Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Computer Science 00 (2014) 000–000

Procedia Computer Science

www.elsevier.com/locate/procedia

The 5th International Conference on Emerging Ubiquitous Systems and Pervasive Networks (EUSPN-2014)

Congestion Detection Strategies in Wireless sensor Networks: A Comparative Study with Testbed Experiments

Mohamed Amine Kafi[a,b,*], Djamel Djenouri[b], Jalel Ben Othman[c], Abderaouf Ouadjaout[a,b], Nadjib Badache[a,b]

[a] *CERIST Research Center, Ben Aknoun, Algiers 16000, Algeria*
[b] *University of Sciences and Technology Houari-Boumediene (USTHB), Algiers, Algeria*
[c] *Laboratoire L2TI. Universite de Paris 13, Paris, France*

Abstract

Event based applications of Wireless Sensor Networks (WSNs) are prone to traffic congestion, where unpredicted event detection yields simultaneous generation of traffic at spatially co-related nodes, and its propagation towards the sink. This results in loss of information and waste energy. Early congestion detection is thus of high importance in such WSN applications to avoid the propagation of such a problem and to reduce its consequences. Different detection metrics are used in the congestion control literature. However, a comparative study that investigates the different metrics in real sensor motes environment is missing. This paper focuses on this issue and compares some detection metrics in a testbed network with MICAz motes. Results show the effectiveness of each method in different scenarios and concludes that the combination of buffer length and channel load constitute the better candidate for early and fictive detection.

Keywords: Wireless sensor networks; transport protocols; congestion control; Congestion Detection.

1. Introduction

A Wireless sensor network (WSN) is a set of tiny wireless devices deployed in a large geographical area to sense different physical events and to monitor the surrounding environment. WSN can be used in many applications, such as industry production, environment monitoring, home automation and health-care. Although light traffic features typical WSN's applications, congestion may affect a WSN in some cases. For example, a sudden detection of an important event in an event-based application may generate a bulk of voluminous traffic that will overload the network. This situation leads to the drop of data packets with potentially important information, as well as the waste of the scarce energy. The energy is consumed herein due to the re-transmission of packets lost in collisions [1].

[*] Corresponding author. Tel.: +213-553-195-197 ; fax:+213-21-91-21-26
E-mail address: kafiamine@gmail.com

To avoid the above aforementioned problem while ensuring the application fidelity, a relevant congestion control protocol should be used. Generally speaking, any congestion control mechanism follows three essential steps. First and foremost, i) reliable detection, ii) notification, iii) and taking appropriate decisions (control)[1]. The reliable detection component is the key element to achieve effective congestion control.

Most of previous studies comparing congestion detection methods are limited to simulation, with a high level of abstraction neglecting real-world aspects[2,3]. In[2], a simulation comparative study was done for multimedia sensor networks. It concluded that delay was the best parameter. While in[3], a simulation comparative study for IPV6 compliance purpose is presented, which concluded that buffer size was the best one. The only experimentation based study was done in[4], but it has compared only buffer length and channel load.

In this work, we compare different methods used in the literature to show their efficiency in early congestion detection. The comparison considers real scenario and uses real motes (testbed) instead of simulation. The remainder of the paper is organized as follows. Section 2 describes the pertinent metrics used in WSN congestion control literature. Next, Section 3 presents the evaluation and comparison results. Finally, the paper is concluded in Section 4.

2. Congestion Detection Strategies

In literature, many congestion detection mechanisms are used and tested. The most common detection methods are: packet loss, queue length, channel load, packet service time, and transmission delay.

In many cases, a single parameter cannot accurately indicate a congestion[1]. The selection of such a parameter should be related to some factors such as the network structure, application and traffic nature, used rate, etc...[2]. In the following, the most used parameters are presented.

2.1. Packet loss

The existing solutions measure this metric either at the sender or the receiver. It is measured at the sender by enabling the use of ACKs (Acknowledgements), whereas at the receiver through sequence numbers use. Further, not overhearing the parent's forwarding on the upstream link by a child node over the downstream link, can be considered as an indication for packet loss[5]. The time to repair losses (if reliability ensured) is used in[6], while loss ratio is used in[7,8]. The main drawback of this metric is that the losses can be caused by wireless errors rather than packets collision. Moreover, the packet reliability is not essential for some sensor applications, such as those using in-network data aggregation techniques[9].

2.2. Queue Length

As every node has a buffer (queue); its length (size) can serve as a simple and good indication of congestion. The buffer size can be used as a threshold, like in[10,11] (a fixed threshold is used and the congestion is signalled as soon as the buffer length exceeds this threshold), or periodically[2] (The buffer size is tested at the beginning of each period and the congestion is signalled at this moment). The remaining buffer length out of the overall size, or the difference between the remaining buffer and the traffic rate can be used as congestion indication as well. If the link layer applies retransmissions, link contention will be reflected through buffer length.

2.3. Channel Load

It measures the channel activity caused by wireless transmissions. For example, the CC2420 radio offers the CCA function which responds with the value 1 if the channel is occupied, or 0 if the channel is empty. The frequency of busyness returned by the sampling of this function, reflects the level of occupation of the wireless channel.

Channel busyness ratio or channel load is the ratio of time intervals when the channel is busy (successful transmission or collision) to the total time. In case of increase in packets collision, and after several unsuccessful MAC

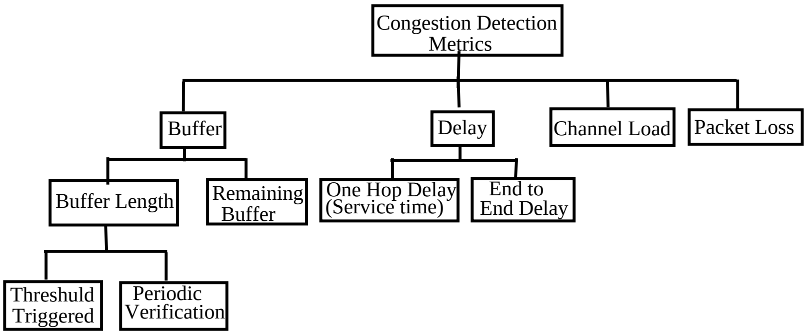


Fig. 1. Classification of congestion detection metrics

(Medium Access Control) transmissions, packets are removed. Consequently, the decrease in buffer occupancy due to these drops may mislead to the inference of the absence of congestion. This when only the buffer state is used for congestion detection. Therefore, for accurate congestion detection, a hybrid approach by using queue length and channel load as a congestion indication is more appropriate in this case[4].

2.4. Delay

It generally quantifies the necessary time since the packet generation, at the sender, until its successful reception at the next hop receiver[12], or end point receiver[13]. It can also be calculated as a part of the total delay, as in ATP[14] (queuing delay).

The one hop delay can be seen also as the packet Service Time, which is the time separating packet arrival at the MAC layer and its successful transmission, which is inversely proportional to the packet service rate. It covers packet waiting time, collision resolution, and packet transmission time at the MAC layer[15]. This value changes according to the queue length and channel load. It can be regarded as another measure of them. In[13], the end-to-end delay is calculated in a similar manner. But limitation to merely the service time may be misleading when the incoming traffic is not higher than the outgoing one through the overloaded channel.

Another delay measurement is that of the ratio of packet service time and packet inter-arrival time (scheduling time). A scheduler between the network and MAC layer switches the packets from network queues to the MAC layer. The scheduling time quantifies the number of packets scheduled per time unit. This ratio indicates both node level and link level congestion[16].

However, the delay may be misleading in some cases, when the largest amount of delay is caused by the sleep latency due to the use of duty-cycling at the MAC layer[17].

In Fig.1 a summary of detection metrics is highlighted, while in table 1, a summary of works with the used detection metrics is presented.

To our knowledge, comparing the previous congestion detection strategies in real world scenarios (using real motes) is missing in the current literature. This is the principal motivation of this work.

3. Experimentation and Comparison

In the following real tests, we aim to create network congestion and investigate the evolution of each parameter in time. The parameter that responds quickly and accurately is considered the best one. The cost of each metric is out of the scope of this study; but we provide some qualitative analyses about the evolved overhead.

3.1. Experimentation Environment

We have used our lab testbed that contains more than ten micaz nodes placed as highlighted in Fig. 2. In the testbed, the sensor nodes are plugged to an Ethernet network interfaces, in order to pick out results in log files that are

Table 1. Summary of works using detection metrics

Related works	Detection metrics							Remarks
	Buffer			Delay		Channel load	Packet loss	
	Buffer length		Remaining buffer	One hop delay	End to end delay			
	Threshold triggered	Periodic						
5							×	Not overhearing packets forwarding
6							×	Time to repair loss
7,8							×	Loss ratio
10,11	×							
2		×						
4	×					×		Combination: buffer and channel
15,12				×				Service time
13					×			
16				×				Service time / inter-arrival time
14				×				Queueing delay

used for plotting behaviour of each mote's congestion detection parameters. The TinyOs[18] is used as the operating system.

In the experiments, the following metrics are compared: queue length, channel load, success reception ratio, packet inter-arrival time/ service time, and finally one hop delay which represents the service time,too.

To construct network topology depicted in Fig. 2, every node detects its neighbors by using a simple hello messages. The messages are gathered from sensor nodes to the base-station by using CTP (collection Tree Protocol)[19].

Five senders have been chosen, namely nodes, 0, 1, 3, 6 and 12. Node 9 was chosen to forward the senders' packets towards the sink (node 5) and it disposes a buffer size of 20 packets. Node 2 is used as the channel load prober for node, 9. This is because node, 9, cannot probe continuously at the same time of receiving other nodes' packets and forwarding them. Remind that in the testbed's topology, node 2 and 9 are in the vicinity of the same senders. Node 2 probes continuously the physical channel, and every 100 milliseconds, the counter value is logged and reset to 0.

3.2. Experimentation Results

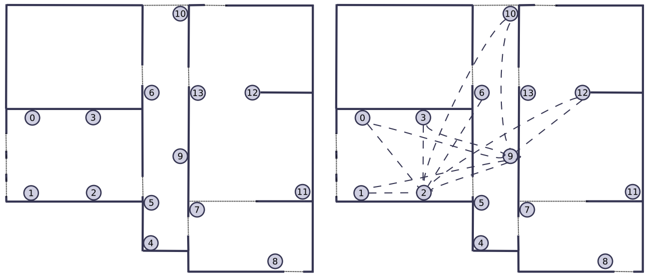
3.2.1. Channel Load Experiments

The goal of our first experiment is to investigate the channel load probing. The five senders, nodes 0, 1, 3, 6 and 12 start their transmissions at different time, respectively at $0sec$, $3sec$, $6sec$, $9sec$, and $12sec$. Moreover, to show the impact of different rates on the channel load, the experiments are repeated for different rates, 100 packets/s, 40 packets/s, 20 packets/s and 10 packets/s. Fig. 3 depicts the channel load, where the amplitude of channel load changes according to the number of senders and to the used rates. It converges to values around 100 for 10packet/s (with some sporadic picks), 200, for 20 packets/s, 300, for 40 packets/s, and 700 for 100 packets/s.

3.2.2. Congestion Detection Metrics experiments

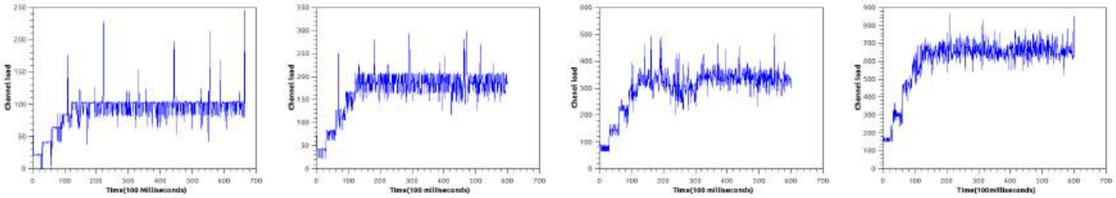
To show the effectiveness of the previous detection methods, namely buffer length, channel load, success ratio, one hop delay (service time) and packet inter arrival time/packet service time, we have conducted two different scenarios regarding sending rates. The first one shows the effectiveness of the previous metrics in low rate sending applications (less than 10 packets/s), while the second one highlights the congestion detection strategies in high rate monitoring applications (more than 40 packets/s). In the two rate scenarios, the rate given to the forwarder may be adequate if it takes into account the sum of senders rates, or blind if it does not take into account the senders rates.

Obviously, with low rate applications that take into account the sending rates to give appropriate rate to the forwarder, no congestion will take place, so no need to show this scenario by real test. On the other hand, if the forwarder rate is less than the sum of the senders, the congestion happens, and this scenario is chosen in our experiments.



(a) Sensor nodes deployment in the test-bed (b) Physical links in the test-bed

Fig. 2. The nodes positions and network topology in the test-bed



(a) Transmitting rate for 10 packets/s (b) Transmitting rate for 20 packets/s (c) Transmitting rate for 40 packets/s (d) Transmitting rate for 100 packets/s

Fig. 3. The channel load against transmitting rate

Whereas for the high rate applications, when the forwarder rate is less than the sum of the sending nodes,congestion will explicitly take place, and no need to show the tests results. We will than interest only to the case where the forwarder rate is sufficient to transmit senders' packets.

In Fig.4, we show the reaction of the previous metrics regarding different low rates, namely $1, 4, 6$ packets/s. In these tests, the forwarding rate is less than the sum of its sending nodes' rates to create congestion. The sender nodes start transmitting with a delay of 5 seconds between each other.

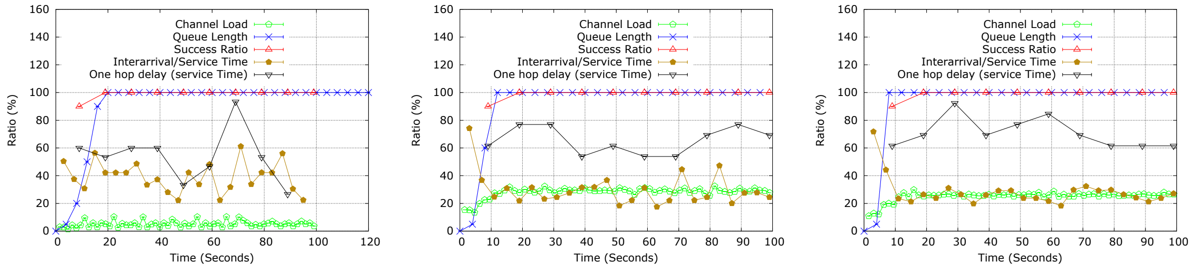
For high rate applications, the congestion is created even if the forwarder node is attributed a rate more than the sum of the sending nodes' rates, so we omit to show the reaction when the forwarding rate is less than the sum of sending nodes as it will lead certainly to congestion.

Any application has to use a MAC layer protocol, for example CSMA based protocol, as in our test. After many experiments using high rate transmissions, we concluded that with channel probing at the MAC layer of interfering nodes before sending a packet, known as CCA, a high number of packets can reach the forwarder, certainly with also a high number of losses and high channel loading. Fig. 5 (left side) shows the scenario of this experiment where nodes start incrementally sending at 50 packets/s, with a 3 seconds delay between their starting time, and using 20 packets as maximum buffer size. Fig. 6 depicts the detection metrics behaviour.

The goal of the next experiment is to show the effect of hidden terminals at the sending nodes. As the nodes of our testbed are in the interfering range of each other, we de-activated the CCA at the sending moment for the sending nodes to emulate the hidden terminal scenario. The only node being aware of that is the forwarder node, as it hears from all the nodes. Fig. 5 (right side) shows the nodes used for the experiment. The nodes start sending at 40 packets/s with a lag of 10 seconds. The behaviour of the different metrics are depicted in Fig. 7.

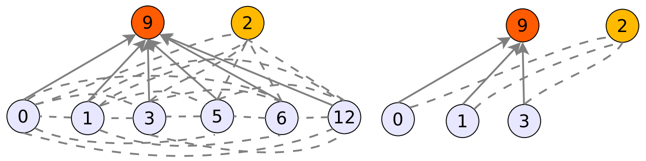
Buffer length behaviour: In the low rate scenario, buffer length depicts the better metric to reflect the forwarder congestion and its maximum value is reached in relation with the sending rates, namely 20 sec for 1 packets/s, 12 sec for 3 packets/s, and 8 sec for 5 packets/s.

The same remark can be given for the high rate transmitting scenario where the interfering nodes are aware of the



(a) Transmitting rate for 1 packets/s (b) Transmitting rate for 3 packets/s (c) Transmitting rate for 5 packets/s

Fig. 4. The metrics reaction for different low transmitting rates



(a) Sensor nodes Scenario with interfering nodes (b) Sensor nodes in a Hidden terminal Scenario

Fig. 5. The Sender nodes in two different scenarios

interferences. This is shown again by the buffer size that reaches its maximum value (20 packets). On the other hand, in the scenario where hidden nodes send their packets, the forwarder could not receive all packets, as it is reflected by the non full buffer. We concludes by this experiment that buffer length could not be aware about collisions at the receiver side. The only detection in this case may be done at the sender side, if packets will not be removed from its buffer until being acknowledged. This will lead to buffer overflow at the sender side.

Channel Load behaviour: In the low rate scenario, the value of channel load is quite petty and does not reflect the buffer overflow. We can explain this by the fact of infrequent transmissions that lead to no busy channel activity. On the other hand when enabling high rate transmissions (in the two high rate scenarios), the channel load presents in this scenario the more early and accurate detection metric, and it depicts the exact level of collisions.

Success delivery ratio: In the case of low rate transmissions, success delivery ratio is perfect because all packets arrive to the forwarder, even they will be dropped after due to the buffer overflow, but this is transparent for the success ratio metric which is not adequate in this scenario.
Whereas in high rate scenarios, the success ratio decreases due the fact of losses caused by repeated collisions. This observation makes the success ratio a good indication of collisions in the two high rate scenarios.

The service time: From the Fig.4, which depicts low rate metrics reactions, service time shows a lot of fluctuation. This can be explained by the fact that service time includes the channel hearing before transmission, which is so low in this case because of the channel freeness, and a random back-off (elapse of time before transmission) between 2 to 15 milliseconds, which justifies the fluctuations in this metric.

For high rate scenarios, service time is more accurate than in the scenario of low rates. This can be explained that channel sampling before sending consumes a part of time, which diminishes the fluctuation of service time and reflects congestion.

The ratio of inter arrival time and service time: even it is the more accurate metric after buffer length in low rate scenario, but presents many fluctuations, too. This can be explained that inter arrival time has a relation to the number of senders' packets (which is reflected on the buffer), but being divided by the service time makes it losses

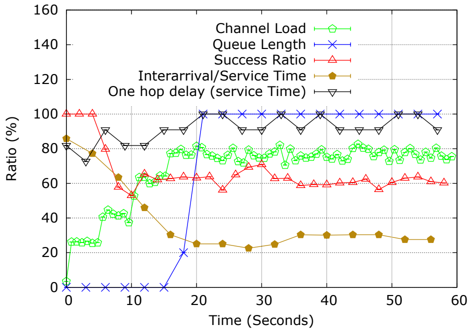


Fig. 6. Interfering nodes Scenario.

some accuracy. For high rate scenarios, ratio of inter arrival time and service time shows the congestion, but with a less efficiency than the channel load, which reflect the rate and number of senders.

3.3. Analyzes

Thorough the extensive real experiments conducted with the testbed scenarios, we have learned the following lessons:

For low rate scenarios, a sender node detects an acceptable channel load when attempting a transmission, but the lack of forwarding rate organization may lead to buffer overflow at the receivers. In this scenario, buffer size is a good indication of congestion, but not the channel load, except in extreme cases of dense interfering nodes. The success receiving ratio does not detect congestion, too. The other presented metrics, namely one hop delay (service time) and the ratio of inter arrival time and service time show fluctuations, which make them not suitable candidates.

In a high rate application scenarios using CSMA-based transmissions, the collisions probability is augmented. Therefore, up transmission failure, both the transmitter and the receiver buffers cannot detect a congestion in the absence of an acknowledgement (ACK) mechanism. Channel load will reflect this type of congestion as it will detect this collision.

If ACKs are used, the sender will know about the congestion as it will not remove the packet until receiving its ACK. This leads buffer overflow. But this is performed slowly compared to channel load. In the two cases, one hop delay (service time), success receiving ratio, ratio between inter arrival time and service time allow for congestion detection but in slow and less efficient manner.

Concerning the inter arrival time/ service time parameter, they are the consequence of other parameters. Service time reflects in part the channel load, while inter-arrival packet time reflects the buffer state. But the ratio of these two parameters does not reflect congestion efficiently.

The success receiving ratio is not an efficient metric, even it may be used for some high rate scenarios, for the cause that it can not detect congestion caused by buffer overflow after the well reception of packets. Moreover, data aggregation strategies make no sense for its use[9].

The combination of the channel load and buffer size is more meaningful and leads to earlier congestion detection as buffer length detects carefully the congestion in low rate scenarios, whereas channel load detects it in high rate scenarios. The use of channel load sampling must be done in effective manner at the sending moment to consume little energy[4]. Table 2 summarises the previous discussion.

4. Conclusion

In this paper, we have compared different metrics of congestion detection widely used in literature, namely buffer length, channel load, success ratio, one hop delay (service time) and ratio between inter arrival time and service time. We have shown through the different real tests their effectiveness for early congestion detection. Through the different scenarios using different transmission rates and interference ranges, we conclude that the combination of buffer length and channel load is the best alternative for early detection in all possible scenarios of congestions.

Table 2. Summary of the early detection behavior for each metric.

	High rate application (with no reliability)	**High rate application (with reliability)**	**Low rate application**
Buffer size	+	++	+++
Channel load	+++	+++	+
delay	++	++	++
Buffer size+ channel load	+++	+++	+++
Packet service time/packet inter arrival time	++	++	++

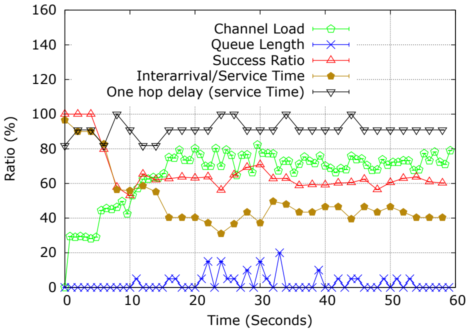


Fig. 7. Hidden terminals Scenario.

References

1. Kafi, M.A., Djenouri, D., Ben-Othman, J., Badache, N.. Congestion Control Protocols in Wireless Sensor Networks: A Survey. *IEEE Communications Surveys & Tutorials, accepted for publication* 2014;.
2. Sheikhi, H., Dashti, M., Dehghan, M.. Congestion detection for video traffic in wireless sensor networks. In: *2011 International Conference on Consumer Electronics Communications and Networks (CECNet)*. ????, .
3. Michopoulos, V., Guan, L., Oikonomou, G., Phillips, I.. A Comparative Study of Congestion Control Algorithms in IPv6 Wireless Sensor Networks. In: *International Conference on Distributed Computing in Sensor Systems and Workshops (DCOSS), 2011*. ????, .
4. Wan, C.Y., Eisenman, S.B., Campbell, A.T.. Energy-efficient congestion detection and avoidance in sensor networks. *ACM Trans Sen Netw* 2011;**7**(4):1–31.
5. Woo, A., Culler, D.E.. A Transmission Control Scheme for Media Access in Sensor Networks. In: *Proc. of ACM Mobicom*. 2001, p. 221–235.
6. Paek, J., Govindan, R.. RCRT: Rate-controlled reliable transport protocol for wireless sensor networks. *ACM Trans Sen Netw* 2010; **7**(3):20:1–20:45.
7. Zhou, Y., Lyu, M., Liu, J., Wang, H.. PORT: a price-oriented reliable transport protocol for wireless sensor networks. In: *Proc. of 16th IEEE International Symposium on Software Reliability Engineering, ISSRE*. 2005, p. 10 pp.–126.
8. Bian, F., Rangwala, S., Govindan, R.. Quasi-static Centralized Rate Allocation for Sensor Networks. In: *Proc. of 4th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks, SECON*. 2007, p. 361–370.
9. Bagaa, M., Challal, Y., Ksentini, A., Derhab, A., Badache, N.. Data Aggregation Scheduling Algorithms in Wireless Sensor Networks: Solutions and Challenges. *Communications Surveys Tutorials, IEEE* 2014;.
10. Hull, B., Jamieson, K., Balakrishnan, H.. Mitigating congestion in wireless sensor networks. In: *Proc. of the 2nd international conference on Embedded networked sensor systems, SenSys*. 2004, p. 134–147.
11. Zgr, Y.S., Sankarasubramaniam, Y., Akan, Ö.B., Akyildiz, I.F.. ESRT: Event-to-Sink Reliable Transport in Wireless Sensor Networks. In: *Proc. 4th ACM international symposium on Mobile ad hoc networking and computing, MobiHoc*. 2003, p. 177–188.
12. He, T., Stankovic, J.A., Lu, C., Abdelzaher, T.. SPEED: A Stateless Protocol for Real-Time Communication in Sensor Networks. In: *Proc. of IEEE 23rd International Conference on Distributed Computing Systems*. 2003, .
13. Sharif, A., Potdar, V., Rathnayaka, A.J.D.. Prioritizing Information for Achieving QoS Control in WSN. In: *Proc. of 24th IEEE International Conference on Advanced Information Networking and Applications (AINA)*. 2010, p. 835–842.
14. Sundaresan, K., Anantharaman, V., Hsieh, H.Y., Sivakumar, R.. ATP: A Reliable Transport Protocol for Ad Hoc Networks. *IEEE Transactions on Mobile Computing* 2005;**4**(6):588–603.
15. Ee, C.T., Bajcsy, R.. Congestion control and fairness for many-to-one routing in sensor networks. In: *ACM, SenSys*. 2004, p. 148–161.
16. Wang, C., Li, B., Sohraby, K., Daneshmand, M., Hu, Y.. Upstream congestion control in wireless sensor networks through cross-layer optimization. *IEEE Journal on Selected Areas in Communications* 2007;**25**(4):786–795.
17. Doudou, M., Djenouri, D., Badache, N.. Survey on Latency Issues of Asynchronous MAC Protocols in Delay-Sensitive Wireless Sensor Networks. *IEEE Communications Surveys and Tutorials* 2013;**15**(2):528–550.
18. TinyOS 2.x. http://www.tinyos.net. ????, .
19. Gnawali, O., Fonseca, R., Jamieson, K., Moss, D., Levis, P.. Collection Tree Protocol. In: *Sensys*. 2009, .