”, in *Proc. 33rd Annual Hawaii Int. Conf. Sys. Sciences HICSS'00*, Maui, Hawaii, USA, 2000, pp. 3005–3014.
- [8] B. Azzedine, C. Xiuzhen, and J. Linus, “Energy-aware datacenter routing in microsensor networks”, in *Proc. 6th Int. Symp. Model. Anal. Simul. Wirel. Mobile Sys. MSWiM 2003*, San Diego, CA, USA, 2003, pp. 42–49.
- [9] K. Lin, Ch. F. Lai, X. Liu, and X. Guan, “Energy efficiency routing with node compromised resistance in wireless sensor networks”, *Mob. Netw. Appl.*, vol. 17, pp. 75–89, 2012.
- [10] M. I. Shukur, L. S. Chyan, and V. V. Yap, “Wireless sensor networks: delay guarantee and energy efficient MAC protocols”, *World Academy of Sci., Engin. Technol.*, vol. 50, pp. 1061–1065, 2009.
- [11] S. R. Gandham, M. Dawande, R. Prakash, and S. Venkatesan, S., Energy efficient schemes for wireless sensor networks with multiple mobile base stations, in *Proc. IEEE Global Telecom. Conf. GLOBE-COM'03*, San Francisco, USA, 2003, vol. 1, pp. 377
- [12] K. Daniluk, E. Niewiadomska-Szynkiewicz, *Energy-Efficient Security in Implantable Medical Devices*, FedCSIS 2012 Proceedings, pp. 773–778
- [13] H. Cam, S. Ozdemir, D. Muthuavinashiappan, and P. Nair, “Energy efficient security protocol for wireless sensor networks”, in *Proc. IEEE 58th Veh. Technol. Conf. VTC 2003*, Orlando, Florida, USA, 2003, vol. 5, pp. 2981–2984.
- [14] Zhu, S. Setia, and S. Jajodia, “LEAP+: Efficient security mechanisms for large-scale distributed sensor networks”, *ACM Trans. Sensor Netw. TOSN*, vol. 2, no. 4, pp. 500–528, 2006
- [15] Zhu, S. Setia, and S. Jajodia, “LEAP: Efficient Security Mechanisms for Large-Scale Distributed Sensor Networks”, in *Proc. 10th ACM Conf. Comp. Commun. Secur. CCS 2003*, Washington, DC, USA, 2003, pp. 62–72.
- [16] L. E. Lighfoot, J. Ren, and T. Li, “An energy efficient link-layer security protocol for wireless sensor networks”, in *Proc. IEEE Int. Con. Elec.-Infor. Technol. EIT 2007*, Chicago, IL, USA, 2007, pp. 233–238.
- [17] S. K. Singh, M. P. Singh, and D. K. Singh, “Energy-efficient homogenous clustering algorithm for wireless sensor networks”, *Int. J. Wirel. Mob. Netw.*, vol. 2, no. 3, pp. 49–61, 2010.

- [18] S. K. Singh, M. P. Singh, and D. K. Singh, "A survey of energy-efficient hierarchical cluster-based routing in wireless sensor networks", *Int. J. Adv. Netw. Appl.*, vol. 2, no. 2, pp. 570–580, 2010.
- [19] C. Castelluccia, A. C.-F. Chan, E. Mykletun, and G. Tsudik, "Efficient and provably secure aggregation of encrypted data in wireless sensor networks", *J. ACM Trans. Sensor Netw. (TOSN)*, vol. 5, no. 3, 2009
- [20] Niewiadomska-Szynkiewicz, P. Kwaśniewski, and I. Windyga, "Comparative study of wireless sensor networks energy-efficient topologies and power save protocols", *J. Telecom. Inform. Technol.*, no. 3, pp. 68–75, 2009