Implementation and Performance Evaluation of nanoMAC: A Low-Power MAC Solution for High Density Wireless Sensor Networks

Junaid Ansari, Janne Riihijärvi and Petri Mähönen
Department of Wireless Networks
RWTH Aachen University
Kackertstrasse 9, D-52072 Aachen, Germany
Email: {jan, jar, pma}@mobnets.rwth-aachen.de

Jussi Haapola
Centre for Wireless Communications
P.O. Box 4500 FIN-90014
University of Oulu, Finland
Email: jhaapola@ee.oulu.fi

***Abstract*— This paper describes the implementation architecture and performance analysis of nanoMAC, a CSMA/CA based medium access control protocol, which is specifically designed for high density wireless sensor networks. We empirically show that nanoMAC performs with high reliability in a variety of network traffic conditions in single and multihop scenarios. For energy efficient operation and minimizing idle-overhearing, nanoMAC uses a specialized sleep algorithm. We also show results from a comparative study of nanoMAC with B-MAC in terms of performance measures.**

Keywords–Wireless Sensor MAC, Multihop communication, Reliability

I. Introduction

Wireless Sensor Networks (WSNs for short) have potentially a wide range of applications. Each of these applications has, of course, its own specific requirements and goals. However, some common requirements and constraints are encountered in practically every scenario. For example, the wireless sensor nodes typically have very limited memory, power resource and computational abilities. These constraints make reliable data transport over multiple hops very challenging and this has become a key area of research in WSNs.

With the advancements in radio design techniques, low power solutions are becoming a reality. However much is dependent on the way the medium access control (MAC) protocols behave. In WSN-MAC protocols, energy consumption and reliability with acceptable throughput is of primary concern. Wei Ye *et al.* [1] identified that the key sources of energy wastage are collisions, over-hearing, control-packet-overhead and idle listening to the wireless medium. We argue that reliable transmission and energy consumption are interlinked with each other. The higher the number of collisions, the higher the number of retransmissions, resulting in more energy consumption and also an added overall latency. Most common causes of collisions are false channel sensing and hidden terminals in a multihop communication environment.

NanoMAC [2,3] is a non-persistent carrier sense multiple access with collision avoidance (CSMA/CA) based MAC protocol explicitly designed for dense WSNs. It is intended to be highly scalable and it supports IEEE-addressing. The nanoMAC control frames: (RTS, CTS and ACK) are designed to achieve a high MPDU-to-packet ratio. The idle channel listening and over-hearing is reduced to a minimum by a sophisticated sleep algorithm employed in nanoMAC. The lightweight RTS/CTS handshake not only works to solve problems associated with hidden terminals, but it also announces the sleep-durations and governs the synchronization mechanism among the nodes in a cluster.

In this paper, we give the design and implementation details of nanoMAC on commercially available Telos motes [4] running TinyOS [5] in addition to extensive performance evaluation using measurements on these testbeds. We also conducted similar experiments on B-MAC using the same testbeds.

The rest of the paper is organized as follows: In section II, we give the state-of-the-art overview of the existing WSN MAC solutions. Section III gives details on the design and implementation of nanoMAC protocol. In section IV, we describe our experimental set-up and the results. In section V, these results are discussed in depth. Finally in section VI, we conclude the paper.

II. Related Work

Great deal of research has been going on in the field of power-aware MAC protocols for WSNs and several solutions have been proposed for minimizing battery usage. Deterministic protocols based on TDMA [6,7] appear to be promising owing to negligible number of collisions and low-power operation. However, such schemes suffer from scalability problems in large and dense WSNs and also the sub-optimal use of the channel.

Simple CSMA-based MAC protocols, while offering good performance in terms of channel utilization are lacking any kind of energy conservation mechanisms and thus perform quite poorly with respect to the goals of the WSN MAC protocol designs. We refer the reader to [8] for more details. Earlier schemes like PAMAS [14], which uses separate channel for sleep scheduling to packet overhearing are sub-optimal and inherently resource hungry. The reason is that it consumes extra resources for the signalling channel and also the idle-listening is not minimized.

Sensor-MAC (S-MAC) [1] is an IEEE 802.11 inspired CSMA-protocol, which uses RTS/CTS-handshake and addresses the problem of battery power conservation by periodic listening and waking-up. The fixed duty cycle with long active periods makes it inefficient in terms of battery usage. In a later paper [17], the authors also included adaptive listening and scheduling. However, the use of explicit SYNC-frames for global synchronization is an inefficient technique [15].

Timeout-MAC (T-MAC) [8] uses RTS-CTS-DATA-ACK exchange with adaptive duty cycle to save energy. It is an enhancement to S-MAC, which dynamically minimizes idle-listening and the wake-up durations.

Berkeley's MAC (B-MAC) is one of the most popular WSN-MAC protocols (being the default in modern TinyOS installations) and is also based on CSMA. By default, it does not use any control frames although support for specialized ACK frames is given as an option. The nodes periodically sense the channel. The transmitting node transmits a long preamble to let the receiving node sense it and therefore get in synchronization to receive the packet correctly. The duration of the transmitting node's preamble must be set long enough so that the periodic sensing of the channel may hear it at the receiver. In other words, the preamble length used should at least match the wake-up interval. The absence of control frames degrades the performance of B-MAC when hidden terminals are present in the network. This is verified by our measurements. Furthermore, the energy consumption is shifted to transmitter side if a node wishes to transfer large amount of data owing to the long preamble per packet in the beginning.

III. NanoMAC Protocol Design and Implementation

In this section, a brief overview of the nanoMAC protocol architecture and functionality is given. We also give the basic implementation details of nanoMAC for Telos wireless sensor nodes later in this section.

A. Overview

NanoMAC is a p-non-persistent (i.e. with probability p, it acts as a non-persistent protocol) CSMA/CA MAC protocol. The basic operation cycle is defined by RTS-CTS-nDATA-ACK handshake. The RTS and CTS frames consist of only 18 bytes and contain the information of the real IEEE-addresses of the transmitting and receiving nodes, sleep information and the number of data frames to be transmitted. With the RTS/CTS reservation, multiple data frames may be transmitted. The data frames only contain temporarily valid, short, randomly assigned addresses specified in RTS/CTS handshake for avoiding extra control overhead. The integrity of the transmission is assured by a single ACK-frame, indicating the data-frames that are not successfully received so that the transmitter needs only to re-send the corrupted frames later on. This insures better channel and bandwidth utilization.

In order to conserve the battery power and to minimize idle listening time, nanoMAC relies on its specialized sleep algorithm. Depending on the traffic density and application requirements, the nodes may operate in one of the four sleep groups. The assignment of the sleep groups are not controlled by nanoMAC itself, rather these are flexibly assigned by the application layer. Sleep Group 0 is the always-on mode. The nodes operating in sleep group 0 go to sleep state only when in contention or to avoid idle overhearing. The channel occupation interval is calculated from the sleep information in the RTS and CTS frames. Sleep group 0 obviously can give the maximum throughput and lowest latency. Nodes operating in sleep groups 1, 2 and 3 undergo regular sleep and wake cycles. The wake-up cycle period is fixed but is different in the sleep groups 1, 2 and 3 and is so designed that the nodes in different sleep groups can also communicate with each other.

The wake duration, called the active period, is very small and same for all the sleep groups. The active period is designed so that the nodes sleep most of the time and transmit burst data only in the active period. During the communication, the nodes extend their active periods to receive and transmit the packets successfully and immediately go to sleep mode afterwards. The nodes, however, follow the normal wake-up instants even if they have to extend their active periods owing to the reception/transmission of packets. Since all the nodes operating in a network are synchronized and a particular node can only transmit during its active period, the packet transmission is heard by all the nodes in the cluster.

When nanoMAC is given a new packet to transmit, it performs carrier sensing (CS) independent of the non-persistence parameter and the backoff counter value. If the channel is found to be busy, the nanoMAC performs a random backoff within a specified window and the node goes to the sleep state. The carrier sensing is performed for long enough duration to be sure of the presence of the carrier with high confidence.

For broadcast transmission, the transmitting node sends a broadcast RTS (BRTS) frame, followed by the data frames so that the receiving nodes extend their active periods in anticipation of the data frames. The receiving nodes do not acknowledge the data frames in broadcast transmission.

For further details on nanoMAC protocol architecture and functionality, we refer the reader to [2], [3] and [15].

B. Implementation Details

We implemented nanoMAC in nesC - a component based enhancement of ANSI-C for embedded environments - into the TinyOS framework. TinyOS is a lightweight, event driven operating system developed at the UC Berkeley for resource constrained embedded devices. The nanoMAC stack essentially consists of the module `TelosMacNanoM` and `TelosMacNanoC` providing interfaces for sending and receiving messages. It automatically handles the construction of control frames, fragmentation, congestion control, synchronization and sleep scheduling. The module `TelosMacNanoM` uses the `CC2420Control` interface to set/control the parameters of the CC2420 radio transceiver. The `HPLCC2420C` in-

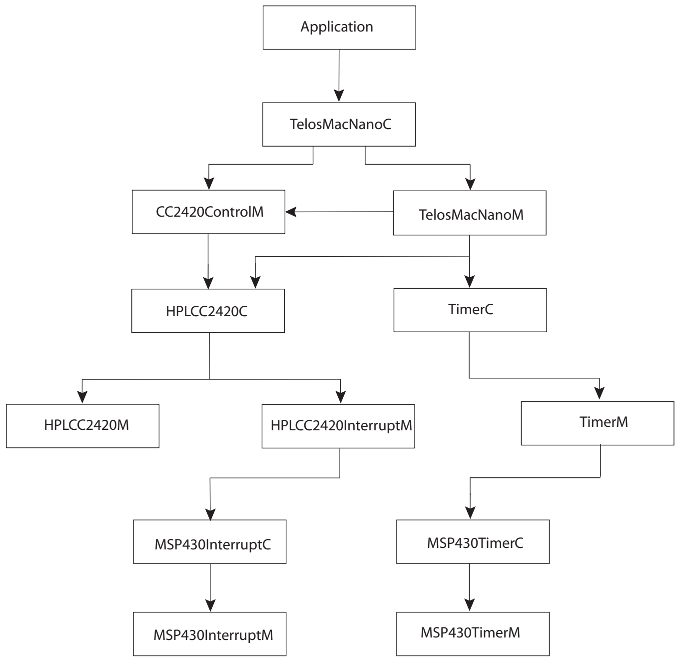

Fig. 1. Component diagram of the nanoMAC implementation.

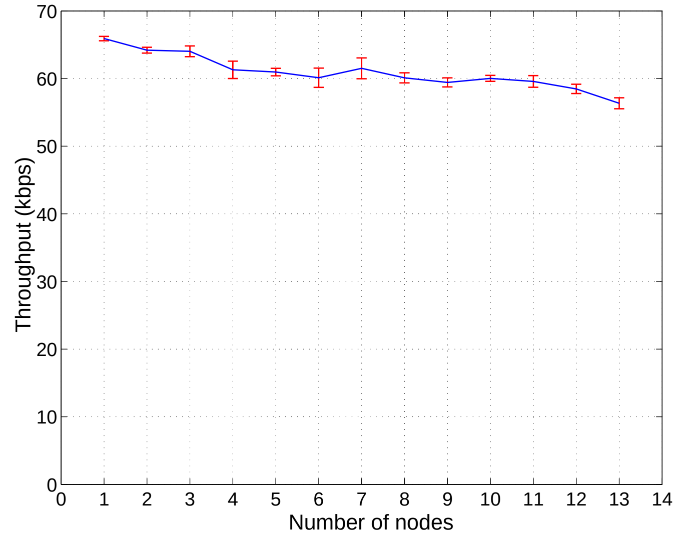

Fig. 2. Mean throughput and standard deviation of nanoMAC in a highly congested channel.

terface is used for hardware specific read/write and commands. The `HPLCC2420FIFO` interface is used to perform TX-FIFO and RX-FIFO specific read/write functions. The interfaces `HPLCC2420Capture` and `HPLCC2420Interrupt` are used for timer captures and handling interrupts, respectively. A simplified component level diagram of the nanoMAC implementation is shown in Fig. 1.

IV. Experimental Details

To evaluate the performance of nanoMAC, experiments were conducted to study and measure the throughput, latency, fairness and effects of hidden terminals. An overview of the testbeds used for the experiments and the performance measures are given in the following sub-sections.

A. Testbed

The testbeds used for the experimental performance evaluation are the 4th generation wireless sensor nodes (Telos motes), equipped with Texas Instrument's MSP430-series [10] ultra low-power 8 MHz microcontroller and Chipcon's CC2420 wireless transceiver [9]. The microcontroller has 10 kB of RAM and 48 kB of flash memory and it runs TinyOS from its internal flash. Telos motes give a radio data rate of 250 kbps in 2.4 GHz ISM band and are compliant with the IEEE 802.15.4 standard. Since 2.4 GHz ISM band is also commonly used by IEEE 802.11 WLANs, channel selection was done in a way to avoid the mutual radio interference. We used the USB interface for sending various command messages for controlling the experiments and for a comprehensive data logging.

B. Throughput

Throughput is a classical metric for MAC protocols. A high throughput with efficient channel utilization is desirable for applications where bulk data transfer is required [11]. In order to observe the throughput behaviour, a receiving node was placed at the centre of a circle, while n nodes were placed on the circular boundary. The nodes were within the perfect wireless range of each other. Each of the transmitting nodes was generating a high enough traffic load to saturate the channel. Throughput results for the nanoMAC with 15 frames per packet (operating in sleep group 0) are shown in Fig. 2. It is observed that nanoMAC's throughput degrades only slightly with the addition of nodes contending for the same receiver even in a highly saturated environment. To observe the behaviour of nanoMAC more closely, one of the randomly selected nodes, called a "test node" in this section, was closely profiled. The number of times it sent RTS-frames and received CTS and ACK frames was recorded. The results are shown in Fig. 3. The ACK to CTS ratio implies that the packet success ratio remains remarkably high and the data frames suffer very few collisions. This also suggests that energy is not wasted in the retransmissions due to collisions. For comparison, the packet success ratio for B-MAC with different packet sizes was measured using the same experimental set-up. In order to generate as high as possible traffic to saturate the channel, the `send` function was called in the `sendDone` event. The number of packets actually sent over the radio at the test node were counted when the `sendDone` event got successful. Similarly, the number of packets successfully received from the test node at the receiver were also counted. The experiment was performed for B-MAC payloads of 28, 96 and 116 bytes. The results are shown in Fig. 4. It was observed that B-MAC achieved relatively higher packet success ratio when MAC level acknowledgment was enabled than the case when it was disabled. However, the overall packet success ratio of B-MAC remained less than that of nanoMAC in both of these cases.

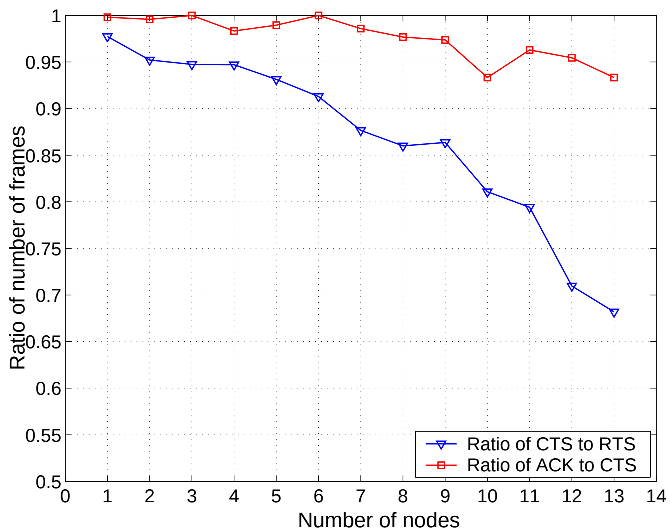

Fig. 3. NanoMAC's control frame ratios at a transmitting node in a highly congested channel.

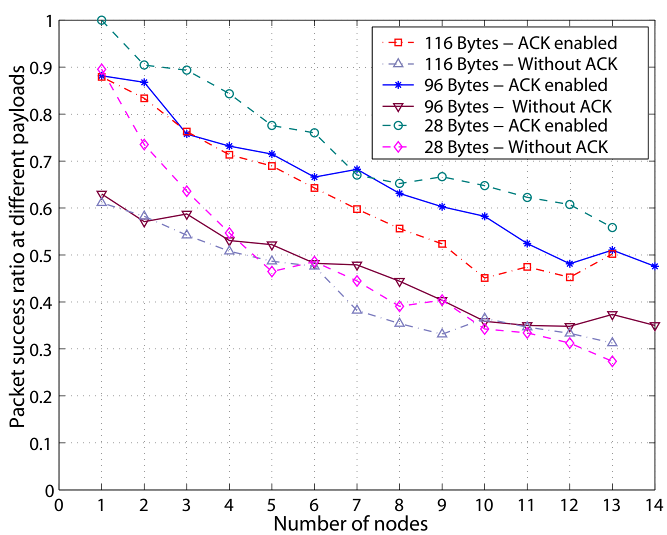

Fig. 4. B-MAC packet success ratio in a highly congested channel.

C. Latency

In most of the sensor network deployment scenarios, multi-hop communication is required. We also conducted nanoMAC latency tests seen at the application level over multihop links. Multiple hops can be realized using a fixed routing scheme even if all the nodes are in the wireless range of each other. An illustration is shown in Fig. 5. Node 1 runs a timer-counter and initiates a unicast transmission to node 2. Node 2 upon receiving the packet, transmits it immediately to node 3. Node 3 transmits the packet to node 4 and so forth. The last node in the chain transmits the packet to the node immediately before it and the same procedure is repeated in exactly the reverse order till the packet reaches back to node 1. Node 1 takes the difference of the timer-counter values as soon as it sends the packet and when it receives the packet back.

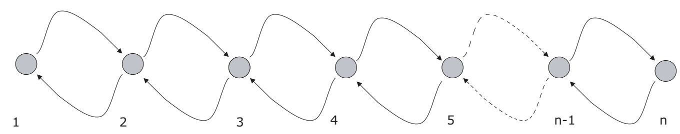

Fig. 5. Multihop latency measurement setup.

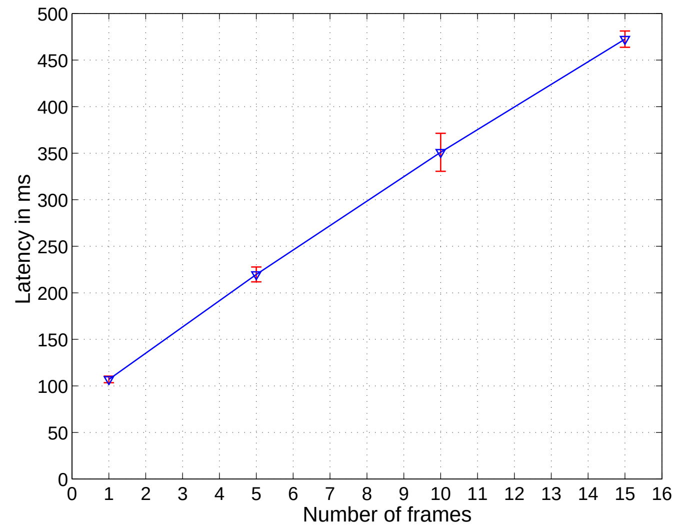

Fig. 6. 2-hops latency of nanoMAC with different number of frames per packet.

It should be noted that this method can only be applied in scenarios where the traffic is extremely low as in the case we are considering here. For higher amounts of traffic, the radio environment would be very different and the results would not be representative. Additionally, we do not attempt to include any kind of route set-up overhead. These issues are highly specific to the routing protocol employed, while our observations are meant to isolate the effects a particular MAC choice has on end-to-end latency.

Fig. 6 shows the 2-hop latency of nanoMAC with different frame sizes. The latency experiences an approximately linear increase with the increase in the number of frames per packet. Fig. 7 shows the multihop latency of nanoMAC with 5 frames per packet which indicates a linear increase in the latency with the number of hops. The latency undergoes a linear decrease per hop, if the initial backoff counter value is set to be zero. Still the latency remains quite high. In comparison, B-MAC offers much lower latency over multihop. The latency of B-MAC was measured over multiple hops using the same experimental set-up for different packet sizes. This is shown in Fig. 8. If the MAC level acknowledgments are disabled, the latency gets down-shifted to approximately 3.5 ms. Generally, MAC protocols using control frames are expected to offer higher latency than the protocols void of any control frames. The reason for the added latency of nanoMAC is also because of the overhead of the control frames (RTS/CTS/ACK). In

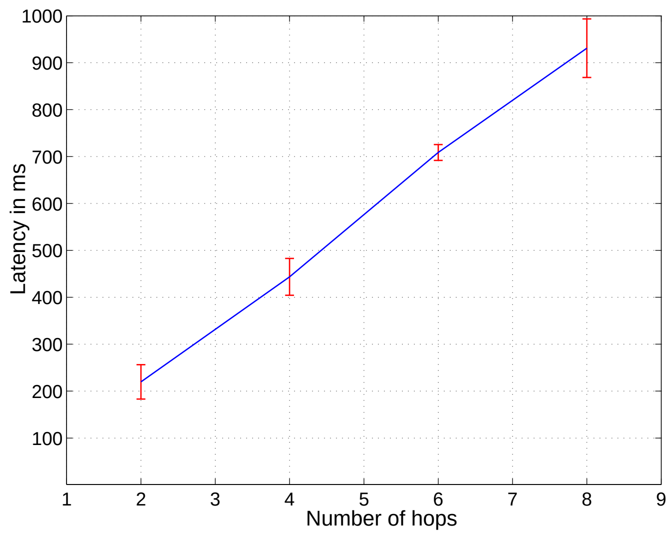

Fig. 7. Multihop latency of nanoMAC with 5-frames/packet.

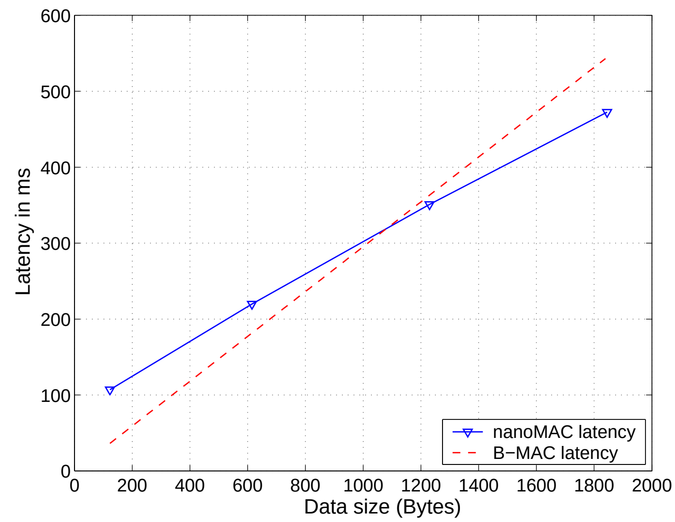

Fig. 9. 2-hop latency comparision of B-MAC and nanoMAC at different data sizes.

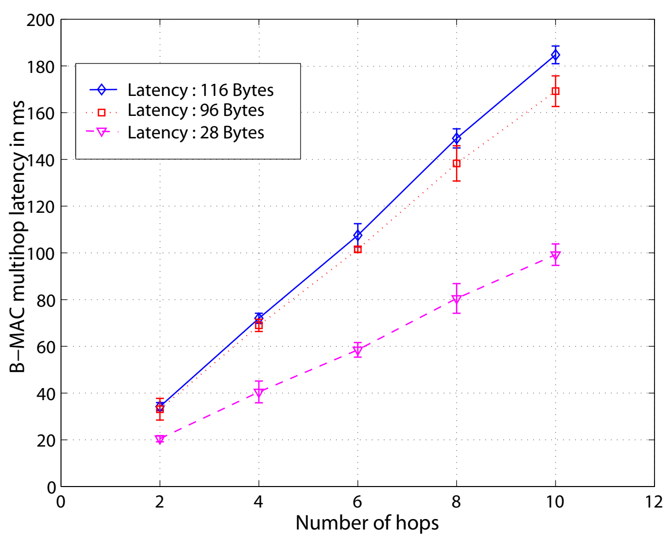

Fig. 8. Multihop latency of B-MAC with acknowledgements enabled.

applications, for instance habitat monitoring [12], where a very little data is to be reported, B-MAC certainly outperforms nanoMAC in terms of latency. In many applications, where the sensor nodes firstly store the measurements locally (e.g. in EEPROM) and later transmit large data on-demand in applications like Wisden [11], nanoMAC has an advantage over B-MAC. A comparison of 2-hop latency versus data size is made in Fig. 9. It can be seen that for smaller data size, B-MAC outperforms nanoMAC while the reverse is true for larger data size.

D. Hidden Terminals

Hidden terminals are present in most of the conceived multihop communication wireless networks. In WSNs, the density of the nodes is generally high and so is that of the hidden terminals. Because of this, nanoMAC's performance was also evaluated in the presence of hidden terminals. The hidden terminals were created by placing two nodes far apart so that they do not hear each other whereas, the receiving node, placed in the centre of these two nodes can hear both.

The success rate of the transmission was measured by counting the number of packets sent at the two transmitting nodes and the number of packets received successfully at the receiving node. The nodes were generating as much traffic as possible. Since nanoMAC is an RTS/CTS based protocol, it performs exceptionally good with very high packet success ratio. Table I shows the packet success ratio of nanoMAC in the presence of the hidden terminals when one frame per packet and ten frames per packet were transmitted.

TABLE I
PACKET SUCCESS RATIO ONE FRAME (TOP) AND TEN FRAMES PER PACKET (BOTTOM).

	RTS	CTS	DATA	ACK	Success Rate
Tx-1	544	424	424	421	99.29%
Tx-2	532	412	409	408	99.75%
Tx-1	106	86	86	84	97.6%
Tx-2	119	91	91	87	95.6%

The larger number of RTS frames transmitted compared to the number of DATA frames, suggests that some of the RTS frames suffer collisions. This is not an alarming problem because RTS frames are very small in size and the induced overhead is not significant.

In contrast to nanoMAC, a protocol without an RTS/CTS handshake undergoes severe degradation in packet success rate due to hidden terminals. With the similar set-up the observations for the 96 bytes B-MAC payload with MAC-level ACK frames enabled and disabled are shown in Table II at the top and bottom, respectively.

TABLE II

PACKET SUCCESS RATIO WITH ACK ENABLED (TOP) AND ACK DISABLED (BOTTOM).

	Packets Sent	Packets Received	Success rate
Tx-1	1652	912	55.2%
Tx-2	1653	961	58.1%
Tx-1	2029	1094	53.9%
Tx-2	2036	969	48.%

V. DISCUSSION

We have conducted experiments to evaluate the performance metrics of the nanoMAC and B-MAC protocol. The experiments were conducted for long enough duration and repeated multiple times to obtain statistically significant results.

Regarding the throughput of nanoMAC in comparison to B-MAC, we have found that nanoMAC gives relatively lower maximum throughput than B-MAC although there is no huge difference. However, B-MAC has a lower packet success ratio in congested channels. In [13] it is shown that the use of RTS/CTS handshake in B-MAC protocol, degrades the throughput to approximately 50% in a highly congested channel. While this result was obtained on MICA2, qualitatively the same results should hold on Telos as well.

Any RTS/CTS based protocol is expected to offer higher latency as compared to protocols void of control frames. In applications like habitat monitoring where very small amount of data is required to be reported, B-MAC certainly outperforms nanoMAC. However in applications where large data measurements are to be reported in bursts, nanoMAC gives an effective lower latency.

NanoMAC works remarkably well when there are hidden terminals in the network. The advantage of RTS/CTS handshake is obvious in the presence of hidden terminals. We argue that the unreliable transmissions that result into a large number of collisions not only wastes the battery power but also add latency. NanoMAC is able to transmit with high reliability even in very congested channels. The RTS/CTS control mechanism not only prevents collisions due to hidden terminals in multihop communication but it also saves idle overhearing thereby conserving energy in two-folds.

VI. CONCLUSIONS

In this paper, we have presented and analysed a CSMA/CA based protocol designed for dense scalable wireless sensor networks and evaluated its performance on Telos motes. NanoMAC is able to perform with high reliability, throughput and efficient channel utilization. The lightweight RTS/CTS handshake not only coordinates the sleeping and synchronization mechanism but also prevents collisions. This saves energy consumption and added latency associated with packet retransmissions. We have also observed that the performance metrics do not suffer an alarming degradation in extreme traffic environments. As a future work, we will experimentally validate nanoMAC's theoretical power consumption analysis in different sleep groups with different amounts of offered traffic as presented in [18]. We are also planning a possible public release of the nanoMAC code for interested parties.

VII. ACKNOWLEDGMENT

We would like to thank DFG (Deutsche Forschungsgemeinschaft), European Union (the RUNES-project) and RWTH Aachen University for the financial support.

REFERENCES

[1] W. Ye, J. Heidemann and D. Estrin, "An Energy Efficient MAC Protocol for Wireless Sensor Networks," *In Proceedings of the 21st International Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2002)*, Jun. 2002.

[2] J. Haapola, "Low-Power Wireless Measurement System for Physics Sensors," *Master's thesis, Department of Physical Sciences, University of Oulu, Oulu, Finland*, May 2002. (unpublished)

[3] J. Haapola, "MAC Energy Performance in Duty Cycle Constrained Sensor Networks," *In Proceedings International Workshop on Wireless Ad- Hoc Networks (IWWAN)*, May-Jun. 2004.

[4] "Telos: IEEE 802.15.4 compliant Wireless Sensor Nodes", *http://www.moteiv.com/products.php*, [Visited Feb.22, 2006]

[5] "TinyOS: Embedded Operating System", *http://webs.cs.berkeley.edu/tos/*, [Visited Feb.22, 2006]

[6] W. R. Heinzelman, A. Chandrakasan and H. Balakrishnan, "Energy-efficient communication protocols for wireless microsensor networks," *In Proceedings of the Hawaii International Conference on Systems Sciences*, Jan. 2000.

[7] L. van Hoesel and P. Havinga, "A lightweight medium access protocol (LMAC) for wireless sensor networks: Reducing preamble transmissions and transceiver state switches," *In Proceedings of 1st International Workshop on Networked Sensing Systems*, 2004.

[8] T. van Dam and K. Langendoen, "An adaptive energy-efficient mac protocol for wireless sensor networks," *In Proceedings of the First ACM Conference on Embedded Networked Sensor Systems*, Nov. 2003.

[9] "Chipcon Corporation. IEEE 802.15.4 compliant low-cost transceiver (CC2420)," *http://www.chipcon.com/files/CC2420_Data_Sheet_1_3.pdf*, [Visited Feb.23, 2006]

[10] "Texas Instrument's ultra low power MSP430-series microcontroller, *http://www.ti.com/msp430*, [Visited Feb. 22, 2006]

[11] N. Xu, S. Rangwala, K. K. Chintalapudi, D. Ganesan, A. Broad,R. Govindan and D. Estrin, "A Wireless Sensor Network For Structural Monitoring," *In Proceedings of the ACM Conference on Embedded Networked Sensor Systems (SenSys)*, Nov. 2004.

[12] A. Cerpa, J. Elson, D. Estrin, L. Girod, M. Hamilton, and J. Zhao, "Habitat monitoring: Application driver for wireless Communications technology,"*In 2001 ACM SIGCOMM Workshop on Data Communications in Latin America and the Caribbean*, Apr. 2001.

[13] J. Polastre, J. Hill, and D. Culler, "Versatile Low Power Media Access for Wireless Sensor Networks," *In Proceedings of the Second ACM Conference on Embedded Networked Sensor Systems (SenSys), Baltimore,MD,USA*, Nov. 2004.

[14] S. Singh and C. Raghavendra, "PAMAS: Power aware multiaccess protocol with signalling for ad hoc networks,"*In Proceedings of the ACM/IEEE Conference on Mobile Computing and Networking*, Oct. 1998.

[15] J. Haapola, Z. Shelby, C. Pomalaza-Ráez and P. Mähönen, "Cross-Layer Energy Analysis of Multi-hop Wireless Sensor Networks,"*In the Proceedings of 2nd European Workshop on Sensor Networks (EWSN) 2005*, Istanbul, Jan-Feb 2005.

[16] J. Polastre, "Sensor Network Media Access Design," *EECS Department, University of California Berkeley, USA*, 2003.

[17] W. Ye, J. Heidemann, and D. Estrin, "Medium access control with coordinated adaptive sleeping for wireless sensor networks," *In IEEE/ACM Transactions on Networking*, vol. 12, June 2004.

[18] J. Haapola, Z. Shelby, C. Pomalaza-Raez, P. Mähönen, "Multi-hop Medium Access Control for WSNs: An Energy Analysis Model", *EURASIP Journal on Wireless Communications and Networking*, vol.4 2005.