A Data Delivery Protocol for Extremely Resource Constrained Wireless Sensors

Rumesh Hapuarachchi[1], Asanka Sayakkara[2], Chamath Keppitiyagama[3]

University of Colombo School of Computing
Colombo, Sri Lanka
[1]rehrumesh@hotmail.com
[2]asa@ucsc.cmb.ac.lk
[3]chamath@ucsc.lk

***Abstract*— Applications based on Wireless Sensor Networks (WSN) have a huge potential to enhance human life. A typical WSN application uses a large number of sensor nodes and they will cover a large geographical area. Such sensor nodes cost a lot and implements entire IPv6 stack for its communication. Since most of the current applications are focused on smaller environments, implementing entire IPv6 stack is a resource wastage. With the current development in the industry it is possible to develop extremely resource constrained sensor nodes which cannot run IPv6 stacks. In this study we present a data communication protocol which is capable of working on top of such extremely resource constrained devices. This protocol will use a cross layer approach where it will only use the MAC layer and Application layer instead of the full IPv6 networking stack.**

***Keywords*— Wireless Sensor Networks, Data communication, Resource Constrained Sensors, MAC protocols**

I. Introduction

A Wireless Sensor Network (WSN) can be described as a self-configuring network of small sensor nodes which are communicating among themselves using radio signals to sense the physical world. There are many types of sensor nodes have been used in WSN. These nodes are designed in a way that they are easy to use and to handle node failures [1-2]. Most of the sensor nodes are equipped with powerful micro controller, extended battery life and much powerful sensors or sensor interfaces. According to the Table 1 it is clear that MCUs are becoming available. With the performance improvement of the Micro Controller Unit (MCU), cost of those devices will increase. Also the power consumption of the overall device will be increased. Even though performance of MCU's have been increased greatly, it is notable that the battery life has not increased a lot. These sensor nodes (motes) are designed to work with special Operating Systems such as Contiki [18] and TinyOS [23].

Primary objective of these sensor nodes are to sense the environment and transmit the results to a different node so the result can be processed. For a given application, large number of nodes have to be used because of this nature. Communication over multi-hop routing protocols are inherent given the above nature. Current nodes make use of either IPv6 or 6LoWPAN as the communication protocol stack. However, the operational lifetime of a wireless sensor mote drastically decreases with the rate of packer transmission and receptions performed. Figure 1 shows IEEE 802.15.4 frame. According to the figure it is clear that around 73 bytes of different headers are inside the frame which is not useful data in application context. Under such circumstances, having a complete networking stack on an extremely resource constrained wireless sensor is questionable. As a solution Figure 2 describes a sample frame which contains only meaningful data.

TABLE I
COMPARISON OF MICRO CONTROLLER UNITS

Module	ROM	RAM	Inst	Current(uA)
Attiny85	8kB	512B	8bit	300@1MHz
ATmega128L	128kB	4kB	8bit	5000@4MHz
MSP430G2553	16kB	512B	16bit	330@1MHz
MSP430F1611	48kB	10kB	16bit	330@1MHz
LPC1100L	16kB	4kB	32bit	840@1MHz
Cortex-M4	64kB	64kB	32bit	47810@80MHz

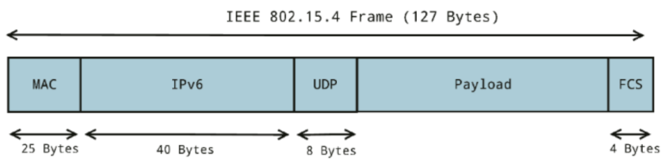

Fig 1. IEEE 802.15.4 Frame

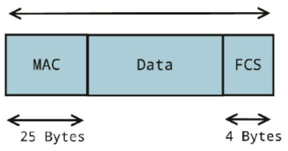

Fig 1 Simple data packet

The paper is organized as follows. Chapter Background describes related work around the domain and what approaches have been used in the Wireless Sensor Network data communication. In Chapter Design, we have described proposed protocol design we used in this research. Chapter Implementation defines the implementation of the protocol design. In Chapter Evaluation we present our evaluation setup and experiment results. Chapter Conclusion used to define future work for our research and to conclude the research.

II. Background

A. *Data communication in Wireless Sensor Networks*

Wireless Sensor Networks consist with large number of application specific sensor motes spread over a large area. These nodes need to communicate among other nodes in order to make use of the data recorded by the sensor mote. Following are the characteristics of data communication in

WSN context discussed in the research done by Purushotham BV, Prakasha S, and K Ganesan [5]:

- Reliability
- Scalability

Reliability of the data communication depends on the application requirements. Studies conducted by Purushotham BV et al. [5], Park et al. [6] and Wan et al. [7] emphasis that the occasional loss of data readings from an application which focuses on monitoring temperature etc. will not have a significant effect on the entire application. But in applications like target tracking, field survey & intrusion detection systems must be highly reliable. Since motes are resource constraint devices number of researches are focused on achieving high reliability and efficiency. When designing a data communication mechanism these factors need to be considered.

B. MAC Protocols

Channel accessing in a network is controlled by the MAC protocol. When considering wireless sensor networks, lack of resources is inevitable. This cause challenges when designing MAC protocols to WSN. Challenging application environment and ad-hoc nature of the network are some of the factors for the difficulty of designing MAC protocols [8]. Since there are number of considerations, it is difficult to design a universal MAC protocol which addresses all the complications.

In the literature, MAC protocols are classified in different ways [8-11]. Rahul et al. categorize MAC protocols as Synchronous and Asynchronous [11]. In the study conducted by Kumar et al. MAC protocols are classified as Contention-based and Contention-free [12]. Thirty-four MAC protocols which are designed around ad-hoc mobile networks have been analyzed by the study done by Judak et al. [13]. Their classification can be generalized in to six main features which includes channel separation, topology, power, transmission initiation, traffic load and range.

C. Analysis of existing MAC Protocols

Different MAC protocols try to address different factors. In this section we have compared some of the widely used MAC protocols in WSN.

S-MAC [4], is one of the classical synchronous MAC protocol. Main focus of this protocol to address the energy wasting factors mentioned in later section. In this protocol nodes have to adopt schedules at the beginning of SYNC period. After the SYNC period nodes can communicate when they are in DATA period. Nodes will be in a sleep state when SLEEP period occurs. In the DATA period prior to the data communication nodes will exchange RTS (Request to Send) and CTS (Clear to Send) packets. Upon receiving the CTS nodes can communicate the data while other nodes will go in SLEEP period.

B-MAC [14] employs an adaptive preamble to reduce idle listening. When a node has a packet to send, it waits during a back off time before checking the channel. If the channel is clear, the node transmits; otherwise it begins a second (congestion) back off. Each node must check the channel periodically using LPL (low-power listening); if the channel is idle and the node has no data to transmit, the node returns to sleep [14].

Asynchronous duty cycling protocols are widely used in WSN and one such protocol is RI-MAC [16]. In this protocol a node will have its own schedule. PW-MAC [17] uses improvements of S-MAC and B-MAC. Since it use pseudo random schedules all the nodes will not wake up at same time to communicate. It will avoid collisions. A node will send a beacon to notify other nodes that it is awake. PW-MAC uses a seed to get the information of other nodes. According to the study conducted by Kabara et al. disadvantage of PW-MAC is the overhead generated by the beacon packets [15].

ContikiMAC [18] uses improved periodic wake ups inspired by B-MAC and BoX MAC. It also uses phase lock optimization mechanism which is suggested in Wise MAC protocol. Rather than sending a different beacon, ContikiMAC keep sending multiple copies of the data packet as a wake up strobe.

Cross layer approach for data communication in Wireless Sensor Networks has been used in many protocols. P-MAC is one such MAC protocol. Even though other MAC protocols are designed in efficient ways, they are designed independently. Hence routing protocol is a must for such networks. Authors have proposed a cross layer design for the protocol so it will handle both Routing and MAC. Another research [19] shows how cross layer design can be leverage to address data communication in high data rate wireless sensor networks.

D. Energy saving mechanisms used in different MAC protocols

According to researchers there are number of mechanisms that can be used to preserve energy. Those mechanisms are listed as follows:

- Duty Cycling

  Applying suitable sleep & wake mechanism to conserve energy known as duty cycling. It has been used in most of the MAC protocols. Many researchers have proposed different duty cycling mechanisms [4] [14] [18] [20-21]. Since idle listening cost as much as energy used to transmit data, duty cycling is an effective mechanism to save energy as it will wake the transceiver only when it is necessary to transmit/receive data.
- Energy-efficient scheduling

  Energy-efficient scheduling is efficient scheduling that can adapt to situation demand can reduce the energy consumption at all levels of the network.
- Scheduled rendezvous

  In this mechanism neighbors of a node will have a prescheduled rendezvous time to wake up and communicate. All the nodes will sleep until the next rendezvous time. This will guarantee that whenever a node wakes up, all the neighbors are also awake at the moment. This mechanism has been used for environmental monitoring [22].
- On-demand wake-up scheme

  In this mechanism multiple radios are being used. One radio which consume less energy will broadcast a wake-up tone to neighbors. In this wake-up tone there will be no information encoded.

III. Design

According to the literature it is clear that MAC protocols are used to access the medium while communicating the data. Some of these MAC protocols are application specific while some of the protocols focus on reducing the energy wastage. According to the literature these protocols are used in applications where sensors spread in wide area. Therefore, these applications will also use routing protocols in order to successfully transfer data to a base station. However, if we consider a small environment like a house, with the power of the current radio transceivers, sensor can communicate with each other directly without using a routing protocol. This have not been addressed in the literature. DispSense [24] is an architecture which can be used in domestic environments. Our goal in this research is to develop a communication protocol for extremely resource constraint devices which are similar to the devices used in DispSense architecture.

Traditional WSN consists with mote devices which implement full IPv6 stack on them. It is justifiable to conclude that the packet complication of IPv6 will introduce additional overhead to the WSN. For simple applications which are used in a small environment, this additional overhead will affect significantly. Therefore, in DispSense architecture we have completely removed the traditional networking stack from sensor nodes and will implement a simple networking stack where a cross layer MAC protocol will enable the communication of nodes.

A. *DispSense Architecture*

DispSense architecture consist with a resourceful Mother mote and resource constraint sensor devices. To preserve the energy, these sensing devices will only focus on sensing and transmitting raw data to the Mother mote. Mother mote is responsible for the processing of data [24]. According to the architecture sensor nodes will only communicate with the mother mote. Therefore, it is unnecessary to use any routing protocol to communication. Simple MAC protocol would be sufficient as sensor node will be communicate with the mother mote via only a single hop. In practical applications there can be multiple Mother motes deployed. Since these mother motes are assumed to be resourceful devices those can be inter connected with complete network stack.

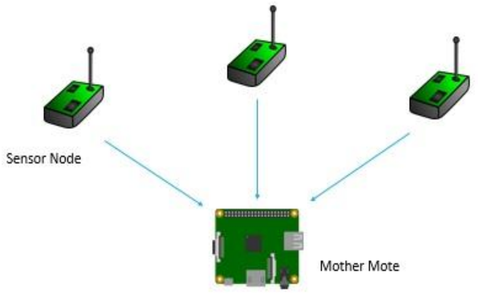

Fig 3 DispSense Architecture

Accoridng to the architecture (Figure 3) data processing will happen only in the Mother Mote. Therefore, in the sensor node there will not be any complicated prcessing. Thuus it eliminates the necessity of having a powerful micro-controller.

B. *Overview of the communication protocol*

DispSence is targeted mostly on sensing applications. Therefore we have designed its sensor nodes targeting more towards the sensing. This eliminates the data processing overhead from the sensor. So the primary task of these sensor nodes would be to use the sensing interface to gather the reading of the sensor and transmit it to the mother mote. One of the key feature of sensing application is over a short period of time there will be significantly higher number of sensor readings. When the number of readings are higher, loss of small number of readings will not affect aggregated results in a significant level. Therefore, we have decided to use a transmit only protocol for the sensor as loss of few data points will not affect the final answer significantly. This allow us to remove few control packets which have been used in other protocols like ACK packet for each data frame.

However these nodes need not to transmit data all the time so these nodes could use a simple duty cycling mechanism to avoid radio module being turned on the whole time. Sensor nodes can have their own schedules or they can communicate with the Mother mote to derive a synchronized schedule. This can be an application specific process. To obtain information from the mother mote a mechanism like beacon could be used. These beacon will initially give scheduling instructions to the sensor node. After receiving the beacon, sensor node can sense and transmit data independently. While communicating, data collisions might occur. To decrease the collision ratio, it is possible to use a CCA (Clear Channel Assessment) or a simple Collision Avoidance Mechanism which is similar to the mechanism used in NFC.

Literature shows that for scheduling mechanism one of the disadvantage would be the clock drifting with other nodes. To fix this clock drifting problem on the mother mote, we can use a phase locking mechanism. However, this will introduce additional overhead to the protocol.

In this protocol we assume that each mother mote is aware of the sensor nodes which are associate with it. Nodes which are joining the network later will send a request to join the network once the Mother mote start accepting the new node requests. Because of the above assumption, self-configuring nature of the protocol will be removed from the protocol. Initially a human interaction would be needed to configure the sensor nodes and the mother mote. Protocol works as follows.

Mother mote maintains a list of nodes which are associating with it. There are few constant values related to the protocol. These constants values can be pre-configured. However, changing these values during the execution time is not allowed.

- Ttotal: Total cycle time
- Tdata: time interval for sensor motes to transmit data.
- Tnewnode: time interval for new nodes to join the network.

Tdata has been divided into slots. Each slot is known as a window. In each window, a sensor node is allowed to transmit data once. This transmission time slot is called as Tframe. Figure 4 shows the protocol in depth.

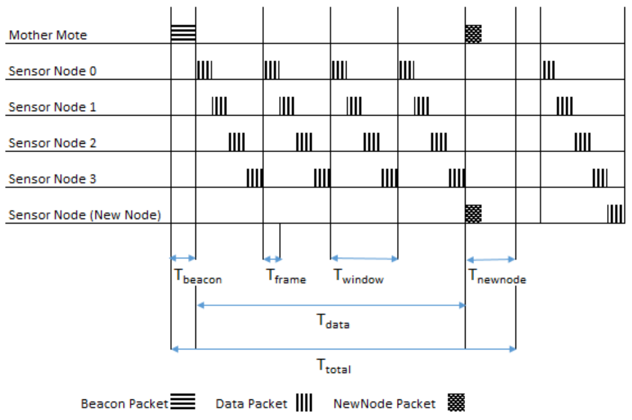

Fig 4 Protocol Design

Initially mother mote broadcast a beacon packet to indicate that it accepts packets. This beacon contains its MothermoteID, number of sensor nodes and the Twindow value. Sensor nodes will receive this packet and nodes which are initialized will start to transmit data according to the following pattern.

In a window, node is allowed to transmit data only once. Therefore it will use a sequence number to determine which frame belongs to which node. For the simplicity we are assuming that the SensorNodeId is the sequence number. Node will sleep till its sequence occurs. Then it will use that frame to transmit data to the sensor node. Once the data is transmitted, sensor node will turn off the radio and sleep till its next sequence in the next window. This will iterate till the Tdata period expires. After Tdata time all sensor nodes will stop transmitting data and let new nodes to communicate with the Mother mote.

Mother mote accepts new node requests till Tnewnode time expires. If a request comes to the mother mote it will update its receivers list and from the next beacon onwards, it will transmit the new node size.

$$Ttotal = Tbeacon+Tdata+Tnewnode$$

$$Tframe = \frac{Twindow}{Nnodes}$$

IV. Design

For the implementation there are two methodologies available.

- Implementation in a simulator. Ex: NS2
- Implementation on top of real hardware.

In this study we are focused on small networks and according to the literature such networks contains only small number of nodes in a network. Therefore, we decided to implement the protocol on top of the hardware rather than implementing on top of the simulator. Implementing this protocol in a simulator and evaluating it for dense networks can be considered as a future work.

We have used Arduino kit to program the Mother mote and the Sensor Node. Arduino is capable of using USBasp to program ATtiny85 MCU. Therefore, we can use the same programming approach for both Sensor Node and Mother Mote.

Figure 5 shows how the protocol is being implemented. MAC layer and the Application layer has been divided in to two layers for the ease of understanding. Our protocol uses a cross layer approach where it uses both Application Layer and MAC layer.

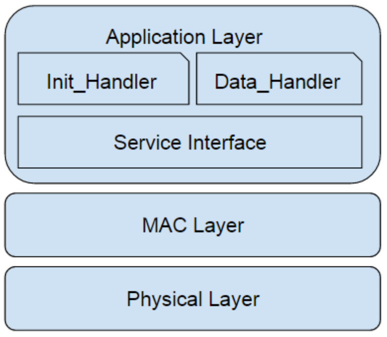

Fig 5 Protocol communication Stack

There are several key components in the application layer of the protocol.

- Init_handler
- Data handler
- Service Interface

Init_handler is responsible for handling new nodes. All the new nodes will communicate with Mother Mote and obtain a Node Id. This process is handled by the Init_Handler. Data_Handler is responsible for requesting sensory data from the device. Service Interface is responsible for passing information generated from Data_handler and Device details to the MAC Layer so that the MAC layer is able to send these data to the relevant Mother Mote.

V. Evaluation

In this chapter we have described how the evaluations were conducted and what the observations were. We evaluated protocol in few aspects.

- Binary file analysis
- Protocol performance analysis
- Theoretical analysis

In the Binary file analysis, we compare the implementation of our protocol with similar application implementation in other platforms. Protocol analysis have been conducted by using many parameters to address different use cases.

A. Binary file size comparison

In this section we will look at what are the binary file sizes of similar implementation for the ContikiOS and TinyOS. After implementing Protocol using Arduino platform we used USBasp device to program the ATtiny85 MCU. This MCU can hold up to an 8KB (8,192 bytes) file size. After the compilation of our protocol, binary file size was 2240 bytes. According to these values it is clear that our implementation easily fit to the ATtiny85 MCU.

According to Levis et al. TinyOS [23] Core is about 400 bytes. But this is only the Core of the operating system. Floating-point libraries of TinyOS is nearly 1024 bytes. An application which is capable of blinking a LED bulb would be size of 683 bytes. Authors have mentioned that even though a typical TinyOS application which uses the radio stack and generic timer would be 16KB, an application which is capable of transmitting data and has minimalistic features would be around 9KB. Size of large TinyOS applications such as DB query applications can go up to 64KB [23].

Figure 6 shows binary size comparison with Contiki and Tiny OS.

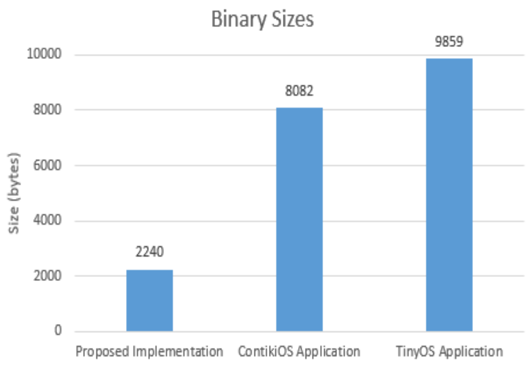

Fig 6 Binary size comparison

B. *Protocol performance evaluation*

Since we have implemented our protocol on top of the hardware we had to use real sensor nodes and mother motes for the experiments. We used following devices for the evaluation.

- Four Mother Motes
- Twenty Sensor Nodes
- Two laptops

Rest of this chapter explains how the experiments were conducted and what the observations were.

1. Evaluation using a single mother mote.

Initially we tested final protocol with a Single mother mote while having constant values for Total cycle time, new node time and data time. Variable values were the Window Size and number of nodes. For a given window time we tested the protocol behaviour. We used 10 sensor nodes.

- Sensor nodes under 1-meter radius.

To conduct this experiment, we placed sensor nodes within the 1-meter radius of the Mother Mote. In this experiment each sensor node transmitted 100 data packets to the mother mote. For this experiments following parameters were used.

- Ttotal: 5seconds
- Tdata: 4seconds
- Tnewnode: 1second
- Variable window lengths (in ms): 100, 200, 400, 600, 800, 1000

Results of this experiment are summerized in Figure 7.

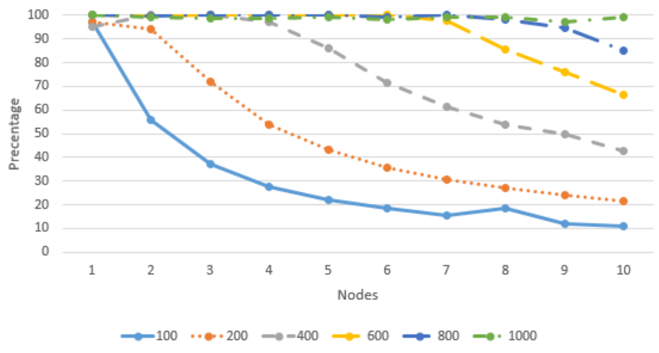

Fig 7 Packet Delivery Ratio under 1 meter radius.

In this experiment we observed that each node will take 115ms in average to successfully transmit a data packet. Therefore, the protocol will give over 99% PDR if a node can have 115ms Tframe size. If a node cannot have a sufficient Tframe size, there will be number of collisions.

- Sensor nodes under 3-meter radius.

In the previous experiments all the nodes were placed within the 1M range of the mother mote. In this we experimented how the nodes will behave if they were placed within the range of 1-3M. For this test we maintained the Ttotal, Tdata, Tnewnode same as the previous experiment. Results of this experiment is shown in the Figure 8.

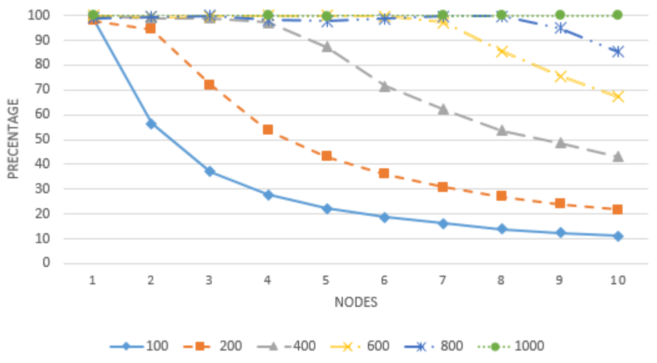

Fig 8 Packet Delivery Ratio under 3 meter radius.

In this experiment we tested for 6 window times similar to the previous example. In this experiment we have obtained similar results to the previous experiment. However, there is a slight decrease in the results up to the 400ms window time. Results of windows sizes which are larger than 600ms are identical to previous experiment.

- Sensor nodes under 5-meter radius

Since previous 2 experiments covered the range up to 3M, in this experiment we tested the behavior of the protocol when nodes are placed in the range of 3-5M. Again for this experiment we tested the behavior of the protocol for 6 different window sizes which were used in previous experiments. Results of this experiments are shown in the Figure 9.

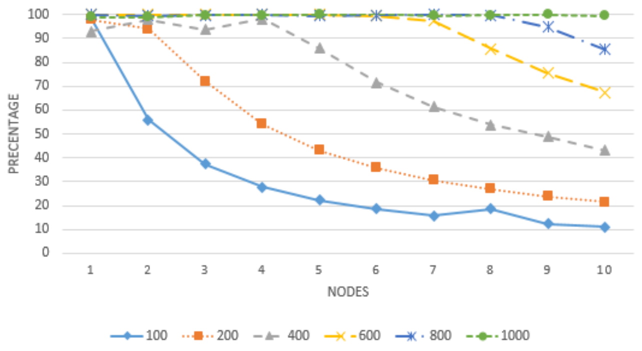

Fig 9 Packet Delivery Ratio under 5 meter radius

Results of this experiments were pretty much close to the results of the previous experiment. However in this experiment there were clear decrease in the PDR up to 400ms window time. Even though there is a clear decrease in results, up to 4 nodes it has been able to have a minimum PDR of 90% in the 400ms window time.

According to these three experiments it is clear that distance of the nodes will not affect the performance of the protocol in a major way. However there will be a slight impact on the performance of the protocol. Another conclusion from these three experiment would be that if there is a sufficient Tframe length, protocol will have over 99% Packet Delivery Ratio.

- Packet delivery with distance

Next experiment we carried out was to check how far packets will be delivered in an open environment. This experiment had two major considerations. One of them is open environment without any obstacle in the direct path and the other consern is that having a large obstacle in the direct path. For the first consideration we placed a single sensor node and the mother mote in an open area and tested for the packet delivery ratio for different distances. For the second consideration we placed the Mother mote inside a room. Distance between the wall and the mother mote was 3 meters. We placed the sensor node outside the room and tested for different distances.

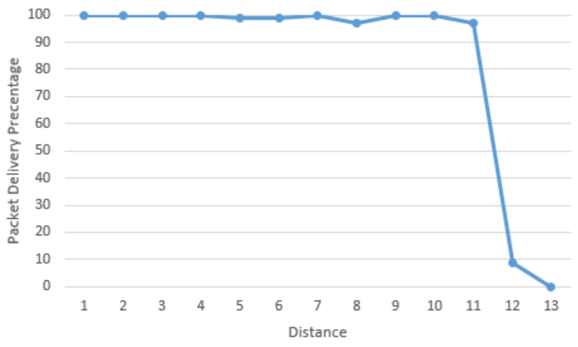

Fig 10 Packet delivery ratio with the distance. Without obstacles.

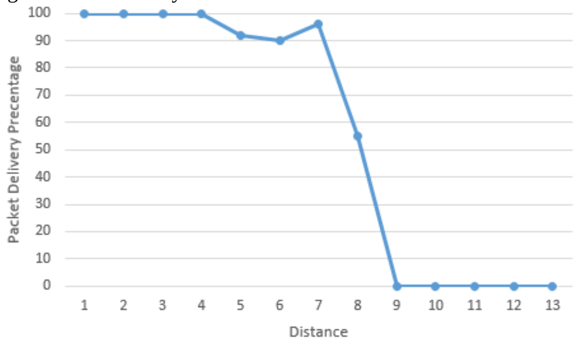

Fig 11 Packet delivery ratio with the distance. With an obstacle

Figure 10 shows the Packet Delivery Ratio when there is no obstacle between the sensor node and the mother mote. Figure 11 shows the packet delivery ratio when there is a large obstacle in between the sensor node and the mother mote.

nRF24L01 wireless transceiver has 4 operating power levels. For this scenario we tested only the lowest power level as one of the objective in this research to have low power consumption. For this experiments following parameters were used.

Ttotal: 5 seconds
Tdata: 4 seconds
Tnewnode: 1 seconds
Twindow: 200ms

When there are no obstacles in between, we observed that the packet delivery ratio is 100% up to 7 meters and till 11th meter it managed to have over 95% delivery ratio. At the 12th meter delivery ratio was 9% and no packets were delivered after that.

When there is an obstacle between Mother mote and Sensor Node, we observed that the delivery ratio will be 100% up to 4-meter range and had over 90% delivery ratio till 7-meter range. At 8m range, delivery ratio was 55% and packets were not delivered beyond that. Also we noticed that sometimes sensor node was not able to receive the beacon packet of the mother mote. Therefore, the total time of the experiment was larger than the initial experiment.

2. Evaluation using 2 mother motes

For the next evaluation we conducted the experiment with 2 mother motes and 10 sensor nodes. Each mother mote was assigned with 5 sensor nodes. Reason for us to place all the sensor nodes in the common receiver range of both mother motes was to observe how the protocol will handle the worst scenario. In this setup both mother motes will receive packets from the sensor nodes which are not assigned to it. Therefore, there is a higher probability for have high collision rate.

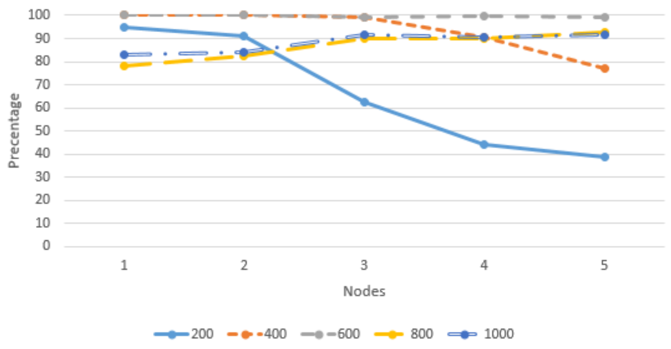

Fig 12 Average Packet Delivery Ratio for 2 Mother motes.

Figure 12 shows the average packet delivery ratio for both mother motes. According to the results packet delivery ratio is behaving similar to the previous experiments except for the window sizes of 800ms and 1000ms. Reason behind this is the Mother mote 1 is having the expected behavior and because of the transmissions of sensor nodes which are assigned to Mother mote 1, some packets which are assigned to Mother mote 2 will be discarded. Conclusion of this experiment would be that even though all the sensor nodes were placed in common receiver range of both mother motes, it is able to achieve higher packet delivery ratio. This is the worst case scenario which we expected to have most number of collisions. Therefore we can expect a better performance if sensor nodes are placed where there will not be a common receiving range for both mother motes.

3. Evaluation using 3 mother motes

For this experiment we used three mother motes. For each mother mote we assigned 5 sensor nodes. All the sensor nodes were placed in common receiving range of all three mother motes. By this placement we expected to have most number of collisions in the experiment.

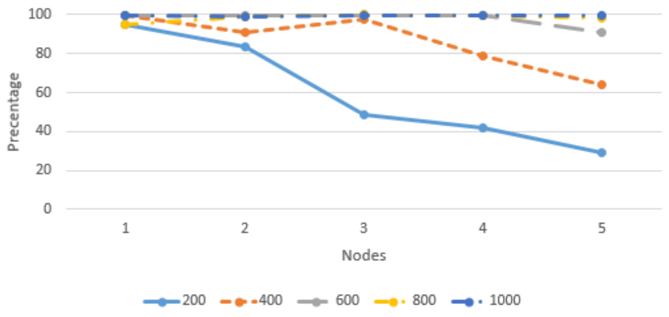

Fig 13 Average Packet Delivery Ratio for 3 Mother Motes.

Figure 13 shows the average results per mother mote. In average there is over 95% packet delivery ratio for up to 5 nodes per mother mote when using a windows size of 600ms. If used a 400ms windows size it is clear that over 90% packet delivery ratio can be obtain up to 3 sensor node per each mother mote. Window size of 200ms will not give a packet delivery ratio than 80% if there are over 2 sensor nodes per mother mote.

According to these observations we can conclude that the protocol will achieve 100% packet delivery ratio if used a window size of 1000ms even though there are 5 sensor nodes per mother mote (15 sensor nodes altogether in the network).

4. Evaluation using 4 mother motes

In a previous experiment we showed that a sensor node will be able to communicate with a mother mote up to a distance of 7 meters while maintaining a packet delivery ratio over 90%. By placing sensor nodes and mother motes carefully, it is able to cover a much larger space. In our research we are focusing on small environments such as homes and small offices. By using up to 4 mother motes it is able to cover a great deal of an environment for nodes to maintain a higher packet delivery ratio. In this experiment we are using 4 Mother motes and 20 sensor nodes (5 sensor node assigned to each mother mote).

Results of this experiment is shown in the Figure 14.

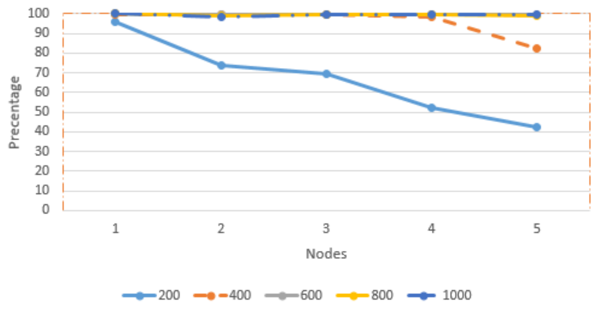

Fig 14 Average Packet Delivery Ratio for 4 mother motes.

According to these results it is clear that protocol has achieved over 99.1% packet delivery ratio for window sizes of 600ms, 800ms and 1000ms.In the 400ms window size protocol manage to have over 98.43% delivery ratio up to 16 nodes (4 nodes per each mother mote).

Evaluations including this experiment shows that the protocol manage to achieve higher data rates (Over 98%) in most of tests with over 400ms windows size.

Number of nodes in the network directly affect the packet delivery ratio. Depending on the application, a user can select an appropriate window size to get the maximum performance of the protocol. With our evaluations we have shown that this protocol will perform well up to 4 Mother Motes and up to 20 Sensor nodes very well in a node placement where all the nodes resides in a common receiving range for all the mother motes. We can assume by placing sensor nodes intelligently it is able to have a higher data packet delivery ratio for most of the scenarios.

5. Duty Cycle Analysis

In the protocol, sensor node will listen to the channel till a beacon packet arrives. After that it will sense and transmit data according to his schedule. Therefore for a given cycle, there will be 1+NumberOfDataPackets number of packets will be transmitted. Duty Cycle percentage can be theoretically obtain according to the following equation.

$$Duty\ Cycle\ \% = \left(\frac{(1 + NPackets) * TtxTime}{Ttotal}\right) * 100\%$$

$$Npackets = \frac{Tdata}{Twindow}$$

According to the above equation if we keep Tdata and Ttotal values constant, with the larger Twindow sizes we should get lover Duty cycles. Let's consider the following conditions.

- Ttotal: 5s
- Tdata: 4s
- TtransmissionTime: 6ms
- Twindow: 100ms

For these values we should get a DC percentage as 4.92%. If we change the Twindow to 1 second, we will get a DC percentage as 0.6%. However transmission time dep ends on other conditions as well. In real situation transmission time continuously changes. Observations are shown in Figure 15.

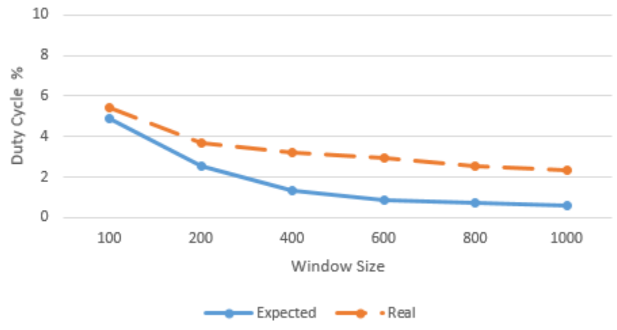

Fig 15 Duty Cycle ratio with window size

Our protocol has a duty cycle percentage below 5 %. We can compare this value with other protocols.

According to the study conducted by Tang et al [17]. X-MAC and RI-MAC have 60% duty cycle percentage and WiseMac has above 10% duty cycle percentage. Therefore it is clear that our protocol achieves lower duty cycle values than few other protocols. However, ContikiMAC, PW-MAC outperforms our protocol as they have even lower duty cycle percentages.

VI. Conclusion

According to the evaluations carried out, it is clear that our proposed protocol offers good results in extremely resource constrained devices. Even though other protocols offer higher data rates, those protocols are implemented leveraging the complete IPv6 networking stack. Our protocol does not use the IPv6 stack and therefore eliminate the overhead of using a real time OS such as Contiki or TinyOS on top of the sensor node hardware. This allow us to save memory as well as the energy. According to the binary file size evaluation, our entire protocol implementation size is not more than 25% of the Contiki implementation of similar application.

Most widely used protocols such as X-MAC and RI MAC have a Duty Cycle percentage close to 60%. This means the wireless transceiver of the device has to be on operating mode for 60% of the entire communicating time. According to the evaluation of our protocol, its average

Duty Cycling percentage is lower than above given protocols. Therefore, it is clear that our protocol is capable of saving energy in a considerable manner while operating on top of extremely resource constrained devices.

When considering the packet delivery ratio, most of the available MAC protocols are designed well. Such protocols achieve Packet Delivery Ratio above 95% most of the time. However X-MAC and WiseMAC protocols have over 95% PDR only for 2 hop communication. If they are having multiple hops PDR is drastically reduced. In the evaluations we observed that our protocol has over 98% PDR most of the use cases. One of the disadvantage of our protocol is the data rate. High data rates are possible but it will affect the PDR directly. However, for environment monitoring applications, average data rates are sufficient. According to these observations we can conclude that our protocol can used efficiently in many applications which are focused in small environments such as offices and homes.

VII. Future Work

We have implemented the protocol on top of the hardware and evaluation was limited to 20 nodes. Even though this number is sufficient to evaluate protocol for small environments. Using this number of nodes, results of protocol performance cannot be generalized to cover dense networks. To evaluate protocol for dense networks, a network simulator such as NS2 can be used. Evaluating this protocol for dense networks can be conducted as another research. This protocol has been evaluated for small environments such as homes and small offices. Another research extension would be to evaluate this protocol in larger environments and test the protocol behaviour.

We have used several libraries provided by the Arduino community for the implementation such as RF24, Mirf, SPI and nRF24L01. Some of the APIs provided by the libraries have not been used in the implementation and there is a possibility that the overhead of those libraries affect the performance of the protocol. Therefore, another extension would be to evaluate the library overhead and explore how to reduce the library overhead and improve protocol performance.

References

[1] Mainwaring, A., Culler, D., Polastre, J., Szewczyk, R., & Anderson, J. (2002). Wireless Sensor Networks for Habitat Monitoring. *Proceedings of the 1st {ACM} International Workshop on Wireless Sensor Networks and Applications,* 88–97. doi:10.1145/570738.570751

[2] Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., & Cayirci, E. (2002). *Wireless sensor networks: a survey. Computer Networks*, 38, 393–422. doi:10.1016/S1389-1286(01)00302-4

[3] K. J. Hintz and D. Tabak, *Microcontrollers: architecture, implementation, and programming*. McGraw-Hill Professional, 1992

[4] Ye, W., Heidemann, J., & Estrin, D. (2002). An energy-efficient MAC protocol for wireless sensor networks. *Proceedings of the Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies,* 3, 1567–1576. doi:10.1109/INFCOM.2002.1019408

[5] P. S. Purushotham BV and K. Ganesan, “Study of reliable data communication in wireless sensor networks*,” in Proceedings of The International MultiConference of Engineers and Computer Scientists,* pp. 777–781, 2008

[6] S.-J. Park and R. Sivakumar, “Sink-to-sensors reliability in sensor networks,”*SIGMOBILE Mob. Comput. Commun. Rev., vol. 7,* pp. 27–28, July 2003

[7] C.-Y. Wan, A. T. Campbell, and L. Krishnamurthy, “Psfq: A reliable transport protocol for wireless sensor networks,” *in Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications, WSNA* ’02, (New York, NY, USA), pp. 1–11, ACM, 2002.

[8] Yahya, B., & Ben-Othman, J. (2009). Towards a classification of energy aware MAC protocols for wireless sensor networks. *Wireless Communications and Mobile Computing,* 9, 1572–1607. doi:10.1002/wcm.743

[9] G. Anastasi, M. Conti, M. Di Francesco, and A. Passarella, “Energy conservation in wireless sensor networks: *A survey,” Ad Hoc Netw., vol. 7*, pp. 537–568, May 2009

[10] C. S. R. Murthy and B. Manoj, *Ad Hoc Wireless Networks: Architectures and Protocols*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2004.

[11] R. R. Lanjewar and D. S. Adane, *“Comparative study of MAC layer protocols in wireless sensor networks: A survey,”* CoRR, vol. abs/1406.4701, 2014.

[12] S. Kumar, V. S. Raghavan, and J. Deng, “Medium access control protocols for ad hoc wireless networks: A survey,” *Ad Hoc Netw.*, vol. 4, pp. 326–358, May 2006

[13] R. Jurdak, C. V. Lopes, and P. Baldi, “A survey, classification and comparative analysis of medium access control protocols for ad hoc networks,” *Commun. Surveys Tuts., vol. 6,* pp. 2–16, Jan. 2004.

[14] J. Polastre, J. Hill, and D. Culler, “Versatile low power media access for wireless sensor networks,” *in Proceedings of the 2Nd International Conference on Embedded Networked Sensor Systems*, SenSys ’04, (New York, NY, USA), pp. 95–107, ACM, 2004

[15] J. Kabara and M. Calle, “Mac protocols used by wireless sensor networks and a general method of performance evaluation.,” *International Journal of Distributed Sensor Networks*, vol. 2012, 2012

[16] Y. Sun, O. Gurewitz, and D. B. Johns on, “Ri-mac: A receiver-initiatedasynchronous duty cycle mac proto col for dynamic traffic loads in wireless sensor networks,” *in Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems*, SenSys ’08, (New York, NY, USA), pp. 1–14, ACM, 2008

[17] Tang, L., Sun, Y., Gurewitz, O., & Johnson, D. B. (2011). PW-MAC: An energy-efficient predictive-wakeup MAC protocol for wireless sensor networks. *INFOCOM, 2011 Proceedings IEEE*, 1305–1313. doi:10.1109/INFCOM.2011.5934913

[18] A. Dunkels, N. Eriksson, F. Osterlind, and N. Tsiftes, “The Contiki Oper- ating System,” Web page Visited, 2006

[19] G. Chalhoub, R. Diab, and M. Misson, “Hmc-mac protocol for high data rate wireless sensor networks,” Electronics, vol. 4, no. 2, p. 359, 2015.

[20] Demirkol, I., Ersoy, C., & Alagöz, F. (2006). MAC protocols for wireless sensor networks: A survey. IEEE Communications Magazine, 44, 115–121. doi:10.1109/MCOM.2006.1632658

[21] El-Hoiydi, A., & Decotignie, J.-D. (2004). WiseMAC: an ultra low power MAC protocol for the downlink of infrastructure wireless sensor networks. *In Proc. 9th International Symposium on Computers And Communications (ISCC 2004)* (Vol. 1, pp. 244–251). doi:10.1109/ISCC.2004.1358412

[22] Burri, N., von Rickenbach, P., & Wattenhofer, R. (2007). Dozer: Ultra-Low Power Data Gathering in Sensor Networks. *In 2007 6th International Symposium on Information Processing in Sensor Networks* (pp. 450–459). doi:10.1109/IPSN.2007.4379705

[23] P. Levis, S. Madden, J. Polastre, R. Szewczyk, K. Whitehouse, a. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer, and D. Culler, “TinyOS: An operating system for sensor networks,” Ambient Intelligence, pp. 115–148, 2005

[24] A. Sayakkara, C. Suduwella, C. Shalitha, R. Hapuarachchi, K. D. Zoysa, "Wireless Sensing: What Simplicity Has to Offer?", *Mobile Ad Hoc and Sensor Systems (MASS),* 2015 IEEE 12th International Conference on, Dallas, TX, 2015, pp. 475-476.