Experimental Characterization of Communication Latencies in Wireless Sensor Networks

A. Ageev, D. Macii, D. Petri

DISI – University of Trento, Via Sommarive 14, 38100, Trento,
Ph. +39 0461881571, Fax: +39 0461882093, e-mails: ageev@disi.unitn.it, macii@disi.unitn.it,
petri@disi.unitn.it

Abstract **-** The communication latency between Wireless Sensor Network (WSN) nodes can significantly affect the performance of real-time monitoring applications, especially when multi-hop network topologies are used. In fact, the end-to-end network delays may hinder the collaboration between different devices, thus preventing the applicability of data fusion algorithms or limiting the accuracy of inter-node time synchronization. Unfortunately, performing a thorough characterization of such latencies is not an easy task because they may depend considerably on the overall network data traffic as well as on the typical vagaries of RF links. In order to have a deeper insight about the communication latencies in nowadays WSNs, in this paper a suitable measurement procedure is described and some experimental results for different packet sizes, traffic conditions and number of hops are reported.

I. Introduction

Estimating the communication latencies in wired or wireless networks is a well-known measurement issue that has been extensively studied in last years [1]-[3]. In wireless networks this problem is even more critical due to the poor robustness that usually affects RF connections. In wireless sensor networks (WSNs) the contributions to the communication latency have been estimated mostly for synchronization purposes [4]-[6]. Nonetheless, just little experimental data are available in literature, although quite a deep analysis of the main uncertainty sources affecting time synchronization has been performed in [7]-[8].

Typically, a WSN node consists of a sensor board, a battery, a memory chip, a radio transceiver and a microcontroller (MCU) clocked either by an internal or an external oscillator. Although the MCU timers are commonly used to estimate the communication latencies (e.g. by measuring the round-trip time [9]), the uncertainty associated with such measurements is quite large due to the limited resolution of the timers, to the moderate stability of the local on-board oscillators and to the additional delays of the software routines reading the timers' content both at the transmitting and at the receiving ends. Alternatively, the inter-node communication latency can be estimated at a low level by measuring the time interval between the rising edge of a pulse generated by the transmitting node on some specific MCU pin just before starting the transmission of one packet, and the rising edge of the corresponding pulse generated by the receiving node on another MCU pin immediately after the packet is successfully received. In fact, since the MCU usually controls the whole communication process, it is possible to map that process onto the executed program and to introduce additional lines of code responsible for the pulse generation.

The increasing use of Medium Access Control (MAC) layers based on Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) protocols (such as those described in the well-known standard IEEE 802.15.4 [10]) makes the communication latency very sensitive to the traffic conditions of the network. As a consequence, when the channel is sensed busy or in case of packet collisions it is very difficult to estimate the actual end-to-end latency just using the MCU timers. In this paper, the problem of measuring the communication latency is addressed by means of an automated procedure based on a LabVIEW™ application. The collected results focus on the dependence of the communication latency on the payload size, network traffic conditions and number of hops.

II. Measurement Procedure

The basic components of the proposed experimental setup are shown in Figure 1 and include not only one sender-receiver pair of nodes, but also a *random traffic generator*, namely a node that is explicitly used to emulate a known amount of traffic within the network, as it will be explained better

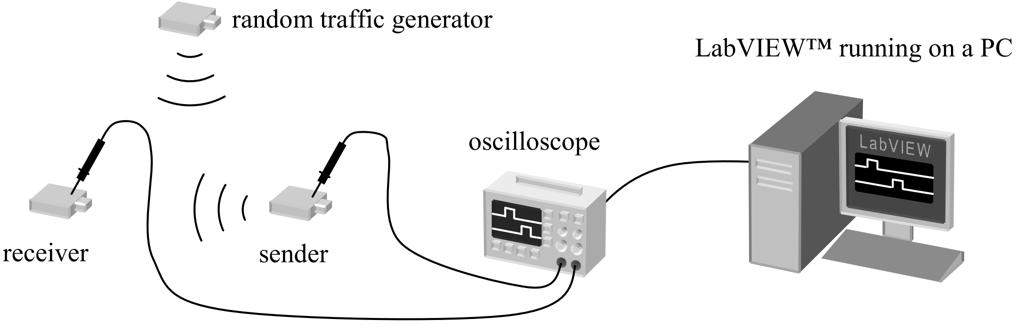


Figure 1. Experimental setup for measuring the inter-node communication latencies.

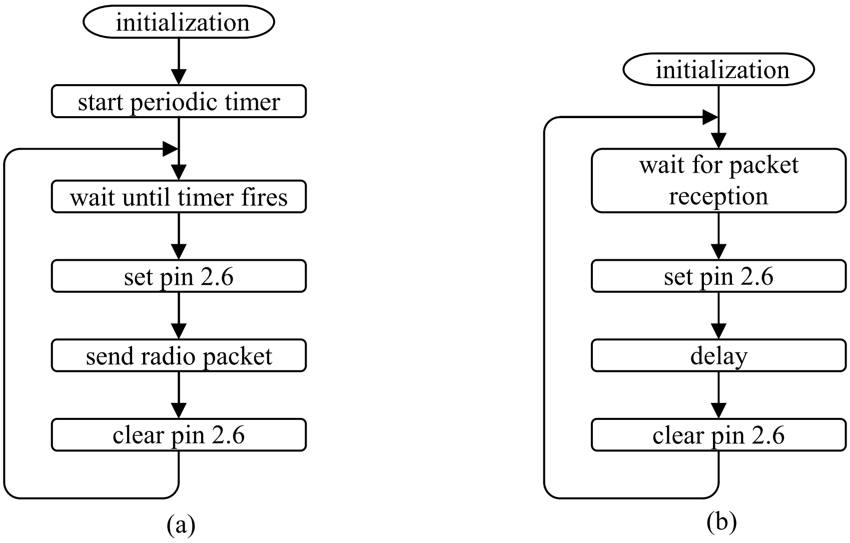


Figure 2. Flowcharts of the applications running on the sender (a) and receiver (b).

in the next section. All measurements were performed on TelosB nodes running the operating system TinyOS 2.0. Such nodes are equipped with an IEEE 802.15.4 compliant 2.4 GHz radio transceiver CC2420 [11], which is controlled by a microcontroller MSP430F1611 via an SPI interface [12]. The sender was programmed to send a radio packet to the receiver once a second. Several additional commands were inserted in the transmitter's code to generate a special pulse on the MCU pin P2.6. The rising edge of the generated pulse appears on this pin (as a result of setting the corresponding bit of the port register), just before running the sending commands. After those commands are executed, the falling edge of the pulse is produced by clearing the bit of the port register. The receiver was programmed to generate a similar pulse on its MCU pin P2.6 by suitably adding several commands to the receiver's program code. In this case, the appropriate bit of the port register is set to generate the rising edge of the pulse as soon as a radio packet is received and the bit is finally reset after 2 ms. The flowcharts of the applications running on the sender and receiver nodes are shown in Figure 2.

By connecting the MCU pins P2.6 of the transmitting and receiving nodes to the channels 1 and 2 of a digital oscilloscope TDS 3012, the total communication latency can be easily measured in different operating conditions using a LabVIEW™ application controlling the oscilloscope through an Ethernet interface. The LabVIEW™ application is specifically designed for measuring the time interval between the voltage pulses acquired on the two channels, with the positive edge on the first channel set as a triggering event. Eventually, the multiple measured time values are saved into a log file in order to be further processed through a MATLAB™ application. Consider that the instrumental uncertainty associated to such a measurement result is negligible, because the worst-case contributions affecting the time measurements using the chosen oscilloscope are several order of magnitude smaller than the interval under test. In fact, the delay time accuracy of the oscilloscope is ±200 ppm over any time interval longer than 1 ms [13]. Also, according to the MSP430F1611 manual [14], moving a constant to a certain destination address requires 5 CPU cycles. Therefore, since the microcontroller operates at the frequency of 8 MHz, the commands setting or clearing a port pin should take less than 1 μs.

III. Experiment Description

A. Communication Latency as a Function of Packet Length

In the first set of experiments one-hop communication latencies between two TelosB nodes have been measured for different payload sizes ranging from 10 to 100 bytes under negligible traffic conditions (i.e. when the random traffic generator is off). Due to 18 additional bytes added at the MAC and physical layers, the whole packet length is included between 28 and 118 bytes. One hundred measurements have been repeated for each packet size to evaluate the effect of random contributions. The mean δ_C and standard deviation values σ_C of the measurement results are shown in Figure 3(a) and 3(b), respectively, as functions of the payload size. Notice that the average communication delay increases almost linearly. In general, the slope of δ_C represents the average time to transfer 1 byte of data between two nodes. Notice that also in negligible traffic conditions the standard deviation of the communication latency may be quite large (in the order of some ms), but it does not depend on the packet payload size. In fact, this random behavior follows approximately a uniform distribution and it is due to the initial channel access delay which is commonly inserted at the MAC layer by any CSMA/CA protocol, to reduce the probability of packet collisions over the wireless channel.

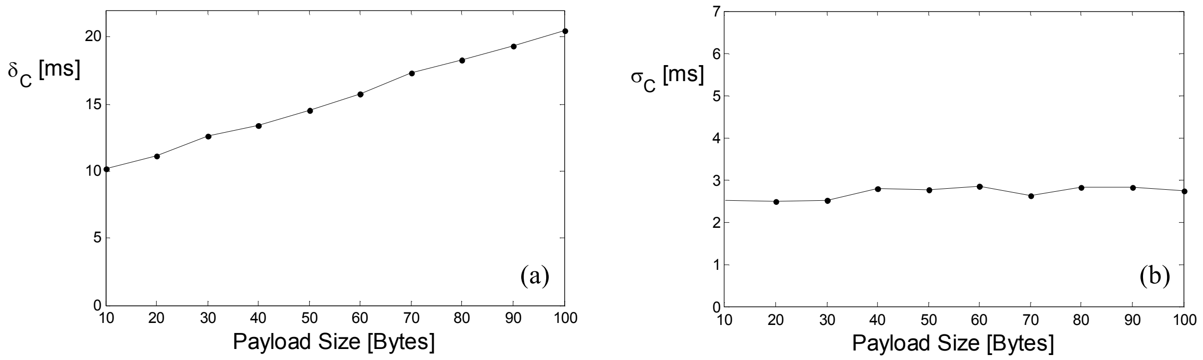
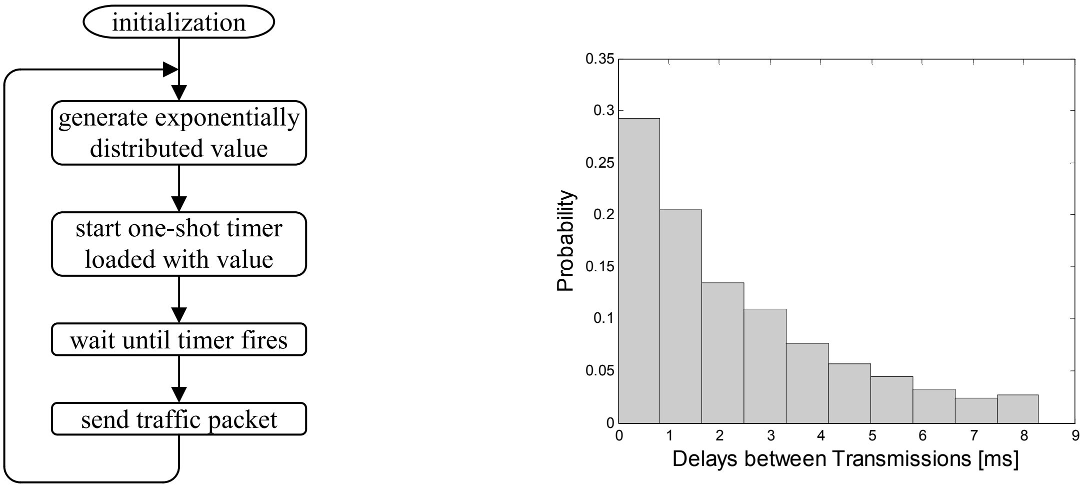


Figure 3. Mean (a) and standard deviation (b) values of 100 one-hop communication latencies between two TelosB nodes. The patterns are plotted as functions of the payload size in negligible traffic conditions.

B. Communication Latency as a Function of Different Traffic Conditions

In the second set of experiments the communication latencies between two TelosB modules have been measured in different network traffic conditions, after setting the payload size equal to 10 bytes. The experimental setup is the same as in the first set of experiments, but in this case the random traffic generator is on. The goal of the traffic generator here is to create radio traffic conditions similar to those in real wireless sensor networks containing a large number of nodes. Thus, whenever the sender tries to send one packet to the receiver, the communication time is affected by network traffic. The traffic generated in our experiments is approximately based on the model described in [15], where it is assumed that each node in the network transmits a radio packet whose duration is small compared to the time interval between successive transmissions. Moreover, by assuming that the intervals between consecutive transmission attempts in the whole network are exponentially distributed, the number of packets to be sent will follow a Poisson distribution (in the following referred to as *Poisson offered traffic* for brevity). This traffic is characterized by the *mean offered traffic rate G* representing the average number of packets which should be sent during one single packet transmission interval. The abovementioned traffic representation was used by researchers to model delay distributions in networks based on CSMA access mechanisms [16]. In particular, the offered traffic rate G is related to the mean value β of an exponential distribution with probability density function $f(t)=\frac{1}{\beta}e^{-\frac{t}{\beta}}$ for $t \geq 0$ by the following expression:

$$G=\frac{T}{\beta} \qquad (1)$$



(a) (b)

Figure 4. Flowchart of the application running on the traffic generator (a) and normalized histogram example resulting from 2000 values produced by the traffic generator when *G*=0.46 (b).

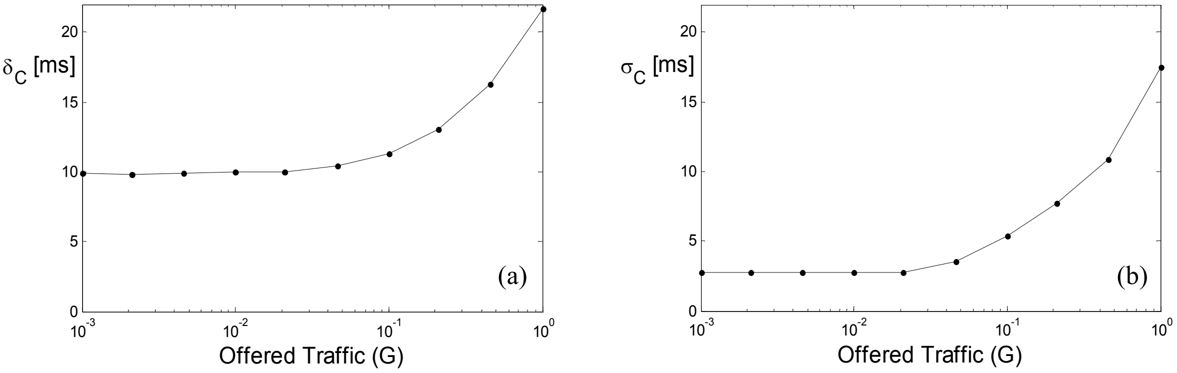


Figure 5. Average (a) and standard deviations values (b) estimated over 100 one-hop communication latencies between two TelosB nodes. The patterns are plotted as a function of various values of the offered traffic parameter *G* on a logarithmic scale.

where *T* is the given transmission time of a single packet. Starting from these assumptions, while the packet transmission between the sending and the receiving node occurs periodically once per second, the traffic generator has been programmed to generate an exponentially distributed random sequence of packets. The random intervals are generated by a routine updating the timer of the traffic generator. In this way, when the timer overflows, the corresponding timer interrupt service routine at first performs some radio sending commands, then it generates the next time interval to be loaded into the timer for the subsequent transmission. The flowchart of the application running on the traffic generator is shown in Figure 4(a). Two thousand consecutive values actually produced by the traffic generator after setting *G*=0.46 have been read from the node memory to verify the exponential nature of their distribution. Notice that the normalized histogram based on the collected values shown in Figure 4(b) exhibits quite a good exponential behavior. Naturally, an increment in traffic should significantly reduce the probability of achieving a successful transmission. In the performed experiments, the sender is able to retransmit each packet up to 6 times in case of collision and the receiving node replies with acknowledgement, when a packet is successfully received. This mechanism greatly improves the data transfer reliability, but it contributes (together with the CSMA-CA algorithm adopted on the TelosB nodes) to the communication delay increase when the network traffic grows. It is important to note that the CSMA-CA mechanism has been switched off in the traffic generator in order to disable any retransmission attempts that could alter the wanted Poisson traffic model. In fact, the data produced by the traffic generator are assumed to emulate already the cumulative effect of the packets' transmissions and retransmissions occurring in a possible large network. Thus, no additional random delays should be added to the exponentially distributed time intervals.

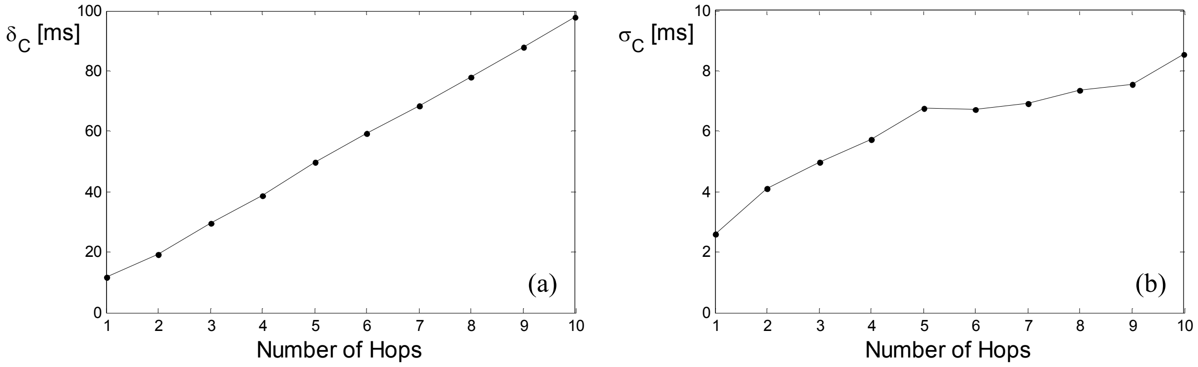


Figure 6. Mean values (a) and standard deviations (b) estimated over 100 communication latencies between two TelosB nodes for different numbers of hops in negligible traffic conditions (*G*≈0).

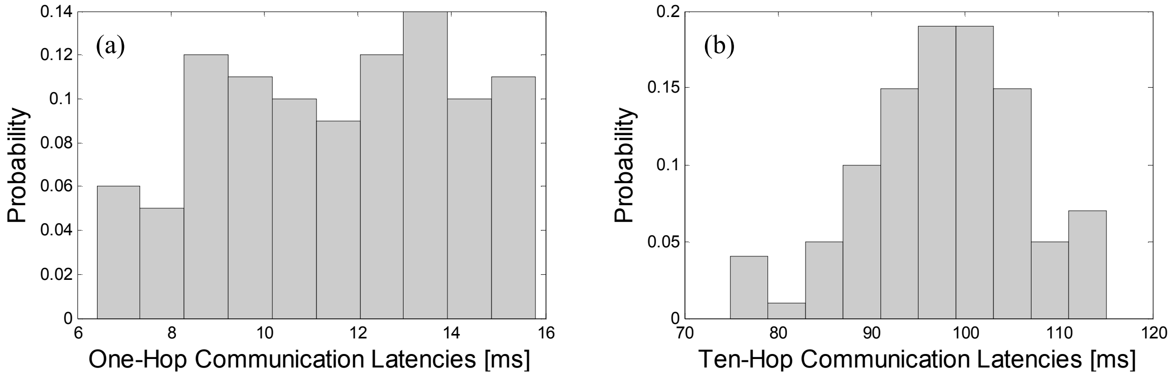


Figure 7. Normalized histograms of one-hop (a) and ten-hop (b) communication latencies estimated over 100 measured values.

In order to evaluate the effect of different amounts of traffic, the average communication latencies and the corresponding standard deviation values estimated over 100 measurement results are plotted in Figure 5(a) and 5(b), respectively, for different values of *G* (namely for *G* = 0.001, 0.0021, 0.0046, 0.01, 0.021, 0.046, 0.1, 0.21, 0.46, 1.0). This figure highlights the almost exponential growth of the average communication latency as a function of the offered traffic. Observe that also the standard deviation increases at a similar rate when the channel is congested (occasionally even more than 90 ms for *G* = 1.0) due to the overall larger number of retransmission attempts.

C. Communication Latency in Multi-hop Links

In the third set of experiments the communication latencies between two TelosB nodes have been measured for different number of hops under negligible traffic conditions (i.e., when the traffic generator is off), after setting the payload size equal to 10 bytes. The experimental setup is the same as in Figure 1, but in this case the sender communicates with the receiver through a variable number of intermediate "bridge" modules (ranging from 1 to 9). In the current settings, whenever a radio packet is received by an intermediate node, it is immediately forwarded to the closest nearby node. In this context, the total communication latency is the time interval between the rising edge of the pulse generated by the sender (as described in Section II) and the rising edge of the corresponding pulse produced by the final receiver.

In Figure 6 the average communication latency (a) and the respective standard deviation (b) values estimated over 100 measurement results are shown as a function of the number of hops in negligible traffic conditions (i.e., for *G*≈0). Notice that the average communication latency is in the order of some tens ms and it grows linearly, whereas the standard deviation pattern exhibits roughly a square root trend. Furthermore, while the single-hop communication latencies (in accordance with what stated in Section III.A) are approximately uniformly distributed as shown in the normalized histogram in Figure 7(a), in case of multiple hops the distribution tends to become Gaussian. This is clearly visible in the normalized histogram shown in Figure 7(b) in the 10-hops case. Such results confirm that, if all WSN nodes are nominally identical, also the various single-hop communication latencies can be assumed to be independent and identically distributed. As a consequence, the hypotheses of the central limit theorem (CLT) are met and the mean value and the variance of the resulting Gaussian distribution increase linearly with the number of hops.

IV. Conclusions

In this paper, an experimental procedure to measure the total communication latency between WSN nodes is described. Such a procedure has been applied to a small network of TelosB nodes in different operating conditions. In particular, one hundred communication latencies have been measured for different payload sizes, under different amount of network traffic and for different number of hops. In all cases the communication latencies have proved to be in the order of several ms. However, the random fluctuations due to intense traffic or multi-hop connections may become in the same order of magnitude or even larger, thus preventing the possibility of performing tightly coordinated actions between the network nodes. In authors' opinion, the obtained results provide interesting quantitative information to determine the maximum timing constraints that can be achieved in distributed real-time applications based on WSN devices.

References

[1] Jean-Chrysotome Bolot, "End-to-end packet delay and loss behavior in the internet," *ACM SIGCOMM Computer Communication Review*, Vol. 23, No. 4, pp. 289 – 298, Oct. 1993.

[2] E. Nunzi, "Uncertainties Analysis in RTT Network Measurements: the GUM and RFC Approaches," *Proc. IEEE International Workshop on Advanced Methods for Uncertainty Estimation in Measurement (AMUEM)*, Trento, Italy, 20-21 April, pp. 87 – 91, 2006.

[3] M. Zekauskas G. Almes, S. Kalidindi. RFC 2681. A Round–trip Delay Metric for IPPM, IETF, Sep. 1999.

[4] M. Maróti, B. Kusy, G. Simon, A. Lédeczi, "The Flooding Time Synchronization Protocol," *Proc. 2nd Int. Conf. on Embedded Networked Sensor Systems*, pp. 39-49, Baltimore, USA, Nov. 2004.

[5] B. Kusy, P. Dutta, P. Levis, M. Maróti, A. Lédeczi, and D. Culler, "Elapsed Time on Arrival: a Simple and Versatile Primitive for Canonical Time Synchronization Services," *International Journal of Ad Hoc and Ubiquitous Computing*, Vol. 1, No.4, pp. 239–251, 2006.

[6] S. Yoon, C. Veerarittiphan, M. L. Sichitiu, "Tiny-sync: Tight Time Synchronization for Wireless Sensor Networks," *ACM Journal of Sensor Networks*, Vol. 3, No. 2, Article 8, Jun. 2007.

[7] D. Macii, A. Ageev, D. Petri, "Modeling Time Synchronization Uncertainty Sources in Wireless Sensor Networks," Proc. *IEEE Int. Symp. on Advanced Methods for Uncertainty Estimation in Measurement* (AMUEM), pp. 98-103, Sardagna, Trento, Italy, Jul. 2007.

[8] A. Ageev, D. Macii, D. Petri, "Synchronization Uncertainty Contributions in Wireless Sensor Networks," Proc. *IEEE International Instrumentation and Measurement technology Conference* (I^2MTC), pp. 1986-1991, Victoria, Canada, 12-15 May 2008.

[9] C. Intanagonwiwat, R. Govidan, D. Estrin, "Directed Diffusion: a Scalable and Robust Communication Paradigm for Sensor Networks," Proc. Int. Conf. on Mobile Computing and Networks (MobiCom), pp. 56-67, Boston, ACM Press, Aug. 2000.

[10] IEEE Standard for Information Technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - specific requirement. Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs), IEEE, Sep. 2006.

[11] 2.4 GHz IEEE 802.15.4 / ZigBee-ready RF Transceiver, SWRS041B, Texas Instruments, Mar. 2007.

[12] MSP430x15x, MSP430x16x, MSP430x161x Mixed Signal Microcontroller, SLAS386E, Texas Instruments, Aug. 2006.

[13] TDS3000 Series Digital Phosphor Oscilloscopes 071-0382-01, Service Manual, Specifications 1-5, Tektronix.

[14] MSP430x1xx Family. User's Guide, SLAU094F, Texas Instrument, 2006.

[15] L. Kleinrock, F. A. Tobagi, "Packet Switching in Radio Channels: Part I-Carrier Sense Multiple-Access Modes and their Throughput-Delay Characteristics," IEEE Trans. on Communications, Vol. COM-23, pp. 1400–1416, Dec. 1975.

[16] Y. Yang, T. P. Yum, "Delay Distributions of Slotted ALOHA and CSMA," *IEEE Trans. on Communications*, Vol. 51, No. 11, pp. 1846- 1857, Nov. 2003.