Energy Efficient Sleep Scheduling for Wireless Sensor Networks

Paul Chiedozie Uzoh, Jilong Li, Zhenbo Cao, Jinbae Kim,
Aamir Nadeem, and Kijun Han(✉)

Department of Computer Science and Engineering, Kyungpook National University,
Buk-gu, Daegu, 41566, South Korea
{Upaul,jllee,zbcao,jbkim,anadem}@netopia.knu.ac.kr,
kjhan@knu.ac.kr

Abstract. Recently, wireless sensor network (WSN) has become a formidable force in almost all areas of our everyday life. However, there are still many issues with wireless sensor networks among which is the energy problem. One of the numerous techniques that have been introduced as a way of solving the problem of energy deficit inherent in a WSN is sleep scheduling. Sleep scheduling allows sensor nodes to periodically take turns to sleep in order to minimize energy cost in a WSN. Apparently, Overemission in a WSN is one of the main causes of energy drainage in sensor nodes. Traditional schemes fail to take into consideration the sleeping timetable of other nodes; hence they let transmitter nodes repeatedly send RTS preamble packets and similar control packets to sleeping nodes. Our proposed sleep scheduling scheme addresses this problem from a whole new perspective using a system called Designated Sensor Node (DSN) mechanism. The DSN scheme significantly reduces flooding of the network with unnecessary control packets. The effect in turn leads to the minimization of unnecessary energy waste in a WSN.

Keywords: DSN · WSN · Sleep scheduling · Overemission · RTS packets · Control packets

1 Introduction

Recent advances in WSN have considerable impact in our everyday life. It is being used in the military to serve many purposes like, monitoring friendly forces, equipment and ammunition battlefield surveillance, reconnaissance of opposing forces and terrain, battle damage assessment nuclear, biological and chemical attack detection. It also has vast applications in environmental monitoring, health, manufacturing, agriculture, and so on. The proliferation and diversity of its usage could not have been without the development of Micro-Electro-Mechanical-Systems (MEMs) technologies. MEMs is the technology of very tiny devices. Large-scale deployment of wireless sensor networks is made possible today with the introduction of MEMs technology. Similarly, MEMs has made it possible for sensors that are smaller, cheaper and intelligent to be developed. These sensors are equipped with wireless interfaces to communicate with one another.

G. Wang et al. (Eds.): ICA3PP 2015, Part I, LNCS 9528, pp. 430–444, 2015.
DOI: 10.1007/978-3-319-27119-4_30

Sensor nodes generally have limited processing and computing resources. However, when compared to traditional sensors they are inexpensive. These sensor nodes can sense, measure, and gather information from the environment. Sensor devices are meant to be low power devices equipped with one or more sensors. Battery is the main power source in a wireless sensor node. There may be a need for a secondary power supply like the harvesting of power from the environment. Perhaps the greatest advantage of wireless sensor networks is the fact that it can be deployed in dangerous regions, even places that cannot be reached by humans who may want to explore those areas for some reasons. For this same reason, powering, recharging or replacement of wireless sensor nodes have proven problematic and have culminated into a very big issue. Therefore, to effectively utilize or maximize the network energy and extend its lifetime, researchers have developed lots of energy-efficient protocols like radio duty cycling to prolong the network lifetime [1–6].

Radio duty cycling introduces the problem of finding moment in time where both the sender and the receiver are active so that a link can be established. Traditionally, Medium Access Control (MAC) protocols use either synchronous or asynchronous approaches to handle challenges of radio duty cycling in WSN [2, 7, 8].

The issue of limited energy resources in WSN poses a performance problem in terms of the throughput of a network and also on the overall lifetime attainable by a network. Extending battery life of the wireless node and prolonging the overall network lifetime of a WSN is thus a foremost task in the design of practical WSNs [9–11].

A Network lifetime can be defined as the time till the first sensor node runs out of its battery capacity [3]. Network lifetime maximization involves all levels of sensor network hierarchy, from hardware/software design, to communication protocols. Recent efforts dealing with communication-related energy costs, mainly focus on two separate but equally important fronts which are energy-efficient routing and sleep scheduling [1].

Radio transceiver consumes more energy compared to other parts of a sensor node. The radio transceiver is controlled by a MAC protocol. MAC protocol also coordinates accesses among nodes that are competing for the shared medium and tries to reduce interference among transmitting nodes. There are several major sources of unnecessary energy consumptions in the MAC layer [2]. The first is idle listening. Idle listening occurs when a node leaves its radio transceiver on even though there is no data being transmitted or received at the time. Energy usage during idle listening state can be compared to that of receiving state energy consumption. Overhearing is yet another cause of energy waste in a wireless sensor network. This happens when a node intercepts a packet that is destined for another node. Since the intercepted packets also have to be decoded before the node realizes that it was meant for another node which leads to unnecessary power consumption. The third waste of energy is over-emission which happens when the transmitter node transmits a packet while the receiver node is not ready to receive.

In the proposed approach, we designed a DSN scheme where the designated node holds schedule information of nodes going into sleep mode. Although the DSN is responsible for the storage of the schedules of all the nodes going into sleep mode, it does not detect how the entire network work. Therefore, it is not entirely a centralized system. However, it is responsible for the effective functioning of the network. The DSN node stores the sleep schedule of all the neighboring nodes and must remain awake while all other nodes are sleeping. However, since the DSN does a lot of processing and controlling, it uses up a larger amount of energy in the process. Therefore, after a time threshold and depending on energy level and other criteria, the DSN selects another node to act as the replacement. This ensures balance of energy consumption between nodes in the same network. The active period of a node is not fixed. However, just like in [3], after a certain time threshold without any event activation, nodes randomly set a sleep schedule and share them with the DSN before going to sleep. Figure 1 shows the basic concept of a DSN-based network. This paper focuses on the introduction of an effective sleep scheduling scheme that helps to achieve energy maximization in a WSN through effective sleep scheduling scheme.

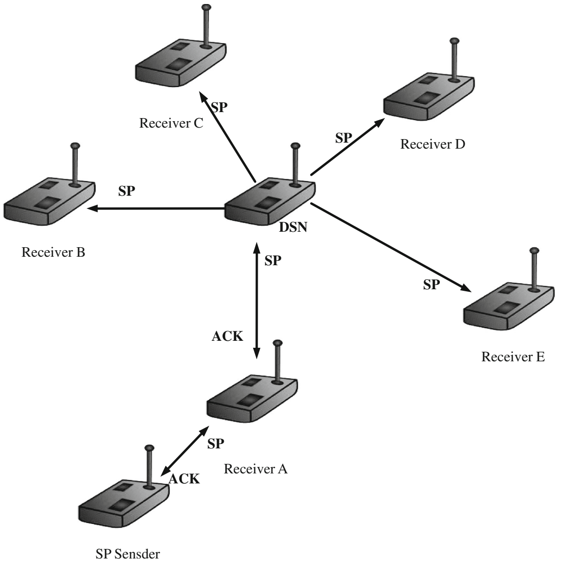

Fig. 1. Designated Sensor Node

The rest of the paper is presented as follows. In Sect. 2, related work is introduced. Section 3 introduces the proposed system model, and in Sect. 4 the scheme is compared to a related work through results and discussions. Finally, Sect. 5, concludes the paper.

2 Related Work

One of the proposed solutions for the reduction of unnecessary energy consumption on IEEE 802.11 is Sensor MAC(S-MAC) [2]. S-MAC protocol uses a periodic listen and sleep schedule scheme to minimize idle listening and ensure energy efficiency. In S-MAC, each node turns its radio transceiver off and goes into a sleep mode for a period of time. The node switches to a listening mode when it wakes up and then communicate with other nodes. The listen period is used for the exchange of control packets, such as RTS, CTS, SYNC, ACK packets, as well as data packets. In S-MAC, neighboring nodes form a virtual cluster and set up a common synchronized listen-sleep schedule. However, this method has some drawbacks as the uniform duty cycle of S-MAC can have a high end-to-end latency. The traffic load across the entire network is huge and S-MAC, having a fixed active time cannot adapt to this much traffic. Many other protocols have been designed to address the inefficiencies in S-MAC. Such protocols are DS-MAC and Time-out MAC (TMAC) [4, 7]. All these schemes have been proposed to enhance the performance of the S-MAC protocol. After a time threshold without any event activation in T-MAC, the listening period ends immediately and the node returns to sleep mode. This was not the case in S-MAC. On the other hand, DS-MAC aims at improving the latency performance of S-MAC by adjusting the duty cycle of the sensor node depending on the traffic condition. A similar work tries to find a way to maximize the lifetime of a WSN by a joint routing and sleep scheduling scheme [1]. Although their work is a theoretically based scheme, this paper follows a similar path with modifications to their proposed sleep scheduling mechanism. However, our work is restricted to only the technical aspect of sleep-wake scheduling mechanism. This work also focuses mainly on the sleep scheduling with a minimal interest in the routing aspect.

Joint Routing and Sleep Scheduling endeavors to jointly optimize energy consumption in a WSN. One of the advantages of this scheme is that it combines both sleep scheduling and routing to maximize the lifetime of a WSN. The combination makes it a more comprehensive approach to solving the issue of energy maximization. However, sleep scheduling mechanism in PS-MAC like most other mechanisms mentioned earlier in this paper fails to effectively tackle the problem of overemission. As shown in the paper, the authors assume a node which initializes its RF circuits as soon as it wakes up from sleep mode. This is shown in Fig. 2. The nodes take time T_{rf} and E_{rf} amount of energy to initialize its radio frequency. When a node has a packet to send after it has successfully initialized its RF, it first listens to the channel for a period of T^{i}_{lis} to see if any of its neighbors are transmitting.

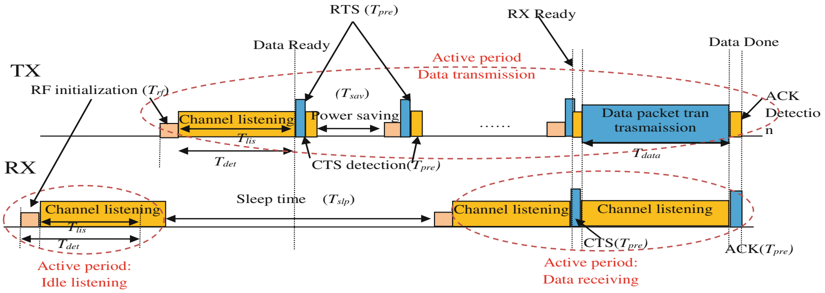

Fig. 2. Timing diagram in different active periods

The paper defines T^{i}_{det} as the total time from when a node wakes up to when it goes back to sleep. This is calculated as $T^{i}_{det} = T_{rf} + T_{lis}$. In the paper, if the channel is idle for T^{i}_{det} time, then the node can send an RTS preamble. If the target receiver is also in an active mode, and gets the RTS preamble, it replies with a clear to send (CTS) packet.

However, if the receiver happens to be in a sleep mode, the transmitter will resend the RTS preamble before going into a power saving mode for a short period of time T_{sav}. The process is repeated over and over again until the receiver node wakes up and captures the RTS preamble. Subsequently, to ensure that all transmitter-receiver pairs are sufficiently connected and that communication between other links within the same RF range is correctly detected, then T^{i}_{det} has to be long enough to cover at least two consecutive RTS preambles including the T_{sav} in-between them. The condition also has to be true for all sensor nodes within its RF range. Contrary to our proposed scheme, transmitters in [1] have no way of knowing the status of the receivers before sending an RTS. Therefore, if a sender has a packet to send to a receiver, it keeps sending an RTS over and over again until one is captured. Here, in the best case scenario, the T^{i}_{det}, must be long enough to cover at least two consecutive RTS preambles including the T_{sav} between them. One can therefore assume that in some cases, RTS could be sent more than two times before it can be captured by the intended receiver. This could happen when the intended receiver (j) goes into sleep mode just immediately before the first RTS arrives. In this case, the transmitter have longer time to wait and therefore more RTS preambles to send before the receiver wakes up. In some situations, extension of sleep duration of nodes may be necessary depending on some conditions like the characteristics of the event to be sensed. Hence, increasing the time a node requires to sleep T^{i}_{slp}, results in a proportional increase in the number of RTS preambles a transmitter has to send before the receiver wakes up and captures a packet.

Probability sensor MAC (PS-MAC) [3], uses a probabilistic method to guess the time a node wakes up. The advantage of PS-MAC is that it can closely guess the expected time to eradicate over-emission in a WSN. In PS-MAC, each sensor node makes use of a pseudo-random number generator to generate a pre-wakeup probability and seed number. When this is done, the node exchanges its pre-wakeup probability and seed numbers with its neighboring nodes to enable each node to know its neighbor's sleep schedule. However, This scheme fails to take into consideration one thing- there is a possibility that some or all neighboring nodes may either be busy or temporarily out of reach. In this case, there can be two possible outcome scenarios- (1) The node may go into sleep mode as soon as the generated sleep time elapses without being able to share its schedule with the neighboring nodes. (2) The node may have to keep cancelling and re-generating fresh sleep and pre-wakeup schedule until it can be able to share them with its neighbors. In the case of the first scenario, the whole system may be thrown into chaos because when nodes that did not get the sleep schedule may eventually want to communicate with the node that is already in sleep mode. At this point they do not have a way of determining the status of the receiver node. Therefore, these nodes have no option but to resort to sending RTS packets until one becomes acknowledged. This can lead back to over-emission energy waste in PS-MAC protocol. Similarly, in the second scenario where a node keeps re-generating fresh sleep schedule when its previously generated (but not shared) schedule time elapses. Since there is no guarantee if and when the node can be able to share its pre-wakeup schedule, the node can go on regenerating fresh schedule until it is able to share it with all its neighbors. This also can go on indefinitely and therefore, it can have serious effects on the whole scheme.

3 Proposed Scheme

Although the target of this work is mainly to reduce to the barest minimum the number of control packets, we still retain the idea of RTS preambles to avoid collision. However, this scheme reduces the number of RTS and other control packet like sync packets being sent by ensuring that these packets are only sent when the intended receiver is awake to receive them. In S-MAC, and other previous schemes, sync packets are sent whenever fresh sleep schedule is generated. However, in this proposed scheme, there is no need for the synchronization phase since it is always possible to obtain the schedule from the DSN node any time if needed.

For simplicity, we say that a node can either be in a sleep mode or in an active mode. The radio transceiver is switched off when in sleep mode. The sleep duration is fixed for all nodes in the network, so for any node i, the sleep duration is denoted by $\mathrm{D}^{\mathrm{i}}_{\mathrm{slp}}$. The reason all the nodes have a fixed sleep period is so that the DSN can calculate and know exactly when a node is supposed to be awake. We refer to this as Expected Time of Awakening (ETA). This can be calculated using Eq. 1.

$$ETA_i = D_{slp} - T^i_{slpt} \tag{1}$$

Where, $\mathrm{D}_{\mathrm{slp}}$ is the sleep duration of any node and $\mathrm{T}^{\mathrm{i}}_{\mathrm{slpt}}$ is how long any given node *i* has already been asleep.

A node initializes its radio frequency (RF) when it wakes up. When in active mode, a node can either be transmitting or receiving data using time slots. Thus, an active period is said to be a data transmission slot if the node transmits a packet in this time slot. Similarly, an active period is a data reception slot if the node receives a packet in this time slot.

After a certain threshold without event activation, a node randomly chooses a time to go to sleep. However, when a node randomly generates a sleep schedule, other nodes have no way of knowing about it, therefore the node has to send its new sleep schedule to the DSN and waits for and acknowledgement (ACK) from the DSN as illustrated in Fig. 3. The node only goes into sleep mode when it receives an ACK from the DSN.

At any given time, a node can get the schedule information and the status of another node from the DSN by sending a request packet (RP). The request packet specifies the ID of the node whose schedule is being requested as well as the ID of the requesting node. This helps the DSN to determine which schedule is going to which node. When the DSN node receives the request packet, it forwards the most recent schedules of all the nodes in sleep mode to the requesting node.

Naturally, nodes go in and out of the networks due to so many reasons. Some of these reasons are explained in [12]. Nodes transition from fully functional to partially functional and eventually to non-functional states as illustrated in Fig. 4. Nodes that are fully functional are nodes that have energy level above the energy threshold θ. Partially functional (PF) nodes have energy levels lower than θ while non-functional nodes are no longer part of the network hence their edges are deleted. PF nodes are still part of the network however, they may not be able to receive or transmit packets with their current

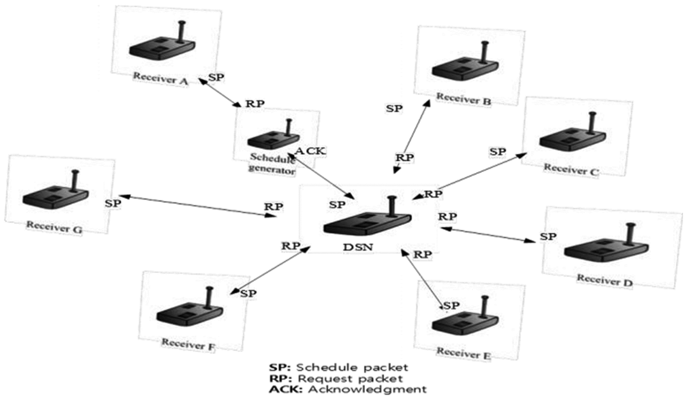

Fig. 3. Schedule generator generates SPs sends it to the DNS and receives and ACK, while other nodes send Request Packet (RP) whenever they need to know the status of a certain node.

energy level. In PS-MAC, PF neighboring nodes will continuously and unsuccessfully try to exchange schedule packets with them thereby causing over emission. However, this problem does not occur in the DSN network because if a node gets disconnected in the network due to low energy, the DNS purges its schedule from the list. Hence, when a node requests a none-functional node's sleep schedule, the DSN replies with a void packet which indicates that the requested node has been disconnected from the network.

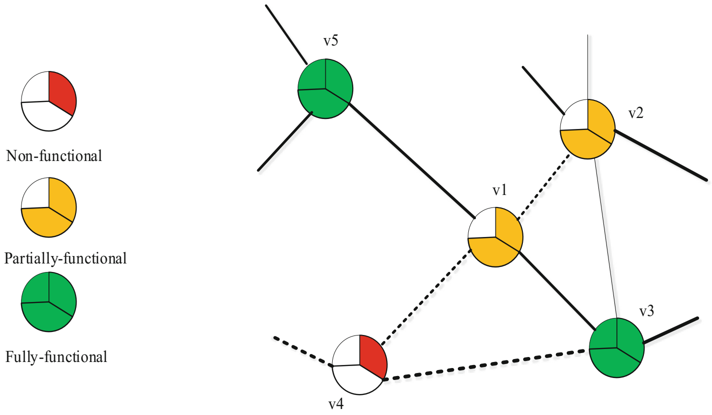

Fig. 4. (Color online) Fractional percolation on a network: the green nodes v3 and v5 remain intact (FF); the amber nodes v1 and v2 are under some nonfatal error (PF); the red node v4 suffers from some fatal error (NF). Dotted lines mean deleted edges [12].

DSN scheme addresses the shortcoming in PS-MAC. In the DSN scheme, each node has to receive an ACK from the DSN before they can go into sleep mode. After receiving the schedule, the DSN saves the schedule for future referencing and also shares this sleep schedule with all the nodes that are awake and ready to receive a packet. Similarly, since the DSN has the expected time of awakening of all the sleeping nodes, whenever a node rejoins the network after its sleep cycle is over, the DSN shares the sleep schedules of all neighboring nodes with the node that is re-entering active mode. When any node, for instance node B in Fig. 3 wants to communicate with a sleeping node C, then B can calculate the time node C will wake up and will add a randomly generated number (γ). I.e. ETA + γ. The random number is used to avoid collision at C when there are more than just one node waiting to communicate with it as soon as it re-enters active mode.

In DSN network, we use the parameters in Table 1. The assumption is that all packets including preambles are received at a constant power of p_{rx}, while it takes a transmitter a power of p_{tx} to transmit any packet. The duration of a data packet is denoted by T_{data} while assuming that all packets (RTS, CTS, ACK…), are of a fixed length T_{pre}. Similarly, energy for a potential transmitter to calculate the expected time of a potential receiver is denoted by p_{eta}. When a node wakes up from sleep, it takes that node a power of E_{rf} and time T_{rf} to initialize its RF circuits. There are two possible times that a node can receive a sleep schedule of its neighbors. (1) When it is fully active and (2) When it has just come out of sleep mode. In the first case scenario, RF initialization is not considered when calculating the amount of energy it takes to receive the schedule packet while, in the second case scenario, the RF initialization is taken into consideration when calculating the average amount of energy it takes a node to receive a schedule packet. Also, in DSN network, the amount of energy to receive a sleep schedule and transmit a data packet is separately dependent on three different characteristics of transmitters- (a) the DSN node, (b) a waking node and (c) an active node.

Table 1. SYSTEM PARAMETERS.

Peta	Power to calculate expected time of awakening	28 (mW)
Ptx	Transmission power	50 (mW)
Prx	Receiving power	30 (mW)
E*rf*	Energy to initialize RF circuits	15.05 (μJ)
Trf	Time to initialize RF circuits	2.1 (ms)
Tdata	Time to transmit one data packet	16 (ms)
Tpre	Time to transmit one RTS/CTS/ACK preamble	0.832 (ms)

3.1 The DSN

In the DSN scheme, the designated node deals with a lot of communications. Therefore, it uses up more energy than other nodes in the network. Also a DSN node cannot go into sleep mode while all the nodes in the network are sleeping. For this reason, the energy of the DSN node may dry up faster than all other nodes in the network. To avoid this problem and ensure that the energy consumption is balanced between all the nodes in the network, each node in the wireless sensor network (WSN) take turns to function as the DSN. However, before going into sleep mode, the current DSN has to share all the information it has with the new DSN. Unfortunately, when a new DSN is chosen, other nodes have no way of knowing the identity of this new DSN. Therefore, the new DSN has to broadcast its address to nodes in the network. As the DSN, the new designated node knows the ETA of all the nodes so it shares its ID together with the status of all other nodes with any node re-entering into active mode. This ensures that all nodes have their neighboring nodes' schedule and also know who the current DSN is at all times. When a new DSN has been successfully selected, then the old DSN (now an ordinary node in the network) can immediately generate a sleep schedule, share it with its neighbors as well as with the new DSN node before going into sleep mode. Unlike other nodes that have to wait for a certain threshold time before generating a sleep schedule, a node that has just finished acting as a DSN can immediately generate a sleep schedule. The next DSN node is chosen when the current DSN has used up a certain amount of energy from its initial energy (say $\frac{1}{N}$ of its initial energy). The initial energy here is the amount of energy it has prior to starting as the DSN. The current energy level (CEL) is given by.

$$\mathrm{CEL} = \frac{\text{Initial Energy level}}{\mathrm{N}} \tag{2}$$

The criteria for choosing the next DSN are- 1) the node must be free at the time the new DSN is being chosen. 2) Nodes being considered by the current DSN must possess and energy level greater than the threshold and the node with the highest energy level than others is chosen. When more than one nodes meet the above conditions, then random selection is used to select the next DSN. Figure 5 illustrates the process of determining the new DSN. The current DSN initiates the process of choosing the next DSN by broadcasting a status request packet to the nodes in the network. Each node then replies with their current energy status, current transmission status, and other relevant data. When the DSN receives these information, it then initiates the DSN choosing process. In the first stage, the process first determines nodes that are currently busy (some nodes may have just finished sending out CTS packets and are waiting for data transmission to commence) and these nodes are dropped. The next stage checks for the energy level of the nodes. Nodes that fall short of the minimum energy level threshold are dropped. The minimum energy level threshold is the energy level of the current DSN which is made available to the process from the start. At this stage, it is possible to have only one node with the required energy level threshold or a node that has greater amount of energy than other nodes. When this is the case, that node is automatically selected as the DSN. However, it is also possible to have more than one nodes that meet the above requirements and possess similar energy level. If there are more than 1 nodes at this stage, then the process uses random selection to choose the next DSN.

In the DSN scheme, only a DSN node has to transmit an ACK packet after receiving a sleep schedule, therefore the energy consumed by a DSN node to receive and transmit and ACK packet ($\mathrm{R_{DSN}}$), is calculated as follows.

$$\boldsymbol{R_{DSN}} = \mathbf{p_{rx}}\left(\mathbf{T_{data}} + \mathbf{T_{pre}}\right) + \mathbf{p_{tx}}\left(\mathbf{T_{data}} + \mathbf{T_{pre}}\right) \tag{3}$$

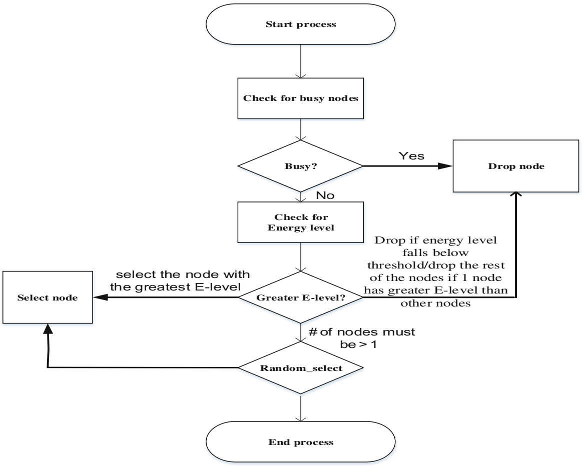
Start process
Check for busy nodes
Busy?
Yes
Drop node
No
Check for Energy level
Drop if energy level falls below threshold/drop the rest of the nodes if 1 node has greater E-level than other nodes
select the node with the greatest E-level
Select node
Greater E-level?
# of nodes must be > 1
Random_select
End process

Fig. 5. Choosing the DSN.

Similarly, the energy required for a DSN node to transmit a data packet to node j in DSN is equal to.

$$T^{j}_{DSN} = 2\mathrm{R}_{\mathrm{DSN}} + \mathrm{P}_{\mathrm{eta}} \tag{4}$$

Where P_{eta} is the energy required to calculate the expected time of awakening of that receiver.

This mechanism looks like a centralized system where the absence of the DSN may lead to the total failure of the whole system. However, this is not entirely so. The work of the DSN is mainly to hold schedule information. It does not determine the functionality of the entire system. Moreover, a temporary DSN can assume the role of DSN if the current DSN remains inactive for a set period of time.

3.2 The Waking Node

A Waking node in a DSN network is considered to be the node that is transitioning from a sleep mode (SM) into an active mode (AM) see Fig. 6. A node that has just woken up needs time T_{rf} and energy E_{rf} to initialize its radio frequency (RF), this is denoted by the gray areas in-between SMs and AMs. These two factors are taken into consideration while calculating the energy consumption to receive and or transmit a single data packet. RF initialization is considered only for the waking node. This is because a node first initializes its RF when it wakes up and this is the only time RF initialization is required. Consequently, the additional energy required to do this (I_{rf}) is given by.

$$\mathrm{I}_{\mathrm{rf}} = \mathrm{T}_{\mathrm{rf}} \times \mathrm{E}_{\mathrm{rf}} \tag{5}$$

A waking node requires RF initialization. However, it does not require to send an Acknowledgement packet whenever it receives a schedule packet from the DSN.

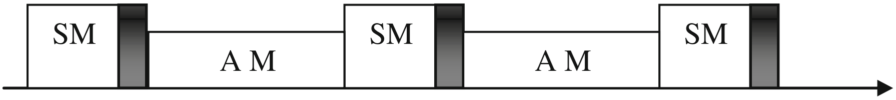

Fig. 6. Transition of waking node.

Therefore, the energy consumption for a Waking node to receive a schedule packet (R_{wn}), is calculated thus.

$$R_{wn} = I_{rf} + p_{rx}\left(T_{data} + T_{pre}\right) \tag{6}$$

Subsequently for a waking node i to transmit a single data packet T_{wn}, to a receiver j, the energy required to initialize its radio frequency, the ETA computational energy, the energy to receive a schedule packet, and energy to transmit an actual data packet is calculated as follows

$$T^{ij}_{wn} = R_{wn} + P_{eta} + p_{tx}\left(T_{data} + T_{pre}\right) \tag{7}$$

3.3 Active Node

An active node is considered to have an already initialized RF. This can be a node that has fully transitioned into an active mode or a node that has just finished communicating with other nodes and is now an active listener. Similar to the waking node in the previous section, this node does not require to transmit an ACK packet whenever it receives a schedule packet, therefore, the energy to receive a sleep schedule packet (R_{an}), is simply

$$R_{an} = p_{rx}\left(T_{data} + T_{pre}\right) \tag{8}$$

Subsequently, to know how much energy it takes a node i to transmit a single data packet to another node j then, the amount of energy to receive a sleep schedule, the energy to compute ETA of a particular receiver and the energy required to transmit an actual data packet to that receiver is calculated. Therefore, for an active node, the energy required to transmit a single data packet T_{an} is.

$$T^{ij}_{an} = R_{an} + P_{eta} + p_{tx}\left(T_{data} + T_{pre}\right) \tag{9}$$

4 Results and Discussions

In a DSN network, a transmitter avoids sending RTS preamble and other control packets repeatedly to the intended receiver who may either be temporarily out of reach or even disconnected from the network. Also, synchronization of sleep schedule is avoided since the DSN has all the information. The transmitter requires to know only the time when a node goes into sleep mode to be able to compute the ETA of that node. The transmitter already has or can obtain this information from the DSN node at any given time. Once the transmitter has this information, it can calculate when the receiver wakes up. Hence it only has to send one RTS to the receiver after it has re-entered active mode. Apparently, no RTS is sent while the receiving node is in sleep mode. In the proposed scheme, a node makes sure that the DSN receives its sleep schedule. The DSN keeps record of the sleep timetable of all the nodes in the network. DSN node hands over all its information to another node to act as the DSN when it has used up about $\frac{1}{N}$ of the initial energy as at the time it started acting as the DSN. This regulates the amount of time a node can act as the DSN. It is also an effective way to ensure balance of energy consumption among all nodes in a DSN-base network.

It is mentioned earlier that sleep duration of sensor nodes in the network may require to be extended depending on the situation. For example, sensor nodes that are deployed in an area were events to be studied happen periodically may opt for longer sleep duration to save the energy consumption that result from frequent turning on and off of the radio transmitter. In this work, we compared the energy consumed by a node in DSN with that in PS-MAC. In the proposed DSN scheme, the energy consumption based on the type of node is separately calculated owing to the fact that there can be three distinct nodes in the DSN network. The result in Fig. 7 below clearly shows that as the sleep duration of the sensors in the network increased, so did the average amount of RTS and control packets sent. This is attributed to the fact that some transmitters in PS-MAC scheme who want to communicate with nodes that just entered sleep mode potentially send more control packets before the intended receivers become active again. Similarly, when some neighboring nodes are busy or temporarily out of reach, some nodes in the PS-MAC either repeatedly try to share their sleep schedules with them, thereby causing them to regenerate new schedules when the time elapses.

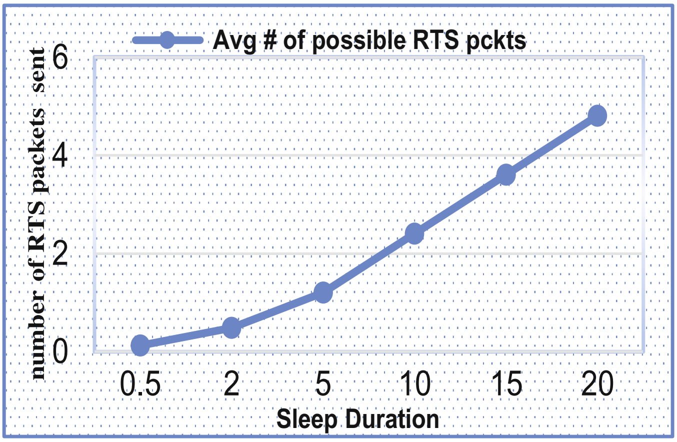

Fig. 7. Possible number of RTS packets sent with and interval of 4.16 ms.

In terms of energy consumption, Fig. 8 shows the average amount of energy consumed by each type of sensor node in a DSN network and PS-MAC respectively to initialize RF, receive and transmit data and control packets with respect to sleep duration. This proves that deploying sensor nodes equipped with PS_MAC protocol in a field where sensors require to go into sleep mode for a longer period of time is clearly not a good idea. However, DSN on the other proves to be much more efficient in this scenario because the transmitter only has to get the schedule of the intended receiver, calculate its ETA and transmits just once only when the node is awake and ready to communicate.

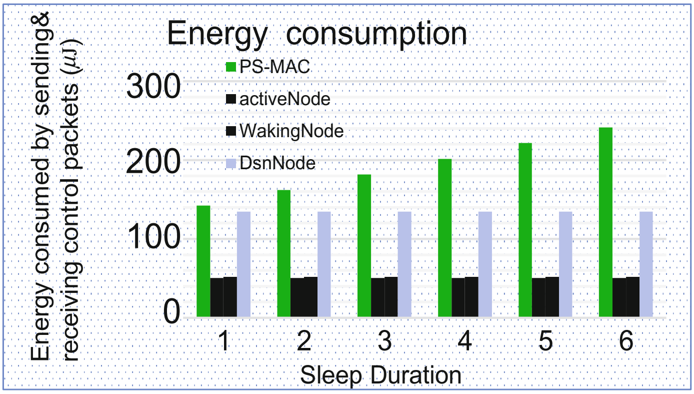

Fig. 8. Duty cycle-based average energy consumption of each node

The scenario in Fig. 7 does not only bring about a significant amount of energy waste, it also considerably impacts on the throughput and quality of service of the entire network. Figure 9 shows that the DSN has an edge over the PS-MAC in terms of the network throughput in an investigation conducted using the NS2 software. The number of sensors are increased from 5 to 35 and the average throughput is calculated which proves that DSN scheme can not only reduce energy waste but also outperform the PS-MAC in terms of the quality of service.

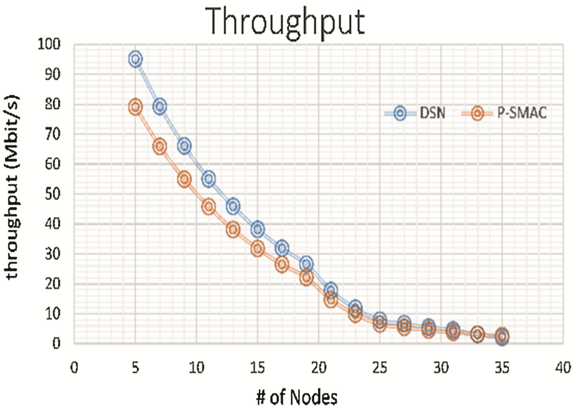

Fig. 9. Average throughput

However in the DSN scheme, as the number of nodes in the network is increased the average throughput becomes more and more similar to that of the PS-MAC. This is because as the number of sensors in the network increased, the number of processing the DSN has to deal with also increased. Since the DSN does a lot of work at this time, this means that it also uses up energy faster and therefore the change of the DSN happens more frequently. This drawback will be studied in our future work to sustain the throughput when the number of sensor increases.

5 Conclusion

A DSN scheme that enhances energy consumption of wireless sensor nodes through sleep scheduling for nodes in the network was proposed. This scheme adopts Designated Sensor Nodes that work not as a coordinator, but as a time schedule keeper in a WSN. It assures that all nodes know the status of all their neighboring nodes at all times. It also allows for them to calculate the ETA of any particular node. By doing so, possible flooding of the network with RTS preambles and other control packets while the intended receiver is in sleep mode is effectively avoided. DSN scheme also introduces an effective way to ensure that energy consumption of all nodes in the network is balanced by means of DSN role shift. It provides an effective way to rotate the job of a DSN among all the sensor nodes in the network by considering their energy levels. This effect ensures the extension of the network's lifetime.

Acknowledgments. This research was financially supported by the Ministry of Education (MOE) and National Research Foundation of Korea (NRF) through the Human Resource Training Project for Regional Innovation (no. 2014H1C1A1067126), the BK21 Plus project (SW Human Resource Development Program for Supporting Smart Life) funded by the Ministry of Education, School of Computer Science and Engineering, Kyungpook National University, Korea (21A20131600005), and the This work was supported by the IT R&D program of MSIP/KEIT. [10041145, Self-Organized Software platform (SoSp) for Welfare Devices].

References

1. Liu, F., Tsui, C.-Y., Zhang, Y.: Joint Routing and Sleep Scheduling for Lifetime Maximization of Wireless Sensor Networks. IEEE Trans. Wirel. Commun. **9**(7), 2258–2267 (2010)
2. Ye, W., Heidemann, J., Estrin, D.: Medium access control with coordinated adaptive sleeping for wireless sensor networks. IEEE/ACM Trans. Networking **12**(3), 493–506 (2004)
3. Choi, S.-C., Lee, J.-W., Kim Y., Chong, H.: An energy-efficient mac protocol with random listen-sleep schedule for wireless sensor networks. In: TENCON 2007 - 2007 IEEE Region 10 Conference, Taipei (2007)
4. Dam, V.T., Langendoen, K.: An adaptive energy-efficient mac protocol for wireless sensor networks. In: Proceedings of the 1st International Conference on Embedded Networked SenSys, California (2003)

5. Niu, J., Cheng, L., Gu, Y., Jun, J., Zhang, Q.: Minimum-delay and energy-efficient flooding tree in asynchronous low-duty-cycle wireless sensor networks. In: Wireless Communications and Networking Conference (WCNC), 2013 IEEE, Shanghai (2013)
6. Rasouli, H., Kavian, Y., Rashvand, H.: ADCA: adaptive duty cycle algorithm for energy efficient IEEE 802.15.4 beacon-enabled wireless sensor networks. Sens. J. IEEE **14**(11), 3893–3902 (2014)
7. Yuan, X., Bagga, S., Shen, J., Balakrishnan, M., Benhaddou, D.: DS-MAC: differential service medium access control design for wireless medical information systems. In: Engineering in Medicine and Biology Society, 2008, EMBS 2008, 30th Annual International Conference of the IEEE, Vancouver, BC (2008)
8. Cunha, F., Mini, R., Loureiro, A.: Sensor-MAC with dynamic duty cycle in wireless sensor networks. In Global Communications Conference (GLOBECOM), 2012 IEEE, Anaheim, CA (2012)
9. Raghunathan, V., Ganeriwal, S., Srivastava, M.: Emerging techniques for long lived wireless sensor networks. IEEE Commun. Mag. **44**(4), 108–114 (2006)
10. Li, J., AlRegib, G.: Network lifetime maximization for estimation in multihop wireless sensor networks. IEEE Trans. Sig. Process. **57**(7), 2456–2466 (2009)
11. Dagher, J., Marcellin, M., Neifeld, M.: A theory for maximizing the lifetime of sensor networks. IEEE Trans. Commun. **55**(2), 323–332 (2007)
12. Shang, Y.: Vulnerability of networks: fractional percolation on random graphs. Am. Phys. Soc. **89**(1), 012813-1–012813-4 (2014)