IEEE—ICET 2006
2nd International Conference on Emerging Technologies
Peshawar, Pakistan 13-14 November 2006

Comparison of S-MAC & TDMA-W Protocols for Energy Efficient Wireless Sensor Networks

Muhammad Zohaib, Tariq M. Jadoon
Department of Computer Science and Engineering
Lahore University of Management Sciences, Pakistan.
{mzohaib, jadoon}@lums.edu.pk

Abstract – *Energy efficiency is a major consideration while designing wireless sensor network nodes. Most sensor network applications require energy autonomy for the complete lifetime of the node, which may span up to several years. These energy constraints require that the system be built such that each component consumes minimum possible power. Consequently, the main objective is to devise strategies for energy reduction in different physical components while implementing all layers of the network protocol stack within a node. This paper compares S-MAC (a random access MAC Protocol) and TDMA-W (a scheduled MAC Protocol) for Wireless Sensor Networks in terms of their energy efficiency. The Network Layer protocol is assumed to be fixed for this comparison and is the Energy Aware Routing Protocol. This paper presents initial results while comparing different S-MAC and TDMA-W schemes.*

Keywords – Wireless Sensor Networks, Energy Efficiency, OMNeT++, S-MAC, TDMA-W, Energy Aware Routing.

1. INTRODUCTION

Wireless sensor networks are a collection of small nodes, connected through a wireless channel, that perform the task of monitoring the environment and processing and communicating the gathered information. The main function of sensor networks is to sense the environment for physical data and to collate this data at a centralized location. This may involve a hierarchical structure, in which certain nodes summarize or fuse data as they forward it. The nodes consist of sensors, a radio transceiver, battery and an embedded processor. The main applications of sensor networks include industrial control, asset tracking and supply chain management, environmental sensing, health monitoring and traffic control [1].

Wireless sensor networks are different from traditional ad hoc networks in certain aspects such as being highly energy efficient, having low traffic rate, low mobility and predefined traffic patterns [2]. Energy conservation is a critical factor for the design of these networks. Energy conservation entails optimizing protocols at various layers of the protocol stack. Optimal placement of nodes in sensor field also contributes to energy conservation but typically the philosophy in wireless sensor networks is to have a large number of cheap nodes which could fail and thus resilience in the distributed protocols is preferred over strategies such as an optimal placement of nodes.

Each layer of the network stack consumes energy depending upon the protocol implemented at that layer. This makes it important to study the effects of choosing a certain protocol at a layer and investigating its impact on protocols in other layers. This paper compares two MAC layer protocols, S-MAC (a random access MAC Protocol) [3] and TDMA-W (a scheduled MAC protocol) [2] while assuming a fixed network layer protocol i.e. Energy Aware Routing [4]. We have simulated these protocols in OMNeT++ [5]. The flexibility of this simulation platform for new protocol implementations and its hierarchical granularity make it an ideal choice for simulating wireless sensor networks.

The rest of the paper is organized as follows. Section 2 describes the MAC layer protocol design and details of our implemented protocols. Details of the network layer protocol have been included in Section 3. Simulations and results are presented in Section 4. Section 5 concludes the paper by summarizing the results and providing directions for the future work.

2. MAC LAYER DESIGN

The Medium Access Control (MAC) layer is responsible for access to the shared medium. MAC protocols, therefore, assist nodes in deciding when to access the channel. MAC layer

1-4244-0502-5/06/$20.00©2006 IEEE

protocol design has a great impact on the energy consumption of sensor nodes, because radio communication is a major source of power consumption and the MAC layer design directly governs the rules of transmitting and receiving over the wireless medium using the radio.

In addition to transmitting and receiving data packets energy consumption at the MAC layer can be attributed to four sources – *collision, idle listening, overhearing* and *control packet overhead* [6]. When two frames are sent simultaneously over the medium, *collision* occurs and frames are corrupted. These frames have to be resent and this consumes further energy. *Idle listening* occurs when the radio of node is in the wakeup state but is not sending or receiving useful data. *Overhearing* involves receiving unnecessary data, i.e. data that is not meant for the node itself. *Control packets* for reservation of the channel and synchronization of nodes are another source of energy consumption.

MAC protocols can be classified into *scheduled (TDMA based)* and *random access (Contention based)* protocols. *TDMA (Time Division Multiple Access)* mechanisms divide the channel into time slots of fixed duration. These slots are repeated in a fixed cycle. A complete cycle of these slots is referred to as a frame. These slots are assigned to nodes and nodes can transmit and receive during their allotted slots. TDMA protocols are inherently energy conserving as they reduce wastage due to *collision*, *idle listening* and *overhearing*. However, the disadvantage is that these protocols require *frequent synchronization* and are not easily *scalable* and *adaptable*. In *random access protocols*, the channel is allocated to nodes on demand instead of using fixed pre-allotted slots. Nodes contend for the channel and whoever grabs the channel, finds opportunity to transmit. The basic contention resolution mechanism in these protocols is *Carrier Sense Multiple Access (CSMA)*, which dictates that a node should listen to channel before transmitting. If the channel is busy, the node should defer its transmission until the channel becomes idle. These protocols are *scalable* and easily adjust to *topology changes*. Furthermore, there is *no strict time synchronization* constraint. The disadvantage however, is that they incur significant control packet overhead.

2.1 S-MAC Protocol

Sensor MAC (S-MAC) [3] is a contention based protocol specifically designed for wireless sensor networks. Its basic principle is *CSMA/CA* (Carrier Sense Multiple Access with Collision Avoidance) and operates similar to the 802.11 Distributed Coordination Function (DCF) mode [7]. It also follows periodic listen and sleep patterns like the 802.11 power saving mode. The major mechanisms applied in S-MAC for energy efficiency are: *Periodic Listen and Sleep, Collision Avoidance, Overhearing Avoidance, Message Passing and Adaptive listening.*

S-MAC uses a *periodic sleep-wakeup cycle* which is known as a frame. Each frame begins with a wakeup period during which nodes exchange data and control frames. The wakeup period is followed by sleep period during which nodes turn off their radio if they have no data to send. If a node wishes to transmit packet while in the sleep state, it delays transmission until its next scheduled wakeup slot. Before each data frame transmission, *Physical* and *Virtual Carrier Sense* mechanisms are used for determining the state of the channel. An RTS-CTS-Data-ACK procedure is followed for each unicast data frame transmission. Broadcast frames are sent without the use of RTS, CTS and ACK frames. In S-MAC it is not necessary for all the nodes to be *perfectly synchronized* and they may adopt different listen-sleep schedules. These schedules are then exchanged between nodes to ascertain when to transfer data to a neighbouring node. A node only sends data to its neighbour if the neighbour is in the wakeup state. Otherwise, it waits for its wakeup interval. Synchronized sleep-wakeup pattern among neighbouring nodes prevents control packet overhead.

The basic *collision avoidance* mechanism is identical to 802.11's DCF mode. If a node fails to access the medium, it goes into sleep mode until its Network Allocation Vector (NAV) indicates that the medium is free. The protocol achieves *overhearing avoidance* such that when a node receives a packet that is meant for another node, it updates its NAV based on the duration field in the received frame. The node then sleeps until its NAV becomes zero i.e. for the time during which other nodes are exchanging data.

The basic mechanism of *message passing* is similar to that of *fragmentation* in 802.11's DCF mode. A single RTS and a single CTS are used for transferring all the fragments of a given message. However, message passing is somewhat different from fragmentation in its duration for the reservation of the medium. The RTS and CTS frames reserve the medium for all the fragments and their ACKs. If an ACK is lost, the sender reserves the medium for extended time by changing the duration field in subsequent transmitted frames. Consequently, the nodes which need to gain control of the channel have to wait longer as compared to fragmentation. This scheme reduces delay for the receiver if it immediately requires sending all fragments to upper layer protocols. In this way, message passing compromises fairness among nodes for energy saving and message level latency.

Adaptive listening is used to improve latency of S-MAC. Instead of waiting for their next wakeup slots, the neighbour nodes of both sender and receiver wake up for a short time immediately after the current transmission. They remain in the wakeup state and immediately contend for the channel if they need to send data.

2.1.1 Implementation Details

We have assumed *perfect synchronization* of nodes i.e. all nodes follow the same sleep-wakeup cycle. The sleep-wakeup frame length is set to be equal to 1 sec. We have implemented *message passing* and *overhearing avoidance* functionality in our simulations. MAC layer frame sizes are based on IEEE 802.11 DCF: RTS – 20 bytes, CTS – 14 Bytes, ACK – 14 bytes and Data Frame – 34 bytes + Network Layer Packet Size.

2.2 TDMA-W Protocol

TDMA-Wakeup (TDMA-W) protocol [2] is a scheduled protocol based on Time Division Multiple Access mechanism. It combines the advantages of scheduled protocols with wakeup timers to allow nodes to turn off their radio even in their assigned transmission slots, in case they do not have any data to send. Contrary to typical TDMA protocols in which each node is assigned only one slot to transmit and receive in a frame, this protocol schedules two slots for each node in a complete frame. These slots are known as the *Transmit/Send Slot (S-Slot)* and the *Wakeup Slot (W-Slot).*

A node listens to channel in its allotted *W-Slot* in every frame. In case this node is addressed by another node that has data to send, it starts listening to channel during the S-Slot associated with the sender. A single W-Slot can be shared by more than one node if the nodes are more than two hop distance apart. A node turns on its radio in an *S-Slot* if it has data to send or is addressed by another node. It remains in sleep state in rest of the slots.

In order to send and receive data from other nodes, the nodes should have knowledge about the slot timings of the neighbours. The selection and sharing of these slot timings is performed by nodes using a *Self-Organization scheme*. This scheme helps nodes in identifying their neighbours and choosing their slots accordingly to avoid collision. Nodes randomly select their slots and then negotiate with their neighbours to see whether they can be assigned these slots or not.

After successfully choosing their slots and finding those of others, the nodes exchange data frames using the *TDMA-W channel access protocol*. Nodes listen to the channel in their assigned W-Slot. Each node also maintains pairs of counters for every other node in the network. An *Incoming Counter* is used for keeping track of nodes from which a node receives data. This counter is initialized to a positive value. In every TDMA-W cycle, if a node receives a frame from another node, it resets the counter to its initial value, otherwise it decrements the counter. When a certain Incoming Counter becomes zero or negative, the node stops listening to the channel in the corresponding S-Slot from the next TDMA-W frame. An *Outgoing Counter* is also initialized to a positive value. If no frame is sent to a node in a particular TDMA-W cycle, the corresponding Outgoing Counter is decremented. This counter is initialized to its starting value with each successful transmission.

When a node has data to send to a particular node, it checks the value of the corresponding

Outgoing Counter. If the counter is less than or equal to zero, the node is considered to be in sleep state in the S-Slot of the sender, and a wakeup message is sent to the node during its corresponding W-Slot. The sender starts sending data to the node during its S-Slot from the next TDMA-W frame. If the counter is positive, a wakeup message is not sent and the sender transmits data in its S-Slot.

On receiving a wakeup message, the node turns on its radio in the S-Slot of the sender in the next TDMA-W frame. If more than one wakeup messages are received in a single TDMA-W frame, a collision occurs and the node starts listening to all its neighbours from the next frame in order to identify the senders.

2.2.1 Implementation Details

Our implemented protocol assumes a *perfect synchronization* of nodes and does not include the *self organization scheme*. We have assigned unique S-Slot and W-Slot to each node. The TDMA-W *cycle length* is set to 1 sec. MAC frame sizes are: Wakeup Frame – 20 bytes and Data Frame – 34 bytes + Network Layer Packet Size.

In this paper S-MAC and TDMA-W protocols have been compared using different parameters while assuming a fixed network layer routing protocol, i.e. energy aware routing.

3. NETWORK LAYER PROTOCOL DESIGN

The major functions of the network Layer are *routing* and *addressing* of nodes. Since energy efficiency is the major requirement for sensor networks, *routing protocols* should be designed such that they use the *lowest energy consuming path* between nodes. Moreover, these protocols must *maximize network life time* by distributing traffic in a way that a single route is not continuously used, thus avoiding depletion of energy resources on this path.

Routing protocols can be divided into *Proactive (Table-driven) Routing Protocols* and *Reactive (Demand-driven) Routing Protocols*. Examples of *Proactive* protocols include Destination Sequenced Distance Vector (DSDV) Protocol and Link-State Routing Protocol, whereas, Ad-Hoc On-Demand Distance Vector Routing (AODV), Dynamic Source Routing (DSR), Directed Diffusion and Energy Aware Routing are the examples of *Reactive* Protocols.

3.1 Energy Aware Routing

Energy Aware Routing is a destination-initiated reactive protocol, in which the destination node establishes routes by sending route request packets towards the source node. As described above, *network survivability* is a major issue while designing an energy efficient network layer protocol. Routing protocols may find the minimum energy route and use that path for every transmission. Using the same path frequently results in depletion of energy sources on this path, which in the worst case may lead to network partition. Energy Aware Routing solves this problem by distributing traffic amongst various routes rather than exploiting the same low energy route.

The protocol finds multiple paths between the source and destination and assigns a probability to each of these paths depending upon a cost metric along this path. The probability of selecting a path is inversely proportional to the cost of the path. While sending data, each node probabilistically chooses the path among the available paths. Therefore, paths with a low cost metric have higher chances of being chosen.

The major steps involved in this protocol include *Setup (Interest Propagation) Phase, Data Communication (Data Propagation) Phase and Route Maintenance Phase*. During *Setup phase* directional flooding occurs from the destination to the source to find the costs of all the routes from source to destination and routing tables are established at each node. During *Data Communication phase,* data is sent from source to destination using information about path costs and probabilities in the forwarding table. A source node randomly sends a data packet to one of its neighbours depending upon the probability of the neighbour in the forwarding table. Each intermediate node follows the same steps until the packet finally reaches the destination. In *Route Maintenance phase*, route requests are flooded through the network to refresh paths from destination to source and to update routing costs.

Energy Metric used in this protocol is a function of both the transmission energy of the path as well as the residual energy at the receiving node. As described in [4], this metric between nodes *i* and *j* is calculated as:

$$C_{ij} = e_{ij}^{\alpha} R_i^{-\beta} \quad (1)$$

e_{ij} is the sum of energy used to transmit and receive over the link. R_i is the residual energy of the receiving node normalized to its initial energy. Parameters α and β can be adjusted to find the minimum energy path or the path with nodes having the most residual energy or a combination of both as the requirements may be.

3.1.1 Implementation Details

We have set α in *energy metric* to zero in our simulations. Transmission energy used in this metric depends on the distance between nodes. In our case, the distance between neighbouring nodes is one hop; therefore transmission energy has no effect on the metric. Our cost metric only depends on the residual energy of the nodes.

$$C_{ij} = R_i^{-\beta} \quad (2)$$

The lengths of packets at Network layer are: Route Request Packet – 32 bytes and Data Packet – 32 bytes + Application Layer Message Size.

4. SIMULATION AND RESULTS

We have extended the functionality of Louisiana State University (LSU) SensorSimulator 3.1 [8] in OMNeT++ for our simulation. SensorSimulator uses the concept of simple and compound modules of OMNeT++ for designing sensor networks. The protocol layers of a node as well as hardware components behave as simple modules, while the node itself is a compound module. All the nodes combine to give a system module or network [9]-[10]. The protocol stack implements *Application, Network, MAC* and *Physical* layers. Hardware components include *Battery, Radio* and *CPU.* Different layers of the node communicate with each other using messages through gates. These messages represent data units at different layers. A *Coordinator* module is implemented to assist hardware components and protocol layers to communicate with each other. The node structure is depicted in Figure 1. These nodes communicate with each other through a *wireless channel* simple module.

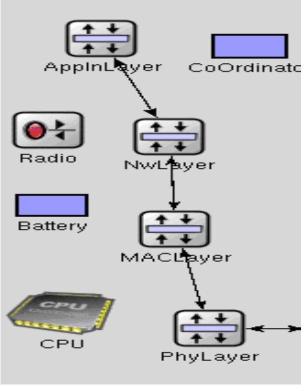

Figure 1 – Node Structure

The simulated network consists of *36 sensor nodes* distributed uniformly over a *120×120 m*^{2} grid. The distance between two adjacent nodes is *20 m*. The node at the extreme top left corner of the network is marked as the source node which sends data to the destination node according to traffic model described ahead. Destination node is placed three hops diagonally away from the source node. The radio range of the nodes is adjusted such that a node can communicate with its immediate neighbours only. This range is set to be equal to *30 m*. Therefore, nodes can also communicate with their diagonally positioned neighbours. The topology of the modeled network is shown in Figure 2.

Energy Aware Routing Protocol dictates that destination node establishes routes towards the source node by sending route request packets. In our model these route request packets are sent by destination after every *60 seconds* in order to refresh the routes. We have assumed that source node contains light sensor, which sends data after a fixed interval of time to the destination node. The data generated at application layer of the source node is 2 bytes in length [1].

The radio consumes energy in different states depending on the *power rating* in that state. These states are defined to be TRANSMIT, RECEIVE, SLEEP and IDLE. Power ratings in these states are 8.25mA, 4.5mA, 5μA and 4.5mA respectively. Radio Data Rate is set to 40 kbps [11]. Our simulations assume that the energy consumption by the CPU is the same irrespective of the protocol. All the simulations are run for 4320 seconds.

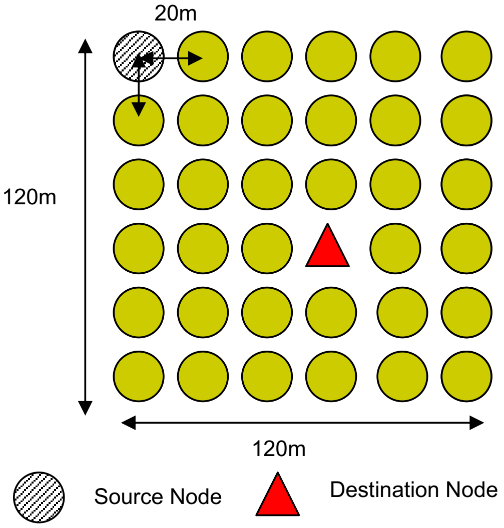

Figure 2 – Simulated Network Topology

What is interesting to note is that each protocol has a number of tuneable parameters and what is interesting to ask is firstly would varying these parameters have an effect on the energy consumption of nodes and if so what set of parameters in what particular protocol would one need to set in order to get the best performance. We have implemented 3% and 5% S-MAC protocols in our simulations. The percentage indicates the ratio of the wakeup time to the complete cycle time. It is not possible to implement an S-MAC protocol with a duty cycle lower than 3% because the transmission delay for data frames and random access delay for acquiring the channel is greater than the wakeup period corresponding to these values. The value of counter in TDMA-W protocol has been changed to 2, 4 and 6. A maximum frame length of 68 bytes and data rate of 40 kbps result in a slot time equal to 13.6 msec and consequently 72 slots per TDMA cycle. Parameter β in Energy Aware Routing Protocol is varied to 1, 10 and 100 for each of these parameters at the MAC Layer. Each set of parameters is simulated 5 times with different random number seeds and 95% confidence interval calculated. The consumed energy shows very low variation. In order to compare the two MAC protocols, we have studied the effect of traffic load on energy consumption. For both the protocols, the time between the generating successive sensor data at the application layer is varied from 10 sec to 30 sec in increments of 5 sec. With decreasing load i.e. increasing inter-arrival time, both the protocols consume less energy as expected.

Figure 3 contains graphs of average energy consumption of nodes versus the inter message generation time at the application layer of source node for 3% S-MAC as well as for different values of Counter in TDMA-W protocol. Figure 4 compares 3% and 5% S-MAC protocols in terms of energy consumption. β is set to 1 in these simulations. The results are found quite similar for other values of β. TDMA-w protocol consumes more energy with increasing counter value, while in S-MAC energy consumption increases with increasing duty cycle.

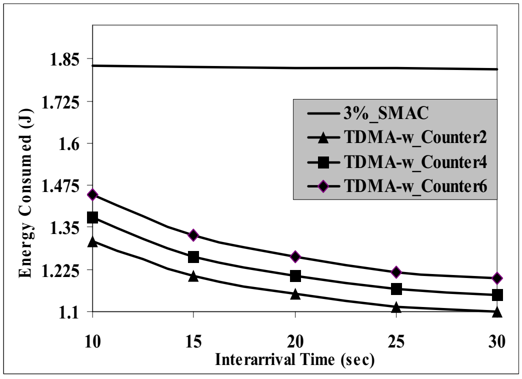

Figure 3 – Average Energy Consumed vs. Interarrival Time

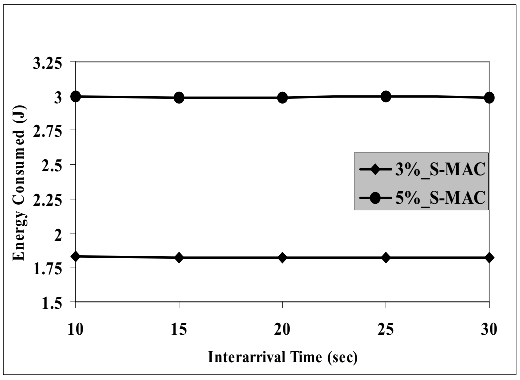

Figure 4 – Average Energy Consumed vs. Interarrival Time

It is evident that for the set of simulation parameters chosen, the TDMA-W protocol consumes less energy than the S-MAC protocol. Our results agree with the results presented in [2].

5. CONCLUSIONS

There is a great choice in rendezvous schemes at the MAC layer as well as routing protocols at the network layer. Furthermore, each protocol has number of tuneable parameters and choosing an optimal set of protocols and parameters is a hard problem and requires further investigation. One approach is through extensive simulations whereas; another might be developing and optimizing cross layer models of the entire protocol stack as a whole rather than a collection of individual layers. This diverges from the classical layered approach to performance modeling in networks.

REFERENCES

[1] S. Mahlknecht, “Energy-Self-Sufficient Wireless Sensor Networks for the Home and Building Environment,” Ph.D. Thesis, Institute of Computer Technology, Vienna University of Technology, Austria, September 2004.

[2] Z. Chen, and A. Khokhar, “Self Organization and Energy Efficient TDMA MAC Protocol by Wake Up For Wireless Sensor Networks,” *IEEE SECON 2004,* August 2004.

[3] W. Ye, J. Heidmann, and D. Estrin, “An Energy-Efficient MAC Protocol for Wireless Sensor Networks,” *in Proceedings of the IEEE Infocom,* NY, June 2002, pp.1567-1576.

[4] R. C. Shah, and J. M. Rabaey, “Energy Aware Routing for Low Energy Ad Hoc Sensor Networks,” *Proc. IEEE WCNC’02,* 2002.

[5] http://www.omnetpp.org

[6] W. Ye and J. Heidmann, “Medium Access Control in Wireless Sensor Networks,” *USC/ISI Technical Report ISI-TR-580,* October 2003.

[7] LAN MAN Standards Committee of the IEEE Computer Society, “Wireless LAN medium access control (MAC) and physical layer (PHY) specification,” IEEE, NY, USA, *IEEE 802.11-1999 edition*, 1999.

[8] http://csc.lsu.edu/sensor_web/simulator.html

[9] C. Mallanda, A. Suri, V. Kunchakarra, S.S. Iyengar, R. Kannan and A. Durresi, “Simulating Wireless Sensor Networks with OMNeT++,” *Sensor Network Research Group, Department of Computer Science, Louisiana State University, Baton Rouge, LA*, 2005.

[10] A. Suri, V. Kunchakarra, S.S. Iyengar, and R. Kannan, “LSU SensorSimulator User Manual Version 1,” *Sensor Network Research Group, Department of Computer Science, Louisiana State University, Baton Rouge, LA*, 2005.

[11] H. Tseng, S. Yang, P. Chuang, E. Wu, and G. Chen, “An Energy Consumption Analytic Model for A Wireless Sensor MAC Protocol,” *in Proceedings of the IEEE Vehicular Technology Conference*s, *LA,* 2004.