01 Jan 2006

Development and Implementation of Optimized Energy-Delay Sub-Network Routing Protocol for Wireless Sensor Networks

Maciej Jan Zawodniok
Missouri University of Science and Technology, mjzx9c@mst.edu

Jagannathan Sarangapani
Missouri University of Science and Technology, sarangap@mst.edu

Steve Eugene Watkins
Missouri University of Science and Technology, watkins@mst.edu

James W. Fonda
Missouri University of Science and Technology, fonda@mst.edu

Recommended Citation

M. J. Zawodniok et al., "Development and Implementation of Optimized Energy-Delay Sub-Network Routing Protocol for Wireless Sensor Networks," *Proceedings of the 2006 IEEE Conference on Computer Aided Control System Design, 2006 IEEE International Conference on Control Applications, 2006 IEEE International Symposium on Intelligent Control*, Institute of Electrical and Electronics Engineers (IEEE), Jan 2006.
The definitive version is available at https://doi.org/10.1109/CACSD-CCA-ISIC.2006.4776634

Development and Implementation of Optimized Energy-Delay Sub-Network Routing Protocol for Wireless Sensor Networks

James W. Fonda*, Maciej Zawodniok, S. Jagannathan, and Steve E. Watkins
Embedded Systems and Networking Laboratory
Department of Electrical and Computer Engineering
University of Missouri-Rolla
Rolla, Missouri 65409
{fonda, mjzx9c, sarangap}@umr.edu, steve.e.watkins@ieee.org

Abstract— The development and implementation of the optimized energy-delay sub-network routing (OEDSR) protocol for wireless sensor networks (WSN) is presented. This on-demand routing protocol minimizes a novel link cost factor which is defined using available energy, end-to-end (E2E) delay and distance from a node to the base station (BS), along with clustering, to effectively route information to the BS. Initially, the nodes are either in idle or sleep mode, but once an event is detected, the nodes near the event become active and start forming sub-networks. Formation of the inactive network into a sub-network saves energy because only a portion of the network is active in response to an event. Subsequently, the sub-networks organize themselves into clusters and elect cluster heads (CHs). The data from the CHs are sent to the BS via relay nodes (RNs) that are located outside the sub-networks in a multi-hop manner. This routing protocol improves the lifetime of the network and the scalability. This routing protocol is implemented over the medium access control (MAC) layer using UMR nodes. Experimental results illustrate that the protocol performs satisfactorily as expected.

Index Terms—Wireless sensor network, self organization, clustering, energy-delay, sub-network.

I. INTRODUCTION

Energy efficient network protocols are an integral part of constructing a practical WSN for deployment [1, 3-17]. Implementation issues that are not always addressed in simulation constrain the type of protocols and hardware that can be deployed. Processing capabilities, on-board battery capacity, and sensor interfacing all become constraints that must be weighed during the design of the hardware components.

Implementations of wireless sensor networks (WSN) protocols are traditionally evaluated through the use of network simulators such as NS2, OPNET, and GloMoSim [1-18]. Simulations allow for establishment of the performance of a particular protocol against others. However, simulations lack the ability to evaluate the protocol against hardware constraints In this work, hardware implementation is shown for the optimal energy delay sub-network routing (OEDSR) protocol [18]. The OEDSR is used in WSN to provide optimal routing calculations in energy and delay dependent environments. The use of hardware developed at the University of Missouri-Rolla (UMR) is shown as a development platform for this implementation.

Available routing protocols for sensor networks are classified as data centric, location-based, QoS aware, and hierarchical protocols. Data centric protocols [17] such as SPIN [4], Directed Diffusion [5], and GRAB [6,7] consolidate redundant data while routing from source to destination. Location-based routing protocols such as GPSR [11], GEAR [12], and TTDD [13] require GPS information to determine an optimal path so that the flooding of routing-related control packets is not necessary. On the other hand, QoS aware protocols such as SPEED [14], address various requirements such as energy efficiency, reliability, and real-time requirements. Finally, the hierarchical protocols such as LEACH [3], TEEN [8], APTEEN [9], and PEGASIS [10] form clusters with cluster heads (CHs) in order to minimize the energy consumption both for processing and transmission of data. These protocols have been evaluated in simulation. However, there are little, or no, experimental results reported how they perform on hardware. In this paper the focus is on hardware implementation of the recently developed OEDSR protocol and its assessment.

This paper will revisit the OEDSR protocol development and its performance evaluation through hardware implementation. In this paper a presentation of an 8-bit 8051 variant microcontroller based implementation platform utilizing 802.15.4 RF communication units is shown. The use of this platform provides high-speed processing, interconnectivity with sensors, and a capable RF communications unit to facilitate a development platform for WSN. The hardware description includes considerations and limitations that the algorithm and hardware incur on one another. A description of the software implementation is next described. First OEDSR is revisited.

II. OPTIMIZED ENERGY-DELAY SUB-NETWORK ROUTING (OEDSR) PROTOCOL

In OEDSR, sub-networks are formed around an event/ fault and nodes wake up in the sub-networks while the nodes elsewhere in the network are in sleep mode. An appropriate percentage of nodes in the sub-network are elected as CHs based on a metric composed of available energy and relative

---

Research supported in part by Dept. of Education GAANN Fellowship, Air Force Research Laboratory Grant (FA8650-04-C-704) through Center for Aerospace Manufacturing Technologies and Intelligent Systems Center.

* Contact author: James W. Fonda can be reached via email: fonda@umr.edu.

0-7803-9798-3/06/$20.00 ©2006 IEEE

location to an event [18] in each sub-network. Once the CHs are identified and the nodes are clustered relative to the distance from the CHs, the routing towards the BS is initiated. First, the CH checks if the BS is within the communication range. In such case, the data is sent directly to the BS. Otherwise, the data from the CHs in the sub-network are sent over a multi-hop route to the BS. The proposed routing algorithm is fully distributed since it requires only local information for constructing routes, and is dynamic in adapting to changes in the network. The BS is assumed to have sufficient power supply, allowing a high power beacon from the BS to be sent such that all the nodes in the network have knowledge of the distance to the BS. It is assumed that all nodes in the network know the distance to the BS at all times. Though the OEDSR protocol borrows the idea of an energy-delay metric from OEDR [1], selection of relay nodes (RN) does not maximize the number of two hop neighbors. Here, the selection of a relay node is set to maximize the link cost factor which includes distance from the BS to the RN.

A. *Optimum Relay-Node-Based Link Cost Factor*

Knowing the distance information at each node will allow the node to calculate the Link Cost Factor (LCF). The link cost factor from a given node to the next hop node 'i' is given by (1) where D_i represent the delay that will be incurred to reach the next hop node in range, x_i is the distance between the next hop node to the BS, and E_i is the energy remaining at the next hop node

$$LCF_i = \frac{E_i}{D_i \cdot x_i} \qquad (1)$$

In equation (1), checking the remaining energy at the next hop node increases network lifetime; the distance to the BS from the next hop node reduces the number of hops and end-to-end delay; and the delay incurred to reach the next hop node minimizes any fading channel problems. When multiple RNs are available for routing of the information, the optimal RN is selected based on the highest LCF. These clearly show that the proposed OEDSR protocol is an on demand routing protocol which consistently outperforms the available routing protocols. For detailed discussion of OEDSR refer to [18]. The route selection is illustrated through the following example.

B. *Routing Algorithm through an Example*

Consider the topology shown in Fig. 1. The link cost factors are taken into consideration to route data to the BS. The following steps are implemented to route data using the OEDSR protocol:

i) Start with an empty relay list for source node n: *Relay(n)={ }*. Here node n_4 and n_7 are CHs.
ii) First, CH n_4 checks with which nodes it is in range with. In this case, CH n_4 is in range with nodes n_1, n_2, n_3, n_5, n_8, n_9, n_{12}, and n_{10}.
iii) The nodes n_1, n_2, and n_3 are eliminated as potential RNs because the distance from them to the BS is greater than the distance from CH n_4 to the BS.
iv) Now, all the nodes that are in range with CH n_4 transmit RESPONSE packets and CH n_4 makes a list of possible RNs, which in this case are n_5, n_8, n_9, n_{12}, and n_{10}.
v) CH n_4 sends this list to CH n_7. CH n_7 checks if it is range with any of the nodes in the list.
vi) Nodes n_9, n_{10}, and n_{12} are the nodes that are in range with both CH n_4 and n_7. They are selected as the potential common RNs.
vii) The link cost factors for n_9, n_{10} and n_{12} are calculated.
viii) The node with the maximum value of LCF is selected as the RN and assigned to *Relay(n)*. In this case, *Relay(n)={n12}*.
ix) Now node n_{12} checks if it is in direct range with the BS, and if it is, then it directly routes the information to the BS.
x) Otherwise, n_{12} is assigned as the RN, and all the nodes that are in range with node n_{12} and whose distance to the BS is less than its distance to the BS are taken into consideration. Therefore, nodes n_{13}, n_{16}, n_{19}, and n_{17} are taken into consideration.
xi) The LCF is calculated for n_{13}, n_{16}, n_{19}, n_{14}, and n_{17}. The node with the maximum LCF is selected as the next RN. In this case *Relay(n) = {n19}*.
xii) Next the RN n_{19} checks if it is in range with the BS. If it is, then it directly routes the information to the BS. In this case, n_{19} is in direct range, so the information is sent to the BS directly.

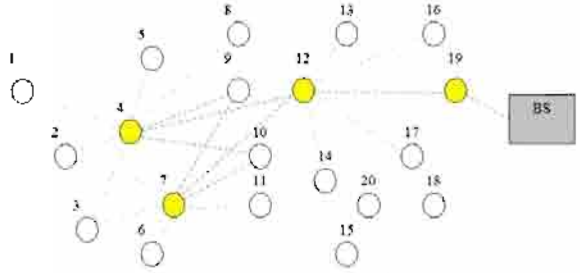

Fig. 1. Relay node selection

III. HARDWARE IMPLEMENTATION DESCRIPTION

In this section an overview of the hardware implementation of the OEDSR protocol is given. Use of customized hardware for development of sensing, processing, and networking will be presented. A description of capabilities, limitations, and support for networking application are given next. Also, in this section an overview of the software architectures are given with respect to the routing protocol and its memory requirements on the hardware.

A. *Hardware Description and Limitations*

Hardware for implementation of the OEDSR was selected to be energy conservative, performance oriented, and of small form-factor. Use of Silicon Laboratories® 8051 variant hardware was selected for its ability to provide fast 8-bit processing, low-power consumption, and ease of interfacing to peripheral hardware components. Next, a treatment of the hardware capabilities and limitations will be given.

Hardware implementation of any algorithm is constrained by the limitations of the hardware. Use of specific hardware must be weighed against the precision, speed, and criticality of an algorithm's implementation. Constraints addressed for the implementation of the OEDSR were use of low-power, small form-factor, and fast processing hardware. For this protocol, low-power consumption was given the highest priority. In turn, the demand for low power limits the types of processor architectures that can be deployed. The selection of the Silicon Laboratories® 8051 variants was based on these criteria. Limitations for the implementation that are incurred through the use of the 8051 variant family are a small memory space and a maximum processing speed. In the next section, a description of the specifications for the hardware implemented nodes will be given.

B. Architecture of the Hardware System and Software

Now, a discussion of hardware and software resources employed for implementation of OEDSR is given. A hardware performance comparison of sensor node platforms used at UMR is shown. Software implementation in terms of architecture, control-flow, and hardware limitations are shown.

1) Sensor Node – Instrumentation Sensor Node

The UMR Instrumentation Sensor Node (ISN), as seen in Fig. 2(a), is used for interfacing sensors to CHs. The ISN allows a sensor to be monitored by a small and low-power device that can be controlled by CHs which is also another node. In this application, the ISN is used as the source of sensor traffic. The ISN is capable of being interfaced with several sensor types and can be instructed by control packets to transmit data in raw or pre-processed form.

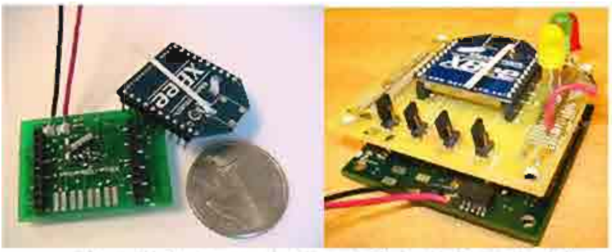

Fig. 2. (a) Instrumentation Sensor Node (ISN), (b) G4-SSN

2) Cluster Head and Relay Nodes

The Generation-4 Smart Sensor Node (G4-SSN), seen in Fig. 2(b), originally developed at UMR and subsequently updated at St. Louis University was chosen as the CH. The G4-SSN has various abilities for sensing and processing. The former include strain gauges, accelerometers, thermocouples, and general A/D sensing. The later include analog filtering, CF memory interfacing, and 8-bit data processing at a maximum of 100 MIPS. The G4-SSN provides memory and speed advantages over the ISN that make it a suitable choice for implementation as a CH or a RN. Future work is being undertaken to develop a better CH that is more powerful than an ISN and smaller in size than a G4-SSN.

3) Comparison of ISN and G4-SSN Capabilities

The abilities of the G4-SSN and the ISN sensor nodes are compared in this section. The ISN was designed to be a simplified sensor node with the abilities to sample, process and transmit data. The ISN has a limited ability to process data relative to the G4-SSN. The abilities of the two nodes are shown in Table I with a comparison to other commercially available harware. As seen in the table the G4-SSN has approximately 4-times the processing speed available relative to the ISN. Memory constraints are also shown between the two sensor nodes, with the G4-SSN having more available code space and RAM. This translates to the design criteria for the ISN to be a 'simple sample and send sensor node'. In comparison, the G4-SSN is used for networking functionality and other tasks that require more memory and processing ability. In the next section, an overview of the software architecture is given for the OEDSR implementation.

TABLE I
COMPARISON OF G4-SSN AND ISN CAPABILITIES

	Ic @ 3.3V [mA]	Flash Memory [bytes]	RAM [bytes]	ADC Sampling Rate [kHz]	Form-Factor	MIPS
G4	35	128k	8448	100 @ 10/12-bit	100-pin LQFP	100
ISN	7	16k	1280	200 @ 10-bit	32-pin LQFP	25
X-Bow	8	128k	4096	15 @ 10-bit	64-pin TQFP	8

4) Software Architecture

The software architecture for 8051 platform is presented in this section. The network stack is presented and the layers are discussed in detail. The software architecture utilized to implement OEDSR protocol on 8051 platform is presented in Fig. 3.

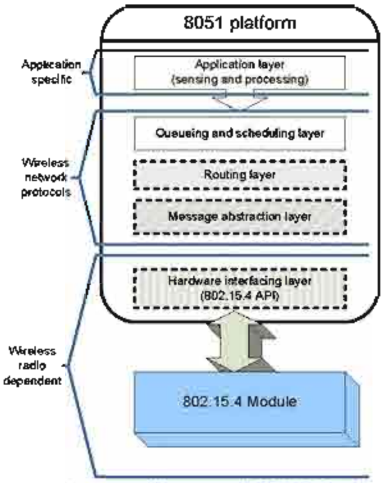

Fig. 3. Software architecture of OEDSR implementation.

The three- tier structure is used to provide flexibility to the radio and application design. The wireless radio dependent components are interfaced with networking layers through the message abstraction layer. This layer provides generic access to the physical and link level parameters and information, for example transmission power level and RSSI indicator. Consequently, cross-layer protocols such as OEDSR can be easily implemented.

The main components of the software architecture consists of

- Physical interface between 8051 and 802.15.4 module – in the used setup a standard serial interface connects processor with radio module
- Abstraction layer – provides generic access to the physical and link layers
- Routing layer – contains OEDSR implementation
- Queuing – a simple drop-tail queuing policy is employed, and
- Sensing application – application dependant measurement and processing of sensor data.

C. *Routing Implementation*

In this section the implementation of the routing protocol is described. Including, packets used by the routing protocol, handling of traffic cases by a node, and memory handling are presented.

1) Routing Packets

The routing aspects of the OEDSR protocol have been implemented on the 8051 platform with an 802.15.4 radio module. Five types of messages have been considered:

- BEAM packet

  The BS broadcasts BEAM packets to whole network to wake-up nodes and initiate data transmission. Radio Signal Strength Indicator (RSSI) is retrieved by the receiving nodes and used to estimate the distance to the base station.
- HELLO packet

  The node while searching for a route to the BS broadcasts HELLO packets to neighbors periodically until ACK is received, or until timeout. The distance to BS is included so that the receiving node can determine the closest node to the BS.
- Acknowledgement (ACK) packet

  ACK is sent as a response to the HELLO packet when the node's distance to BS is smaller than the requesting node's distance. Also, ACK contains node's remaining energy and distance to the BS. The HELLO source node receives ACK packet and calculates a transmission delay. The link cost is calculated and temporarily stored to compare it with later responses.
- SELECT packet

  When HELLO/ACK timeout elapses, the node selects the route based on the link costs from the stored ACK information. Subsequently, the SELECT packet is sent to the selected node to indicate route selection. The receiving node starts route discovery toward BS by sending a HELLO packet.
- DATA packet

  The DATA packet conveys application specific data to the BS.

2) Traffic Cases

Fig. 4 presents a block diagram of the routing implementation. The handling of the received message starts at the RX block, where the type of the packet is determined. Next, the processing proceeds depending on the packet type.

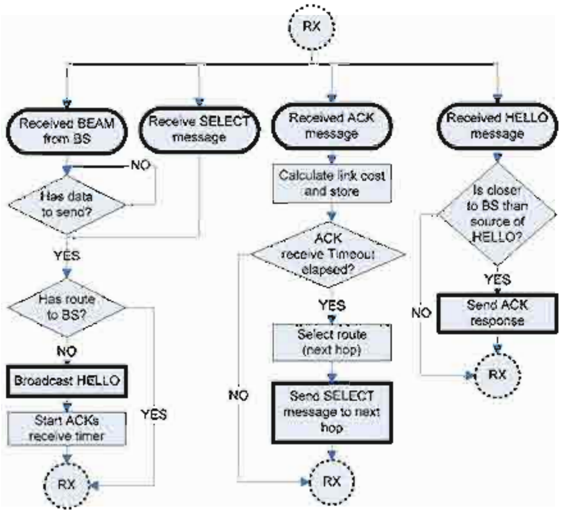

Fig. 4. Control flow scheme for OEDSR routing implementation

3) Memory Limitations

Memory limitations are incurred by the hardware. The routing protocol requires a particular amount of memory to store the routing table and temporary information from ACK. The number of routing table entries depends on expected number of active CHs. Moreover, the routing tables store only a link cost value calculated from HELLO-ACK exchange. Furthermore, in order to reduce memory requirements, periodically inactive sources are purged from the routing table.

IV. PERFORMANCE EVALUATION

Experiments for OEDSR where performed using a network of UMR ISN's and the G4-SSN's. Experimental results are compared to static routing to demonstrate the dynamic routing of the OEDSR. Use of static routing provides an initial assessment, while future work will provide comparison to existing protocols.

The nodes use 802.15.4 modules transmitting at a 250 kbps RF data rate. The ISN is used to generate CBR traffic and provide data source functionality. CH's and RN's are implemented using the G4-SSN. The CH provides the OEDSR routing capabilities by choosing the RN for routing of traffic toward the BS. The node's processor interfaces to the 802.15.4 module at 38.4 kbps, the maximum supported data rate. The ISN, CH, and RN's are equipped with low-power 1-mW 802.15.4 modules; while the BS is equipped with a high transmission power, 100-mW, 802.15.4 module to increase the BS range for beam signals.

A. *Description of the Experimental Scenario*

Experimental scenarios where performed with 12 nodes placed in the topology illustrated in Fig. 5. The topology was then modified by the amount of energy available in each node to perform testing of the protocol's ability to provide dynamic optimal routing based on energy, delay, and distance. Testing demonstrates the ability of the OEDSR protocol to evenly balance the energy consumed in the entire network while providing suitable delay in the transmission of packets.

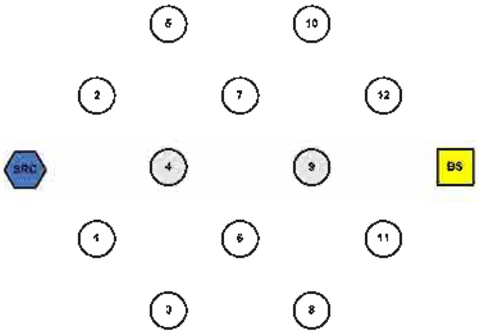

Fig. 5. Network schematic

B. *Experiment Results*

The network performance is measured in terms of throughout, end-to-end (E2E) delay, drop rate, and number of total dropped packets. Experiments were repeated for varying energy levels at each node, thus enforcing route changes. In Table II the performance measurements are shown for the 6 experimental cases. Each test was ran for 3 minutes and an average results is shown. The experimental scenarios were prepared to generate four-hop routes thus providing comparable data sets. Throughput and E2E delay are consistent across all six cases since the routing algorithm selects an optimal route regardless of energy distribution in the network. Variance in the number of dropped packets and in the drop rate is attributed to the distribution of packet collisions. In Table III a comparison of OEDSR network performance with varied packet size is shown. The network performance degrades as the packet size reduces and the number of generated packets increases. Since the amount of bandwidth used to transmit overhead bits increase at the expense of user data throughput, decreasing packet size increases overhead.

Fig. 6 illustrates throughput when an active RN is removed from the network and OEDSR reestablishes communication.

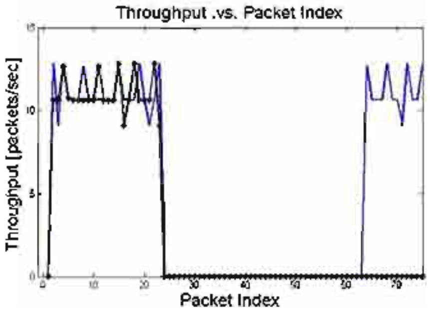

Fig. 6. Throughput for data rate of 1kB/s and 90 bytes per data packet

At packet index 25 there is a drop in throughput when the RN is removed. Subsequent reestablishment of an alternate route by OEDSR is demonstrated since the throughput is restored. In comparison, static routing is not able to recover and would require manual intervention causing continued network downtime.

TABLE II
OEDSR PERFORMANCE FOR DIFFERING TOPOLOGIES

Test	Throughput [bps]	E2E Delay [sec]	Dropped [packets]	Drop Rate [packets/sec]
T1	1152.0	0.7030	181	1.9472
T2	970.6	0.7030	3	0.0321
T3	972.0	0.7030	6	0.0643
T4	1035.5	0.7020	73	0.7811
T5	1048.0	0.7020	83	0.8862
T6	1047.3	0.7030	84	0.8968
AVG	1037.6	0.7027	72	0.7680

TABLE III
OEDSR PERFORMANCE FOR DIFFERING PACKET SIZE

Packet Size [bytes]	Throughput [bps]	E2E Delay [sec]	Dropped [packets]	Drop Rate [packets/sec]
30	1075.7	0.2500	188	2.0181
50	1197.0	0.3440	167	2.7991
70	1096.1	0.5620	156	1.6715
90	1047.3	0.7030	84	0.8968

Static routing was compared to the OEDSR protocol. The route was manually configured to mimic a desired route. Experimental results show a similar throughput, E2E delay, and drop rate for the static routing and OEDSR. However, a lack of dynamic network discovery period is observed during network initialization with static routing. In the case of OEDSR, the setup time is dependent on the number of hops and the query time for each hop. In contrast, static routing requires manual setup for each topology change which can take long periods of time. It is important to note that static routing is not normally preferred due to node mobility and channel fading.

1) Future Work

Future work will involve evaluating the proposed protocol in the presence of node mobility, and channel fading. Additionally, a performance comparison with other protocols such as LEACH is planned. Preliminary results of the OEDSR hardware implementation as compared to a static routing show promise. Future work will include implementation of protocols such as AODV and DSR on UMR hardware. Comparisons of OEDSR to other standard protocols Can be shown. Other considerations include larger topologies, differing traffic loads and patterns, and vehicular mobile nodes.

V. CONCLUSION

In this paper a hardware implementation is shown for the OEDSR WSN protocol. The objective was to develop a fully distributed routing protocol that provides optimal routing. The route selection is based on a metric given by the ratio of energy available and delay multiplied with distance which is used as the link cost factor.

The proposed OEDSR protocol computes the energy available and average E2E delay values of the links and this information along with the distance from the base station determines the best RN. While ensuring that the path from the CH to the BS is free from loops, it also ensures that the selected route is both energy efficient, and has the least E2E delay. Additionally, the lifetime of the network is maximized since the energy is taken into account while selecting nodes from a route. Due to the energy level being considered in the routing protocol, there is also a balancing of energy consumption across the network.

Implementation of the OEDSR protocol was shown using the G4-SSN and ISN hardware at UMR. The protocol was shown to provide suitable traffic rates and short E2E delays. Drop rate and E2E delay are dependent on the packet size that is being transmitted. Drop rate increases and E2E delay decrease as the packet size decreases. A decrease in E2E delay is expected due to the larger number of packets required to send the same information, however higher traffic volume also increases the probability of packet collisions on the channel and increases overhead.

A series of tests taking a nominal of 4 hops was performed to show the capabilities of the OEDSR routing protocol to provide needed throughput on the network with dynamic routing capabilities. An average throughput of approximately 1 kbps and an E2E delay of 0.7 seconds are observed for a nominal route.

In reference to implementation, several issues where confronted. First, the issue of hardware capabilities is of concern. Selection of hardware must consider the complexity and memory footprint of an algorithm. The constraints of the 8-bit hardware become known during implementation of the OEDSR protocol. For example, the ISN nodes were designed to minimize physical size of the node and reduce energy consumption. However, the selected processor does not have enough RAM to support the OEDSR routing. Therefore, minimum hardware requirements in terms of memory size, processing power, energy consumption, physical size, and the corresponding tradeoffs have to be explored before the particular protocol is targeted and implemented. Additionally, the limitations of the off-the-shelf radio modules are limiting current capabilities of the proposed solution. In particular, the 38.4 kbps limit on the interface to the 802.15.4 module reduces the overall throughput and increases delay at each hop, when compared to a theoretical 802.15.4 capabilities.

ACKNOWLEDGMENT

Thanks go to Dr. Kyle Mitchell of St. Louis University, St. Louis, MO. His continued support and advice on the G4-SSN system and hardware implementation issues is much appreciated.

REFERENCES

[1] N. Regatte and S. Jagannathan "Optimized Energy-Delay Routing in Ad Hoc Wireless Networks," *Proc. of the World Wireless Congress*, May 2005.

[2] Global Mobile Information Systems Simulation Library, http://pcl.cs.ucla.edu/projects/glomosim/

[3] W. R. Heinzelman, A. Chandrakasan and H. Balakrishnan, "Energy-Efficient Communication Protocol for Wireless Microsensor Networks," *IEEE Trans. Wireless Commun.*, pp. 660-70, Oct. 2002.

[4] W. R. Heinzelman, J. Kulik and H. Balakrishnan, "Adaptive Protocol for Information Dissemination in Wireless Sensor Networks," *Proc. Fifth Annual ACM/IEEE Internation Conference on Mobile Computing and Networking (MobiCom)*, Aug. 1999.

[5] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann and F. Silva, "Directed Diffusion for Wireless Sensor Networking," *Networking, IEEE/ACM Transactions*, pp. 2-16, Feb. 2003.

[6] F. Ye, G. Zhong, S. Lu and L. Zhang, "GRAdient Broadcast: A Robust, Long-Lived Large Sensor Network," *ACM Wireless Networks*, March 2005.

[7] F. Ye, G. Zhong, S. Lu and L. Zhang, "PEAS: A Robust Energy Conserving Protocol for Long-Lived Sensor Networks," *Proc. 23rd International Conference on Distributed Computing Systems (ICDCS)*, pp. 28-37, May 2003.

[8] A. Manjeshwar and D. P. Agrawal. "TEEN: A Routing Protocol for Enhanced Efficiency in Wireless Sensor Networks," *Proc. 15th Parallel Distributed Processing Symposium*, pp. 2009-2015, 2001.

[9] A.Manjeshwar, and D.P. Agrawal, "APTEEN: A Hybrid Protocol for Efficient Routing and Comprehensive Information Retrieval in Wireless Sensor Networks," *Proc. Int'l Parallel and Distributed Processing Symp. (IPDPS 2002)*, April 2002.

[10] S. Lindsey and C. Raghavendra, "PEGASIS: Power-Efficient Gathering in Sensor Information Systems", *IEEE Aerospace Conference Proceedings*, pp. 1125-1130, 2002.

[11] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler and K. Pister, "System Architecture Directions for Networked Sensors," *Proc. ASPLOS*, pp. 93-104, 2000.

[12] J. Zhang, Z. Yang, B. H. Cheng and P. K. McKinley, "Adding Safeness to Dynamic Adaptive Techniques," *Proceedings of ICSE 2004 Workshop on Architecting Dependable Systems*, May 2004.

[13] H. Luo, F. Ye, J. Cheng, S. Lu and L. Zhang, "TTDD: A Two-Tier Data Dissemination Model for Large-Scale Wireless Sensor Networks," *Proc. of International Conference on Mobile Computing and Networking (MobiCom)*, pp. 148-159, Sept. 2002.

[14] T. He, J. A. Stankovic, C. Lu and T. Abdulzaher, "SPEED: A Stateless Protocol for Real-Time Communication in Sensor Networks," *Proc. of International Conference on Distributed Computing Systems*, pp.46-55, May 2003.

[15] C. E. Perkins, E. Belding and S. Das, "Ad hoc On-Demand Distance Vector (AODV) Routing," *IETF MANET Working Group*, Internet Draft, www.ietf.org/internet-drafts/draft-ietf-manet-aodv-14.txt, Feb. 2003.

[16] D. Johnson, D. Maltz and Y. Hu, "The Dynamic Source Routing Protocol for Mobile Ad Hoc Network (DSR)," *IETF MANET Working Group*, Internet Draft, draft-ietf-manet-dsr-09.txt, April 2003.

[17] M. Esler, J. Hightower, T. Anderson, and G. Borriello "Next Century Challenges: Data-Centric Networking for Invisible Computing: The Portolano Project at the University of Washington," *ACM/IEEE International Conference on Mobile Computing and Networks (MobiCom)*, Aug. 1999.

[18] S. Ratnaraj, S. Jagannathan, and V. Rao, "Optimal Energy-Delay Subnetwork Routing Protocol for Wireless Sensor Networks", Proc. of IEEE Conference on Networking, Sensing and Control, April 2006.