FACTA UNIVERSITATIS
Series: **Mechanical Engineering** Vol. 9, N° 2, 2011, pp. 169 - 182

CONNECTING WIRELESS SENSOR NETWORKS TO INTERNET[*]

UDC 681.324

Mirko R. Kosanović[1], Mile K. Stojčev[2]

[1]High Technical School of Professional Studies Niš
E-mail: mirko.kosanovic@open.telekom.rs
[2]Faculty of Electronic Engineering, University of Niš

Abstract. *Wireless Sensor Networks (WSNs) have become one of the most interesting, and probably the most researched areas in the field of electronics in the last decade. The paper represents a collection of tiny, inexpensive wireless sensor nodes (SNs), organized in clusters and networks, deployed over a wide geographical area, capable of integrating continuous and unobtrusive measurements, computing and wireless communication, completely autonomously. But, WSNs usually cannot operate independently, in full isolation, i.e. they have to be connected to some other kind of network (LAN, WAN). As the Internet becomes de-facto standard for WANs, it is a challenge now to connect WSNs to WAN and to allow the collected information to be visible from various places. In this paper we will firstly give the basic characteristics of WSN affecting the WSN connection with Internet, and then proceed to analyze the current well-known solutions. After that we will consider the requirements that SNs must meet in order to be able to connect to the Internet, and propose a simple solution.*

Key words: *TCP/IP Networks, Data Collection, Internet, Wireless Sensor Networks*

1 INTRODUCTION

WSNs are composed of a large number of radio-equipped sensor devices that autonomously form a network, through which SNs are capable of sensing, processing and communicating among each other. Some crucial properties of WSNs are the following: a) nodes are densely deployed in a region and are very often prone to failures; b) broadcast communication paradigm, mainly used without global identification (ID), is implemented; c) nodes operate under limited power; and d) computational capacity and memory space of each sensor node is limited, too. In general, WSN can operate as stand-alone networks or be connected to other networks. But for many applications, WSNs do not work efficiently in full isolation. There must be a way for a monitoring entity to gain access to the

Received February 9, 2011
[*] **Acknowledgments**. This work was supported by Serbian Ministry of Science and Technological Development, project No. T-32009 - "Low Power Reconfigurable Fault Tolerant Platforms"

data produced by the sensor network. So, it is imperative to connect them to an existing network infrastructure, such as Local Area Network (LAN), Metropolis, WAN or global Internet. In this way a very complex heterogeneous distributed network (HDN) can be configured. Such connection, on one hand, provides transparent operation of the WSN for all HDN's end users, but, on the other, creates new challenges related to the development and research in this field. Having in mind that TCP/IP protocol suite becomes de-facto standard in network connectivity, it is quite reasonable to look at some efficient methods for interconnecting protocol specific WSN to TCP/IP based network, such as for example Internet. The task of connecting WSN to the existing Internet infrastructure brings with it several challenges. Any network wishing to be connected to the Internet needs to address the question of how it will interface with the standard TCP/IP protocol suite. This problem is in focus of our interest in this paper. The purpose of the proposed research will be to investigate ways of connecting such WSN to the Internet, i.e. enabling TCP/IP support. Having in mind that communications have a dominant effect on power consumption, we propose header length reduction. The obtained results show that energy efficiency achieved by involving shortage of TCP/IP header is better for 60 % with respect to standard TCP/IP header used in Internet.

2 WSN CHARACTERISTICS

Before starting with explanations of how to cope with the problem of connecting WSN with Internet, it is necessary firstly to point out to some details, concerning specifications of WSN operation and SN architecture, which have a dominant impact in communication between SNs, between SNs and master node (sink), and between sink and Internet [1].

2.1 Dynamic topology

In most WSN applications we assume that the SNs are stationary. This means that SNs can in advance determine the optimal paths among SNs, which are valid until the application is running. However, in reality it is not so, because WSN topology changes frequently.

2.2 Limited data rate and short distance

The SN's electromagnetic range covers short distances (from one to several tens of meters). This directly determines the necessity of application multi-hop topology in WSN, which leads to a dual role of all SNs: as hosts and routers.

2.3 Different traffic intensity

The most basic use of WSN is to treat each node as an independent data collection device. Periodically, each node in the network sends its readings to a central SN usually called sink. As a direct consequence of this is that the highest traffic density in WSN is happening around the central SN (sink), because it collects all data coming from other SNs located in its vicinity (upstream traffic). Quite opposite, very little traffic is happening around SNs which directly collect data and in the other direction, from sink to SNs (downstream traffic).

2.4 Application-specific networking

The traditional IP-based networks follow the layering principle which separates the application level concerns from network layer routing. This is necessary because a multitude of applications are expected to run over a common networking. By contrast, sensor networks are likely to be quite limited in the applications they perform, only one in time, for specific WSN. This indicates that we must have cross-layer optimizations and application-specific designs.

2.5 Energy Constraints

The nodes in unattended large-scale sensor networks are likely to be battery powered, with limited recharging capabilities. For deployment with hundreds of sensors, this means that a battery will need a replacement every few days, what represents an unsuitable rate for many applications. Several solutions to the power problem exist, such as reducing power consumption to the point where batteries can elongate the sensor module's lifetime. Another solution is energy harvesting or energy scavenging, which means that SN is capable for extracting energy from ambient sources. Common energy ambient sources for energy harvesting include mechanical energy resulting from vibration, stress and strain; thermal energy from furnaces and other heating sources; solar energy from all forms of light sources, ranging from lighting to the sun; electromagnetic energy that is captured via inductors, coils and transformers: wind and fluid energy resulting from air and liquid flow; human energy which depends of human movement by foot, human skin and blood; and chemical energy from naturally recurring or biological processes.

2.6 Time Synchronization

Synchronized network time is an essential aspect for energy efficient scheduling and power management of SNs within a WSN. It allows SNs to shutdown their RF transceivers and other peripherals, even microcontrollers, in order to enter power saving mode, and later to return to normal operation mode. The following three solutions for time synchronization in WSN are used [2]:

1. *One-way message dissemination*: the simplest form of synchronization deals only with ordering of events or messages. Using this approach, it is possible to tell whether an event $\mathbf{E_1}$ has occurred before or after another event $\mathbf{E_2}$.

2. *Receiver-receiver synchronization*: SNs run their local clocks independently, but they keep information about the relative drift and offset of their clock to other clocks in the network.

3. *Two way message exchange*: the most complex form of synchronization called "always on" model, where all SNs maintain a clock that is synchronized to a reference clock in the network.

2.7 Sensor Node Constraints

The SN is constituent element of each WSN. It consists of several building blocks: power supply, sensing, computing, communication, and optional: mobile unit, coordinate unit, time system synchronizer and so on (Fig. 1). Sensors as part of a SN are usually devices which measuring physical parameters like temperature, pressure, humidity, energy

consumption or acceleration and transforming them into electrical signals. These signals need now to be transformed into the corresponding data with A/D conversion [3]. Due to limited amount of battery capacity, SNs are characterized by a restricted number of hardware resources (sensors, memory capacity, input-output peripherals, etc.). Because of that, the implementation of the full IP stack in WSN may not be feasible primarily because they require additional computational and memory resources on SNs. WSNs are thus in many ways fundamentally different from traditional IP-based networks. For these reasons, all-IP large-scale sensor networks are neither desirable nor feasible.

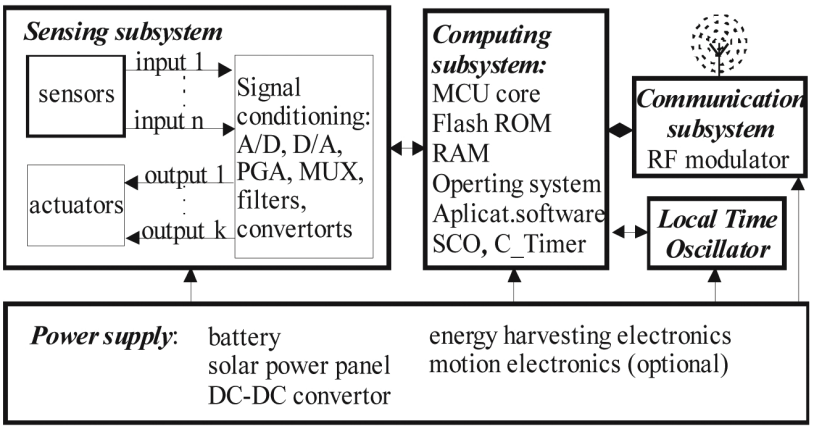


Fig. 1 System architecture of a typical sensor node

3 APPLICABILITY OF TCP/IP PROTOCOLS IN WSNS

With the rapid development of wireless technology, the increased requirements for implementation of TCP/IP based networks become a necessity. To solve this problem efficiently is not an easy task, especially in WSNs, where SNs are realized with many limited resources. Without doubt, one of the more pronounced design challenge relates to SN's is power consumption. Namely, since the energy of the battery powered SN is limited, in order to prolong its life, micro power consumption for SN is of paramount importance. Numerous research projects done in this field [4], [5], [6], [7] show that the communication buildings block is the largest energy consumer of the wireless SN. For example, the energy which is consumed to send only one bit of data is less or equivalent than the amount of energy needed to process 100 instructions for Berkeley node [8]. The TCP/IP protocol is often perceived to be "heavy-weight" protocol, because its implementation requires large amounts of resources both in terms of memory and processing power. There are, however, a number of other problems that currently prevent TCP/IP from being directly applicable to sensor networks. These include:

- Tiny TCP/IP implementation: it has often been said that the TCP/IP protocol stack is too heavy to be squeezed into such a tiny system as a wireless SN. Many other protocols similar TCP/IP, which is small enough to be useful in such systems, is developed such as μIP TCP/IP, SIP, and u-IP [9].
- Spatial IP address assignment: In IP networks, each host is required to have an IP address. In large-scale sensor networks, it is not possible to manually configure the addresses and we cannot rely on a central server. Instead, we have designed a spatial IP address assignment scheme, whereby each SN constructs its IP address from its physical location. Since most WSN applications already require the SN to keep track of their location, this mechanism does not increase the complexity of the system.
- Shared context header compression: For TCP/IP, the overhead created by headers can be quite large, particularly for small messages (IPv4-24 byte, IPv6-40 byte, UDP-8 byte, and TCP-24 byte). For example, a four-byte data message would have a header overhead of nearly 90%. Therefore, in this paper we propose to use a header compression mechanism that utilizes the special conditions in sensor networks in order to reduce the header overhead to only a few bytes for messages carrying sensor data.
- Application overlay networking: The address-centric routing in IP does not match the data-centric applications of sensor networks very well.
- Distributed TCP caching: The TCP/IP protocol suite is developed for networks with very low error rates and does not work well in error-prone environments such as WSN. Because of this we must have some mechanism that lets SNs help each other in caching data segments. If the segments are lost because of errors on the radio channel, neighboring SNs are able to re-transmit the lost segments.

Having all this in mind, a direct implementation of TCP/IP protocol in WSN results in an inefficient design solution, i.e. we have to send 30 bytes of message for only 2-3 bytes of useful payload data. To cope up with this problem several methods have been proposed [9], [10], [11].

3.1 Communication Models

WSNs use the following three types of communication models:

1. **Address-Centering communication** – in this case all SNs have a unique ID number and the routing is performed according to IDs. This kind of protocols is referred to as table-driven routing protocols [12].
2. **Data-Centering communication** – SNs are without ID numbers. Communication is based on broadcast messages. There are two kinds of messages: *Interest packet*, which propagates an information interest through the network, and *Advertisement packet* which is replay from SNs on which an interest has been registered [13].
3. **Location-Centering** – SNs use the location as a primary means for addressing and data routing. Each sensor node has a unique spatial address which depends on its physical location in the deployed region.

If we analyze the communication possibilities of all three models and try to implement these models in TCP/IP network, we can conclude the following: Data-Centering model is not a good candidate for TCP/IP networks. In order to provide consistency between WSN and TCP/IP, the Address-Centric and Location-Centering communication paradigms are better solutions for interconnecting WSNs and TCP/IP networks [14].

3.2 Communication Architecture

According to the communication architecture, we divide the communication methods (see Fig. 1) as those that are based on: Proxy architecture, Overlay based architecture and Gateway architecture.

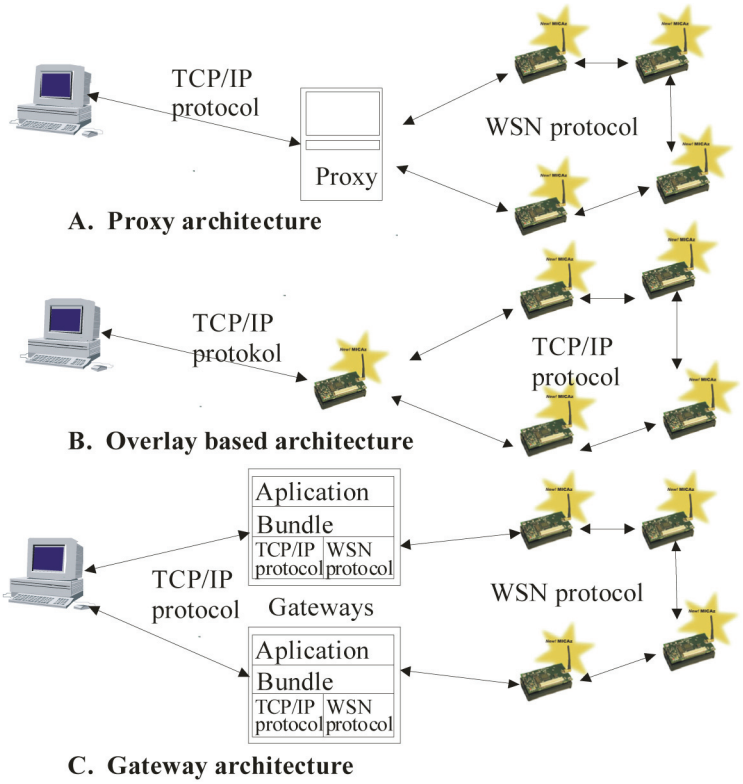


Fig. 2 Types of communication architectures

3.2.1 *Proxy architecture*

The communication between TCP users and SNs is done through the proxy computer. The communication protocol used in the sensor network may be chosen freely. There are two various modes according to which proxy can be operative and can interconnect WSN with TCP/IP networks as [11]:

1. *relay* – in this mode all data which are coming from one network are passed on to another network.
2. *front-end* – the proxy pro-actively collects data from SNs and stores this information in its database. The users from TCP/IP networks can query for specific data in a variety of ways, such as SQL queries or WEB based interfaces.

Both the solutions have some drawbacks, what make them inapplicable in general. As for the first one, a single point of failure exists. All communications to and from the WSN are broken, when the proxy stop working. As for the second, the proxy implementation usually depends on the specific task or a particular set of protocols. This means that for each application a different proxy is needed [11].

3.2.2 Overlay Based Architecture

There are two kinds of overlay based methods: *TCP/IP overlay sensor networks*, and S*ensor networks overlay TCP/IP* [6]. The first approach points out that it is possible to implement TCP/IP protocol stack to microcomputer system with very poor resources: 8-bit microprocessor with only 2kB RAM memory [12]. In *Sensor networks overlay TCP/IP* the protocol stack of WSN is deployed over the TCP/IP stack and each Internet user is considered as a virtual sensor node. The virtual sensor node can interpret WSN packets since it has installed the WSN protocol stack in addition to TCP/IP stack. Numerous problems accompany the implementation of TCP/IP in WSNs. They can be identified as: header overhead, high bit error rates, high energy consumptions for end-to-end multi hop retransmissions, differences in routing protocols and implementation of addressing and routing schemes [4].

3.2.3 Gateway Architecture

One of the essential devices which provide for a connection between wireless and TCP/IP network is a gateway. It performs several tasks such as protocol conversion, message delay, etc. All solutions, that use the gateway as an interconnecting device, can be grouped into the following two categories: *Application gateway* and *Delay Tolerant Network (DTN)*. The application gateway is a simple gateway-based approach which works in application layer [9]. The DTN is a similar solution. The main difference with respect to the *Application gateway* is the following: It implements one new layer, both in TCP/IP and WSN networks, referred as *Bundle Layer.* The main function of the bundle layer is to store and forward packets between two networks (Fig. 2).

4 A GATEWAY: PRINCIPLE OF OPERATION

Main design challenges concerning the interconnection between Internet based networks and WSNs are related to the fact that it is necessary to provide: a) access to each SN through the TCP/IP based network; b) efficient communications from aspect of SN's energy consumption; and c) transparency in operation between TCP/IP based protocols and WSN protocols.

The method which we propose in this paper is suitable for the application domains such as: health monitoring (diagnostics, tele-monitoring), environmental monitoring (fire

detection, water pollution, tracing movements of birds, animal or insects, detection of chemical and biological agents), military and security (movements of soldiers and vehicles, monitoring critical terrain), industrial process control, smart buildings, traffic control, etc. [15]. In general, both regarding the topological hierarchical network organization, on one hand, and control, on the other, these systems are heterogeneous in nature. Therefore, in order to design an optimal solution, for a particular case, it is necessary firstly to foresee the crucial assumptions and requirement that these kinds of networks have to fulfill:

- WSN is organized as a set of clusters.
- Cluster topology is arbitrary.
- Each cluster is organized around one Main Sensor Node, referred as a MSN.
- The MSN acts as a gateway between the WSN and TCP/IP based network.
- To each MSN two addresses are appended. The first one is TCP/IP address, while the second is local WSN's address.
- SNs can access to TCP/IP users through MSN, or contrary.
- Within each cluster, it is possible to address 255 single-addressed SNs, 128 double-addressed SNs, 64 quad-addressed SNs, etc.
- The number of addresses appended to each SN determines the number of messages with which the SN can manipulate simultaneously, i.e. at the time.
- For data transfer, store and forward technique is used.
- Single hop and multi-hop data transfers within the cluster are possible.
- During the initialization phase, the MSN assigns different group of addresses (single, double, quad, octal etc.) to each SN, according to the predicted traffic intensity among SNs and MSN.
- To the TCP/IP users each SN is visible through its first group address.
- MSN can transfer data to: a) SNs located within single cluster area; b) to MSNs that are constituents of other clusters; and c) to TCP/IP users.

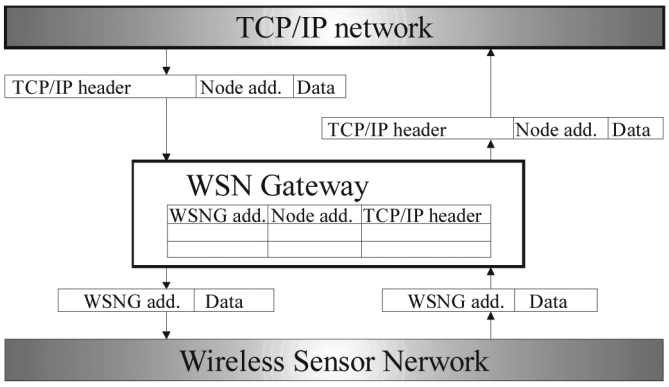


Fig. 3 Block scheme of Gateway principle of operation

Fig. 3 depicts the principle of messages transfer between TCP/IP network and WSN. As can be seen from Fig. 2 the MSN acts as a protocol translator between both the net-

works. From software point of view it maps addresses from one network domain to another, and translates larger TCP/IP header into smaller WSN header. In order to perform this activity it uses data stored in a translation table. The translation table has 256 entries. Each entry has two fields. The first field points to the SN's local address, while the second to the TCP/IP header. For single addressed SN one table entry is appended, to double address SN two table entries are appended, etc. This kind of table organization allows us to direct several TCP/IP messages to the same SN at the time, without extending the WSN header. Smaller header sizes are directly related to lower communications cost, and indirectly to decreased energy consumption of the SN [16].

4.1 Gateway Performance

In order to evaluate the performance of the proposed solution we have assumed the following: a) each SN within the cluster is seen as a TCP/IP addressable unit; b) protocols above and below the network layer remain unchanged; c) data transfer between two communication units is of a store-and-forward type; d) single hop or multihop are allowed; e) all data transfers are error free, i.e. without retransmissions. Successful data transfer of one message between two SNs (message transfer and response) depends on the shortest end-to-end delay. This kind of communication delay includes the following items:

1. T_t (transmission delay) – time needed for one message transmission. It depends on the channel bandwidth, bit rate, message length, and coding techniques.
2. T_p (propagation delay) – signal propagation time between two SNs.
3. T_c (processing delay) – the time needed for processing one message.
4. T_q (queuing delay) – an average time during which the message waits in a queue for transmission.

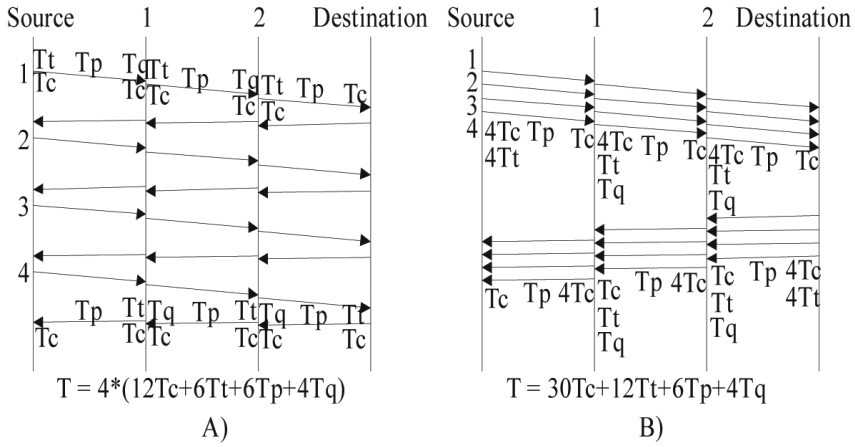


Fig. 4 Time needed for transmission of four messages

The total communication time for the solution given in Reference [4] is defined as:

$$Tref = 2mh(2Tc + Tt + Tp) + 2m(h-1)Tq \quad (1)$$

while for the solution proposed in this paper is:

$$Tps = 2(m+1)hTc + 2(m+h-1)Tt + 2hTp + 2m(h-1)Tq \tag{2}$$

where m is the number of transferred messages, and h corresponds to the number of hops. In Fig. 4A the principle of single message data transfer among four SNs is shown. Fig. 4B corresponds to data transfer of four messages. Note that in this case there is overlapping between data transfers.

In order to evaluate the performance of the two proposals we assume the following:

1. the data transfer rate is R = 720 kbps,
2. one WSN message usually consists of N = 46 bytes,
3. the signal propagation velocity is $v_p = 3*10^8$ m/s,
4. the distance between SNs is uniform, and is within a range d = (10-150) m,
5. CPU clock frequency is f = 12 MHz and the average number of instruction to process one byte is n = 10 instructions with t = 4 clock periods per instruction.

According to the given assumptions and by substituting these values into T_c, T_t and T_p we obtain:

$$Tt = N/R = 46*8/720 \quad kbps = 499.13 \quad \mu s \tag{3}$$

$$Tp = d/v_p = 0{,}1 \quad \mu s \quad \text{for} \quad d = 30 \quad m \tag{4}$$

$$Tc = N*n*t/f = 46*10*4/12 = 153{,}33 \quad \mu s \tag{5}$$

For traffic without retransmissions $Tq = 0$

Having in mind that $T_t >> T_p$ and $T_c >> T_p$ we can ignore T_p and T_q with respect to T_t and T_c, respectively. We will involve now a new metric $\Phi(m,h)$ called traffic reduction factor. The metrics $\Phi(m,h)$ is defined as a ratio between the total communication time defined in our proposal and the total communication time defined in Ref. [4]. This metrics points to the percentage of decreasing the total communication times T_{ps} with respect to T_{ref}, in terms of number of messages m, and number of hops h, as parameters.

$$\Phi(m,h) = \frac{Tps}{Tref} = \frac{mh + h + \frac{Tt}{Tc}(m+h-1)}{2mh + \frac{Tt}{Tc}mh} \tag{6}$$

By substituting the values for T_t=499,13μs and T_c=153,33μs we define $T_t/T_c \approx 10/3$. Accordingly (6) we obtain:

$$\Phi(m,h) = \frac{Tps}{Tref} = \frac{3mh + 13h + 10m - 10}{16mh} \tag{7}$$

Fig. 5 sketches metric Φ in terms of m, with h as parameter. As can be seen from Fig.5 by increasing m and h, metric $\Phi(m,h)$ decreases what means that our proposal has better performance (from 17 % for m=10 and h=1, up to 66 % for m=10 and h=8).

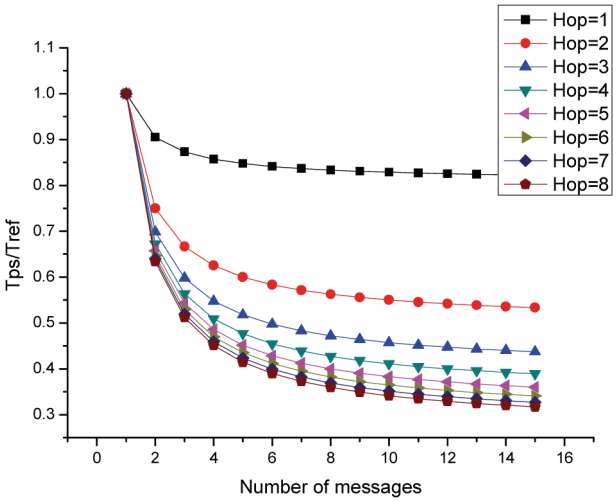


Fig. 5 Metric Φ(*m,h*) in terms of *m* with *h* as a parameter

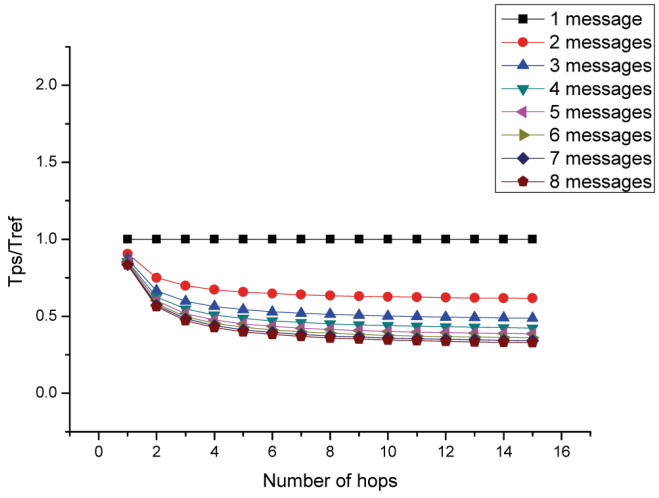
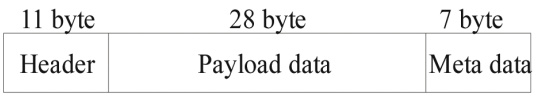


Fig.6 Metric Φ(*m,h*) in terms of *h* with *m* as a parameter

Fig. 6 presents metric Φ(*m,h*) in terms of *h*, with *m* as parameter. By analyzing Fig.6 we can conclude that for larger messages (m>2) metric Φ(*m,h*) decreases what implies that our proposal has better performance in respect to Ref.[4] (from 33 % for *m*=2 and *h*=4, up to 58 % for *m*=8 and *h*=4).

4.2 Sensor Node Lifetime Efficiency

As above mentioned, one of the most important design requirements related to SN operation deals with energy consumption. In order to increase the lifetime sensor node should operate with minimum possible energy consumption. Battery, as the main source of power supply in WSN, can only store a limited amount of energy. This requires power aware computation/communication component technology, low-energy signaling and networking, and power aware software communication. Lifetime refers to the period for which a SN is capable of sensing and transmitting the sensed data to the base station(s). In WSNs, thousands of nodes are powered with limited battery power budget. As a result, the lifetime analysis becomes an important aspect for efficient usage of the available energy. We will study the case of standard structures message_t (see Fig. 7), which is used to send packets in the most popular operating system for WSN, Tiny OS [17]. As Fig. 7 shows, it is a frame size of 46 bytes in length (11 bytes header, 28 bytes of payload data and 7 bytes of Meta data). We have already noted that the TCP / IP packet header size is of 48 bytes (24 bytes for TCP and 24 bytes for the IP protocol). By simple computation we get that we save 30 bytes per packet for our solution. This means that we minimize the total traffic between SNs for about 60 % (60.53 %). Having in mind that at first, a communication between SNs is the highest part of energy consumption in WSN, and second a multihop topology of WSN, this means that with our proposal we prolong the lifetime of each SNs for about 60 and more percent.



Header:		*Meta data*:	
lenght - data lenght	(1 byte)	**tx_power** - energy value	(1 byte)
fct - frame control field	(2 byte)	**rssi** - received signal strength	(1 byte)
dsn - data sequence number	(2 byte)	**lqi** - link quality indicator	(1 byte)
destpan - destination PAN ID	(1 byte)	**crc** - CRC code	(1 byte)
dest.addr. - destination address	(2 byte)	**ack** - ACK for each package	(1 byte)
sour.addr. - source address	(2 byte)	**time** - time stamp for each pac.	(1 byte)
type - packet type of data	(1 byte)	**rx_interval** - receive interval time	(1 byte)

Fig. 7 Structure of message_t from TinyOS

5 CONCLUSION

Until a few years ago, the end point of the Internet was a home computer or laptop. Nowadays, Internet capable devices are found in many other devices such as: mobile phones, internet radios, televisions, tablet computers and navigation systems. The number of devices with Internet connectivity will constantly grow in the next years. Standardizations and implementation of protocols like IPv6 confirm this trend. In its simplest form Internet compatible SNs will become an inevitable necessity. The problem of connectivity

between WSN and Internet is considered in this paper. This possibility allows us to access information from each SN in WSN Anywhere, Anytime, and Anything. This rule, known as rule 3A, is a basic condition for classic client-server system. This confirms that in the future it can be expected that every SN should become a standard client/server node, and therefore, all the principles of client/server communication will be established and applied in WSN. All of this will further extend the capabilities of WSN, and thus increase the number of applications in which they are used. In this paper, we have considered a possibility to connect a WSN to Internet. Bearing in mind that communications have a dominant effect on power consumption, we have proposed header length reduction. According to the obtained results, we conclude that energy efficiency achieved by involving shortage of TCP/IP header prolongs the lifetime of SN for 60 %, compared to the standard TCP/IP header used in Internet.

REFERENCES

1. Ian F.Akyildiz, Mehmet Can Vuran, *Wireless Sensor Networks*, ISBN 978-0-470-03601-3, WILEY, 2010
2. Y.Chung Wu, Q.Chandhari, E.Serpedin, *Clock Synchronization of Wireless Sensor Networks*, IEEE Signal Processing Magazine, Vol.28 No.1, January 2011, pp.124-138
3. Branislav Petrovic, Mile Stojcev, *Phase Measurements system based on embedded microcomputer,* Facta Universitatis, Series: Mechanical Engineering Vol. 1, No 10, 2003, pp. 1355-1368
4. S.Lei, W.Xiaoling, Xu Hui, Z.Jie, J.Cho, S.Lee, Connecting Heterogeneous Sensor Networks with IP Based Wire/WirelessNetworks, SEUS-WCCIA'06,2006
5. H.Dai, R.Han, Unifying Micro Sensor Networks with Internet via Overlay Networking, Proc.IEEE Emnets-1, Nov, 2004
6. K. Mayer, W.Fritsche, *IP-enabled Wireless Sensor Networks and their integration into the Internet*, http://portal.acm.org/ft_gateway.cfm?id=11426878&type=a5-mayer.pdf, acc. 10.01.2012
7. M.Zhang, S.Pack, K.Cho, D.Chang, Y.Choi, T.Kwon, *An Extensible Interworking Architecture (EIA) for Wireless Sensor Networks and Internet*, www.mmlab. snu.ac.kr/publications/docs/EIA_APNOM2006.pdf, acc. 12.12.2011
8. M.Zuniga, B.Krishnamachari, *Integrating Future Large-scale Wireless Sensor Networks with Internet*, www.cs.usc.edu/Research/techreports/papers/03-792.pdf, acc. 12.12.2011
9. Z.Z.Marco, K.Bhaskar, *Integrating Future Large-scale Wireless Sensor Networks with Internet*, USC Computer Science Technical Report CS 03-792, 2003
10. K.Mayer, W.Fritsche, *IP-enabled Wireless Sensor Networks and their integration into the Internet*, http://portal.acm.org/ft_gateway.cfm?id=11426878&type=a5-mayer.pdf.
11. M.Zhang, S.Pack, K.Cho, D.Chang, Y.Choi, T.Kwon, *An Extensible Interworking Architecture (EIA) for Wireless Sensor Networks and Internet*, www.mmlab.snu.ac.kr/publications/docs/EIA_APNOM2006. pdf.
12. Adam Dunkels, *Towards TCP/IP for Wireless Sensor Networks*, Malardalen University Licentiate Thesis No. 45, Swedish Institute of Computer Science, March 2005.
13. C.Intanagonwiwat, R.Govindan, D.Estrin, *Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks*, Proc.ACM MobiCom'00, Boston, MA, 2000, pp. 56-67
14. S.Lei, W.Xiaoling, Xu Hui, Z.Jie, J.Cho, S.Lee, Connecting Heterogeneous Sensor Networks with IP BasedWire/WirelessNetworks, SEUS-WCCIA'06,2006
15. M.Kosanović, M.Stojčev, *Implementation of TCP/IP Protocols in Wireless Sensor Networks*, ICEST 2007, Ohrid, Macedonia, June 2007
16. C.Westphal, *Layered IP Header Compression for IP-enabled Sensor Networks*, www.people.nokia.net /cedri c/Papers/icc06.pdf.
17. P.Levis, TEP 111, message_t, http://www.tinyos.net /tinyos-2.x/doc/html/tep111.html.

POVEZIVANJE BEŽIČNIH SENZORSKIH MREŽA NA INTERNET

Mirko R. Kosanović, Mile K. Stojčev

Bežične senzorske mreže (BSM) postale su tokom poslednje decenije jedno od najinteresantnijih, a verovatno i najviše istraživanih područja u oblasti elektronike. One predstavljaju kolekciju malih, jeftinih bežičnih senzorskih čvorova (SČ), koji su organizovani u klastere ili mreže i raspoređeni u širokom geografskom području, a sposobni su da potpuno samostalno kontinuirano prate ili mere pojave u prirodi, izvrše njihovo ažuriranje i iste pošalju bežičnom komunikacijom. Ali, BSM obično ne mogu da rade potpuno samostalno, u potpunoj izolaciji, odnosno one moraju biti povezane sa nekom drugom vrstom mreže (LAN, WAN). Kako je Internet postao de-facto standard za WAN mreže, postalo je neophodno povezati BSM sa Internetom i tako omogućiti da prikupljeni podaci budu vidljivi sa različitih mesta. U ovom radu, u početku, mi ćemo dati osnovne karakteristike BSM koje utiču na povezivanje sa Internetom, i analiziraćemo aktuelna poznata rešenja. Nakon toga razmotrićemo uslove koje moraju ispuniti BSM kako bi se povezale sa Internetom i predložićemo jedno jednostavno rešenje.

Ključne reči: *TCP/IP mreže, prikupljanje podataka, Internet, bežične senzorske mreže*