A Multi-Channel Frame-Slot Assignment Algorithm for Real-Time MACs in Wireless Sensor Networks*

Van Vinh Phan and Hoon Oh**

Ubicom Lab, School of Computer Engineering and Information Technology,
University of Ulsan, Ulsan, Korea
pvvinhbk@gmail.com, hoonoh@ulsan.ac.kr

Abstract. In this paper, we propose a multi-channel frame-slot assignment algorithm to reduce the total number of slots required for data packet transmissions and enable a slot reuse against the irregular interference caused by the gradual signal fading. This becomes possible thanks to the efficient scheduling scheme of multiple channels. In our approach, each node determines a sending channel and a receiving channel based on the channel selection rule in a totally distributed manner. The channels are scheduled to the nodes at different depths such that any vertically two-hop away nodes can use the same slot without causing any inference. We evaluate the performance of the proposed algorithm by resorting to simulation. The simulation results show that our proposed approach significantly improves network throughput, packet latency and superframe size.

Keywords: Sensor node, scheduling, data channel, frames/slots, superframe.

1 Introduction

Wireless Sensor Networks (WSNs) can be used for building a Safety Monitoring and Control System (SMOCS) that monitors the safety of the workers who have to work long in the closed and dangerous working environments and warns them of any hazardous situation. The SMOCS server stores, manages, and analyses data or context information that were sent periodically by every node, and judges whether the target field is safe or not based on the collected context information. If a dangerous situation is perceived, the server sends a request to the workers so that they can take some measures against the danger.

In this paper, we propose a new frame-slot scheduling algorithm for a real-time MAC protocol in WSNs. The existing MAC protocols in WSNs can be classified into the contention-based ones and the Time Division Multiple Access (TDMA) based ones. A contention-based MAC protocol, based on the Carrier Sense Multiple Access

* This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (2013R1A1A2013396).
** Corresponding author.

R.C.-H. Hsu and W. Shangguang (Eds.): IOV 2014, LNCS 8662, pp. 170–179, 2014.

(CSMA) scheme, is widely employed in wireless networks due to its simplicity, flexibility, and robustness. Because of energy constraint, most of the early contention-based MAC protocols focused on achieving a low-duty cycle. For example, S-MAC [1] with an active-sleep cycle puts nodes to sleep periodically in idle listening period to conserve energy. However, a static active-sleep cycle of S-MAC can cause packet delay and low throughput in case of variable traffic loads. T-MAC [2] can mitigate the drawbacks of S-MAC by using an adaptive active-sleep cycle. B-MAC [3] employs Clear Channel Assessment (CCA) to enhance the utilization of channel and uses Low Power Listening (LPL) scheme to minimize the energy consumption. However, because of using the long preamble mechanism, the latency is gradually accumulated when packet travels through multi-hop path and energy is wasted at both sender and receiver after the receiver has woken up. In general, most of early MAC protocols for low duty applications try to improve energy efficiency, but increase latency as well since they use an active-sleep cycle. In addition, they usually suffer from low transmission reliability because of interference or collision problem. This phenomenon will become worse in the network with higher traffic. On the other hand, the TDMA technique can mitigate the interference and collision problem, and reduce packet latency. Several TDMA-based MAC protocols have been proposed for wireless ad hoc networks [4-6] and wireless sensor networks [7]. TreeMAC [7] suggests a frame-slot pair slot assignment (FSA) algorithm in which every node is allocated a frame-slot pair such that a node is assigned a number of frames in proportion to the number of its descendants. FSA allows a slot reuse for any two nodes that are three-hop away vertically. However, this suffers from the lack of data packet aggregation and filtering and irregular interference because of the gradual fading of a radio signal. It also suffers from the waste of slots due to the characteristic of the slot scheduling. To resolve these problems, DSA [6] does not allow a slot reuse and schedule the slots to each node in the form of a sequence of receiving slots and then a sequence of sending slots. However, DSA tends to increase a superframe size as the network size increases due to the inherent refusal of the slot reuse. Generally, the MAC protocols proposed so far mostly operate on a single channel. However, the above problems can be well addressed if multiple channels are used. Some multi-channel approaches have been proposed recently. TMCP [8] partitions the network into multiple sub-trees, each subtree is assigned one unique channel and uses the CSMA technique within the same channel. This is simply to reduce the contention scope of each node. In TFMAC [9], each node is assigned transmission slots during a control contention slot, each transmission slot is assigned a different channel that is used for data transmission. However, it may not be easy that two-hop away nodes make conflict-free channel selection.

In this paper, we propose a new multi-channel frame-slot assignment algorithm to reduce the total number of slots for data transmissions and enable a slot reuse against the irregular interference by the gradual signal fading. In our approach, each node is assigned a number of frames (each frame has two slots) that corresponds to the number of its descendants. Then, each node determines a sending channel and a receiving channel based on its depth according to the channel selection rule which works in a totally distributed manner. The channel selection rule guarantees that any vertically two-hop neighbors do not interfere with each other in either sending or receiving. Our experimental results indicate that the proposed approach improves network performance significantly.

In what follows, the network model is described in Section 2. We present the formal description of the proposed approach in Section 3. Section 4 covers performance evaluation. Finally, we make concluding remarks in Section 5.

2 Network Model

In this study, we consider a wireless sensor network that consists of one server (sink) and several sensor devices integrated with various sensor modules such as a thermal sensor module, a gas sensor module, an oxygen sensor module, a smoke sensor module, etc., which can be used for building a Safety Monitoring and Control System (SMOCS) to monitor the safety of the workers who have to work long in the closed and dangerous working environments. The SMOCS server is wall-powered, while sensor nodes are battery-powered. A sensor node senses data from environment periodically and sends it toward the server in multi-hop fashion. The server collects data from sensor nodes in the target environment, and analyses these data to judge whether a dangerous situation has occurred or not. Sometimes, a link between nodes can be broken because of node failure, battery depletion, or the intervention of some communication obstacles.

3 Multi-Channel Frame-Slot Scheduling

3.1 Reliable Tree Construction

At the initialization, all nodes co-operate to build a tree which is rooted at the sink. A link (x, y) is said to be *reliable* if node x can transmit packet to node y successfully when there is no interference. Then, we can construct a robust tree such that every tree-link is bi-directionally reliable (*B-reliable*) as follows [10].

Tree construction is initiated by an *advertisement* (*ADV*) message issued by the sink which is the only tree member at the initialization stage. Upon receiving the *ADV* message, an orphan that has a *reliable link* joins the sink by sending a *join request (JREQ)* message. A sending node includes a set of its neighbors with a reliable link in the *ADV* or *JREQ* message so that the receiver can determine whether it has a B-reliable link to the sender or not. Upon receiving *JREQ*, the member sends a *join response* (*JRES*) message and takes the orphan as its child if the corresponding link between them is *B-reliable*. When the orphan receives *JRES*, it takes the member as its parent. Another orphan who has overheard *JREQ* can take the same procedure to become a member if its link is *B-reliable*. If an orphan overhears *JREQs* from multiple members with *B-reliable* links, it pairs with a member that has the shortest distance to the sink. During the operation time, if a certain node detects the failure of the link to its parent, it tries to find one neighbor that can provide the *B-reliable* link and shortest distance to the sink and then joins that node by sending *JREQ*.

3.2 Frame-Slot Assignment

After the tree construction period, the frame-slot assignment process that consists of a frame demand calculation (FDC) function and a frame start-time assignment (FSA) function is performed. Each data frame consists of two slots, *sending slot* and *receiving slot*. A node uses the sending slot to send one data packet to its parent while it uses the receiving slot to receive one data packet from one of its children. Then an intermediate node needs a number of data frames to receive and transmit all the packets generated by its descendants plus some data frames to transmit data packets generated by its own node. The total number of data frames that a node needs to process (i.e., receive and send) data packets is referred to as a *frame demand*.

In the FDC function, starting with leaf nodes, every node calculates its frame demand, and then sends the frame demand to its parent. If an intermediate node receives all frame demands from its children, it calculates its frame demand and then forwards it to its parent. In this way, a sink can know total frame demand for the whole network. Thus, each node i sends a *frame demand calculation message, FDCM = (nFs(i), sysTime()),* to its parent, where *sysTime()* is the current time of system and *nFs(i)* as a frame demand of node i is given as follows:

$$nFs(i) = \sum_{x \in C(i)} nFs(x) + \eta_i = \sum_{x \in T(i)} \eta_x \quad (1)$$

where $C(i)$ is a set of children of node i and $T(i) = D(i) \cup \{i\}$, $D(i)$ is the set of node i's descendants, and η_x is the number of packets that node x has to send for one round of data transmission.

Fig. 1 shows an example of the frame demand calculation when $\eta_i = 1$ for every node i, except for a sink.

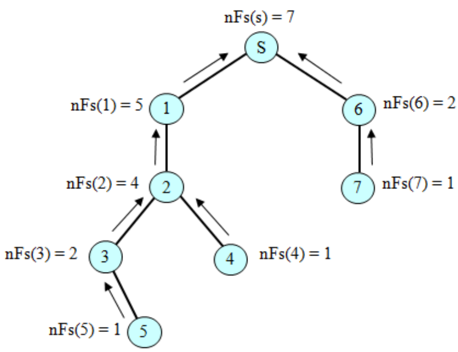

Fig. 1. An example of the frame demand calculation

As soon as the FDC function is finished, starting with a sink, the FSA function tries to assign the start frame number to each node based on the start time of the whole data frame demand (i.e., nFs(s)). For convenience, suppose that *startFrame*(i) indicates the start frame position of node i and also node i has k children represented as (c_{i1}, c_{i2}, …, c_{ik}), $k \geq 0$. Each node i assigns data frames to its children by sending the *frame start-time assignment* message, *FSAM* = (*schedFrame(i),sysTime*()) where *schedFrame(i)* = (*startFrame(*c_{i1}*),startFrame(*c_{i2}*),* ..., *startFrame(*c_{ik}*)*), and *startFrame(*c_{ij}*)*, $1 \leq j \leq k$, indicates the start frame position of the j^{th} child of node i. Then, *startFrame(*c_{ij}*)* is calculated as follows:

$$startFrame(c_{ij}) = startFrame(i) + \sum_{x=1}^{j-1} nFs(c_{ix}), j \leq k \tag{2}$$

According to Eq. 2, a node *i* appoints the start frame to each child c_{ij} since it knows the frame demand of each of its children. Then, the remaining data frames are used for sending its own data packets.

Now, it is necessary to perform a *slot scheduling* that determines the usage of the slots in each frame as either sending slot or receiving slot. Basically, each node either sends data packet to its parent after receiving it from its child, or only sends data packet to its parent if it is a leaf. Thus, for a node at depth *i*, receiving and sending have to alternate for its frames, and sending or receiving for its child at depth *i+1* has to alternate in each allocated frame.

Furthermore, to make parallelism (slot reuse) in data transmission possible, for each slot in the same frame, the same slot scheduling has to be repeated every other depth. For example, suppose that two slots of any frame are numbered as 0 and 1. If a node at depth 1 is scheduled as (Rx, Tx), a node at depth 2 should be scheduled as (Tx, Rx) where Rx indicates the receiving slot and Tx does the sending slot. For next pairs of depths, 3 and 4, 5 and 6, and so on, the same slot scheduling is repeated, so that these transmissions can be performed simultaneously. Then, the sending slot (S_d^{TX}) and the receiving slot (S_d^{RX}) of a frame of depth *d* can be determined as follows:

$$\begin{cases} S_d^{TX} = d \bmod 2 \\ S_d^{RX} = (d+1) \bmod 2 \end{cases} \in \{0, 1\} \tag{3}$$

where *d* is the depth of a node in a tree. Especially, for a sink (with depth 0), $S_0^{TX} = 0$. The sending slot of the sink is slot 0. However, since the sink does not have to send, the 0th slot of a sink becomes *sleeping slot.*

3.3 Channel Scheduling

Even though the slots are scheduled as discussed in the previous section, the parallel transmissions may not be made because of channel interference. For example, if a node at depth 2 and a node at depth 4 use the same slot for a data transmission, a node at depth 3 that receives data packets from the node at depth 4 is interfered by over-hearing the data packet that the node at depth 2 sends to the node at depth 1. Thus, we try to avoid channel interference by allocating different channels every pair of depths such that if a node at depth 1 is a receiving slot, the nodes at a pair of depths (1+4k, 2+4k) gets channel *1* and those at depths (3+4k, 4+4k) gets channel *2*, and if a node at depth 1 is a sending slot, the nodes at a pair of depths (0+4k, 1+4k) gets channel *1* and those at depths (2+4k, 3+4k) gets channel *2*. In this way, a slot using the same channel is reused every four depths vertically. Thus, a sending node at depth 1 can be interfered with a sending node at depth 5. Even though signal attenuation makes interference, the possibility of collision is significantly lowered or negligible. This is compared with the irregular interference of the vertical three hops in TreeMAC [7].

For channel scheduling, each node at depth *d* determines the sending channel (Ch_d^{TX}) for a sending slot and the receiving channel (Ch_d^{RX}) for a receiving slot according to the following formulae:

$$\begin{cases} Ch_d^{TX} = \left\lceil \dfrac{((d-1) \bmod 4)+3}{4} \right\rceil \\ Ch_d^{RX} = \left\lceil \dfrac{(d \bmod 4)+3}{4} \right\rceil \end{cases} \in \{1,2\} \tag{4}$$

Accordingly, *our approach needs two data channels only* such that the same channel is reused by the nodes every other depth.

Fig. 2 shows an example of the execution of the frame-slot assignment for the simple tree shown in Fig. 1 where each frame consists of two slots, sending slot or sleeping slot (if there is not data to send) and receiving slot and the small number in each data slot indicates the channel number that the corresponding node uses. A sink node S distributes 7 data frames to nodes 1 and 6 according to their demands. Then *startFrame(1)* = frame #1 and *startFrame(6)* = frame #6. In turn, node 1 distributes 4 data frames to its only child 2 and reserves one data frame for its own data transmission. The curly upward arrow indicates the movement of the data packet generated by node 5.

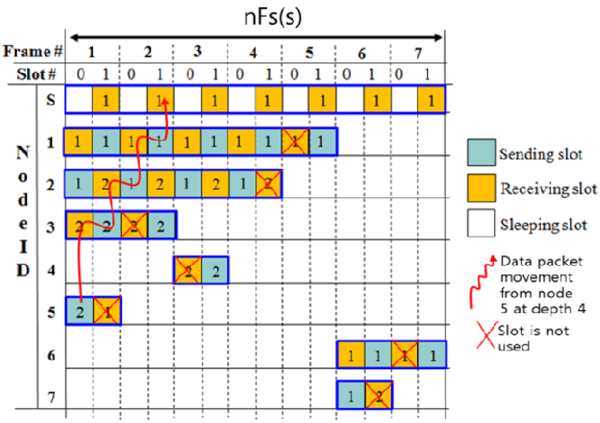

Fig. 2. An example of frame-slot assignment for the scenario in Fig. 1

4 Performance Evaluation

4.1 Simulation Setup

For evaluation of the proposed approach, we used the QualNet 5.0.2 simulator which is commercially used. We compared our proposed approach (abbreviated as MC-FSA) and the DSA approach [6] which showed the good performance over the existing protocols. In the experiments, sensor nodes are static and are randomly distributed in a square terrain 100 x 100 (m^2). Three different scenarios, S1, S2 and S3, which have the sink locations at the center, the top center and the corner of the simulation area, respectively, are used to reflect the variation of tree size in the experiments. Every node generates one data packet of 100 bytes, and then sends the data packet to the sink. The number of sensor nodes (*nNodes*) is varied from 25 to 100. Some key simulation parameters and values in the experiments are shown in Table 1.

Table 1. Simulation parameters and values

Parameter	Value
Default transmission power	–25 dBm (power level 3)
Sensor energy model	MicaZ
Path loss model	2-ray
Noise factor	10 dB
Slot size	6 ms
Dimensions	100 × 100 (m^2)
Simulation time	600 s
Number of nodes	1 sink; 25 sensor nodes
Data packet length	100 bytes

In the simulations, each sensor node uses one transceiver for its operation. Therefore, a node can operate in receiving mode or transmitting mode, but cannot do both at the same time. DSA uses the channel frequency of 2.405 MHz in the IEEE 802.15.4 band while MC-FSA uses two channels: channel 1 with the frequency of 2.405 MHz and channel 2 with the frequency of 2.430 MHz. We compare the proposed MC-FSA approach and the DSA approach by using the following performance metrics:

- Network throughput: The rate of successful data packet delivery measured at the sink per second (*bps* or *kbps*).
- Average packet latency: The average elapsed time that a data packet is delivered from a sensor node to the sink.
- Superframe size: The number of data slots required by all nodes in the network for data packet delivery in one cycle.

4.2 Simulation Results

(a) Network Throughput

One primary goal of the protocol design is to achieve higher network throughput. In this work, we evaluated the network throughput by measuring the number of data packets that are successfully delivered from all nodes to the sink in one second in the three different scenarios (S1, S2 and S3) with the network size of 25 nodes.

As shown in Fig. 3, our proposed approach MC-FSA can achieve the network throughput of 67 kbps, outperforms the DSA approach by more than 30%. Indeed, it can be seen that, the network throughput is affected by the superframe size (SF). The SF of MC-FSA depends on the number of nodes in the network, but not the network topology. However, the SF of DSA increases as the scenario changes (the tree size increases), resulting in a decrease of the network throughput.

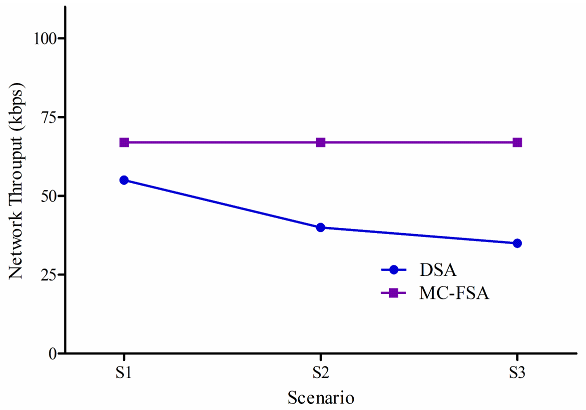

Fig. 3. Network throughput

(b) Average Packet Delay

In order to measure the average packet delay, we measure the average elapsed time that a packet is delivered from the source node to the sink. Fig. 4 compares the two protocols in terms of average packet delay with the network size of 50 nodes and the three different deployment scenarios (S1, S2 and S3).

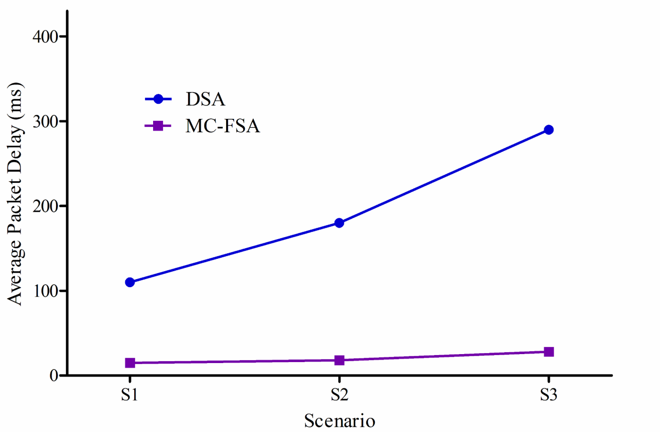

Fig. 4. Average packet delay

The three deployment scenarios have the same number of nodes, but have the different tree sizes (the scenario S3 has the longest tree size with the maximum depth of 9). Fig. 4 also shows that MC-FSA achieves the better performance in terms of the average packet delay while the packet delay of DSA is much more than the other. This is because in DSA, the data slots are allocated to a node in the form of a sequence of receiving slots and then a sequence of sending slots. Hence, before forwarding to the next hop, the packet is buffered at the receiving node until that node receives all packets from its children.

(c) Superframe Size

In the slot scheduling-based algorithms, the superframe (SF) size is directly related to the average packet latency. Thus, in this work, we compare the SF size between the two protocols by using the S1 scenario and changing the number of nodes (*nNodes*) in the network.

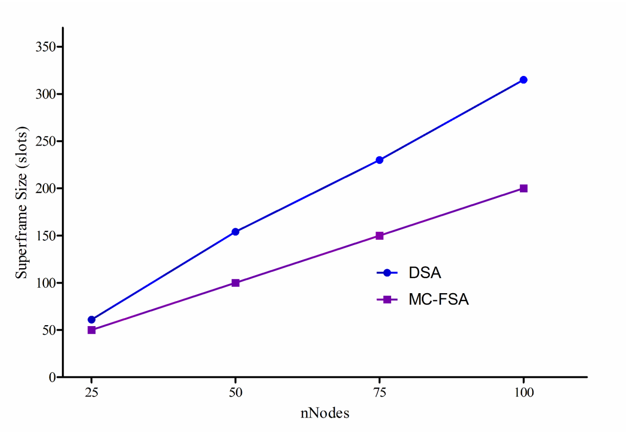

Fig. 5. Superframe size

As shown in Fig. 5, the SF sizes of the two approaches have the increasing curves as *nNodes* increases. It is obvious since the SF size is proportional to the number of sensor nodes. However, Fig. 5 also shows that the SF size of MC-FSA is smaller than that of DSA. The reason is that our proposed approach uses the spatial reuse technique, which allows multiple parallel packet transmissions. On the other hand, DSA does not use the slot reuse technique; therefore the SF size of DSA increases proportionally to the tree size and depth. In general, MC-FSA can reduce the SF size by more than 30% comparing with DSA, resulting in a decrease of the packet latency significantly.

5 Concluding Remarks

In this paper, we proposed a new frame-slot assignment algorithm for real-time applications to deliver data packets timely and reliably. Our proposed approach uses the spatial reuse technique, thus it can achieve higher network throughput, reduce packet latency significantly and satisfy the tight time constraint of real-time applications. The simulation results show that our proposed approach outperforms the other protocol in terms of network throughput, packet latency and superframe size. Therefore, our approach can be potentially used for real-time applications which require the high bandwidth and tight time constraint in the medium-sized to large-sized monitoring and control sensor networks.

References

1. Ye, W., Heidemann, J., Estrin, D.: Medium access control with coordinated adaptive sleeping for wireless sensor networks. IEEE/ACM Trans. Netw. 12, 493–506 (2004)
2. van Dam, T., Langendoen, K.: An adaptive energy-efficient MAC protocol for wireless sensor networks. In: Proceedings of the 1st International Conference on Embedded Networked Sensor Systems, pp. 171–180. ACM, Los Angeles (2003)
3. Polastre, J., Hill, J., Culler, D.: Versatile low power media access for wireless sensor networks. In: Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems, pp. 95–107. ACM, Baltimore (2004)
4. Han, T.-D., Oh, H.: A Topology Management Routing Protocol for Mobile IP Support of Mobile Ad Hoc Networks. In: Ruiz, P.M., Garcia-Luna-Aceves, J.J. (eds.) ADHOC-NOW 2009. LNCS, vol. 5793, pp. 341–346. Springer, Heidelberg (2009)
5. Oh, H.: A tree-based approach for the Internet connectivity of mobile ad hoc networks. Journal of Communications and Networks 5793, 523–534 (2009)
6. Oh, H., Han, T.-D.: A demand-based slot assignment algorithm for energy-aware reliable data transmission in wireless sensor networks. Wirel. Netw. 18, 523–534 (2012)
7. Wen-Zhan, S., Renjie, H., Shirazi, B., LaHusen, R.: TreeMAC: Localized TDMA MAC protocol for real-time high-data-rate sensor networks. In: Proceedings of IEEE International Conference on Pervasive Computing and Communications (PerCom 2009), pp. 1–10 (2009)
8. Yafeng, W., Stankovic, J.A., Tian, H., Shan, L.: Realistic and Efficient Multi-Channel Communications in Wireless Sensor Networks. In: Proceedings of the 27th Conference on Computer Communications (INFOCOM 2008), pp. 1193–1201 (2008)
9. Jovanovic, M.D., Djordjevic, G.L.: TFMAC: Multi-channel MAC Protocol for Wireless Sensor Networks. In: Proceedings of the 8th International Conference on Telecommunications in Modern Satellite, Cable and Broadcasting Services, TELSIKS 2007, Serbia, pp. 23–26 (2007)
10. Oh, H., Van Vinh, P.: Design and Implementation of a MAC Protocol for Timely and Reliable Delivery of Command and Data in Dynamic Wireless Sensor Networks. Sensors 13, 13228–13257 (2013)