

Research Article

A TDMA Scheme for Mobile Sensor Networks

M. Abdullah-Al-Wadud

Department of Industrial and Management Engineering, Hankuk University of Foreign Studies, 81 Wangsan, Mohyeon, Cheoin, Yongin, Gyeonggi 449-791, Republic of Korea

Correspondence should be addressed to M. Abdullah-Al-Wadud; wadud@hufs.ac.kr

Received 2 February 2013; Revised 24 April 2013; Accepted 3 June 2013

Academic Editor: Nazif Cihan Taş

Copyright © 2013 M. Abdullah-Al-Wadud. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We propose a TDMA-based protocol for mobile sensor networks. The proposed protocol overcomes the shortcomings of the other available TDMA-based protocols in a dynamic network where the cluster membership may change frequently. Unlike other existing TDMA-based protocols, we propose to vary the number of timeslots in the TDMA frame to allow underutilization of unused slots and successful allocation of timeslots to the newly joined sensors through minimizing collisions. The approach is fully decentralized and efficient to be used by the cluster head (leader) and the sensor nodes in a cluster. Simulations also show that the proposed mechanism provides significant performance improvements compared to the other existing approaches in terms of different network performance metrics.

1. Introduction

A sensor can be regarded as an atomic computing particle, which can capture and process different data about the surrounding geographical location where it is deployed [1]. After deployment to a target location, the sensors work as a wireless network, through which the data sensed by the sensors can be gathered for analysis.

Mobile sensor networks (MSNs) have gained much commercial as well as research interest in last few years. The critical applications, where MSNs are used, include underwater data collections, rural networks in irregular terrains [2], and other military, scientific, and industrial aspects. Since sensor networks do not need any specific infrastructure, the sensors can be deployed very much rapidly and with ease. However, if the medium access control (MAC) protocol used in MSN cannot provide the required throughput, the deployment of application may be hampered, reducing the utility of the usage of the network [2]. The conventional random access-based MAC using request/response-based contention (such as CSMA/CA) may not fulfill this requirement due to a lot of interference and collisions among the messages sent by different sensors. Moreover, fairness may also not be guaranteed as some nodes may experience a busy channel for an extended period of time.

To minimize the collisions, some time division multiple access (TDMA) [3, 4] protocols have been proposed. In TDMA protocols, one frame (alternatively known as round or cycle) of transmission time is divided into some slots specified for the sensors' transmissions. The frame repeats over time, and a sensor may send message(s) during its dedicated slot only. This ensures collision-free transmissions. The TDMA approaches usually use fixed number of timeslots in frame, which is equal to the number of nodes in the networks (or in the interference range). This works well for static networks. However, in MSNs, some sensors may not have always messages to transmit (many of them may change their statuses to the so-called "sleep" mode to conserve energy). In such cases, the timeslots dedicated to them remain unused, overlooking the opportunity to make use of these slots to improve the performance of the network. Moreover, since the sensors are mobile, the cluster membership may also change. If a sensor node moves out of its current cluster, its assigned slots will remain unused. Again, it should be allocated slot(s) in the frame being used in the new cluster where it moves in. However, use of a fixed number of timeslots in a frame may not handle such situations effectively as there may be shortage of timeslots. Moreover, contention-based approaches [2, 5, 6] are often used during making the decision scheduling the slots among sensors. This also makes it prone to a good

amount of collisions among the control messages sent by different sensors that make delays in joining to a new cluster. Focusing on some slot allocation mechanisms proposed to be used in smart antenna base stations [7, 8], Wong and Jia [6] propose a contention-based protocol to minimize collisions. Here the sizes of the contention windows and the duration (number of slots) of contention period are estimated based on the number of nodes. However, these terms may not be properly estimated if the number of nodes being served is rapidly changed, which is a very common phenomenon in MSNs.

In this paper, addressing all these issues, we propose a flexible TDMA (FTDMA) scheme, which can adapt well with the changes in an MSN. Different performance metrics also show the better performance of the FTDMA compared to the other proposed protocols.

The rest of the paper is organized as follows. Section 2 presents some other works related to ours. Section 3 describes the proposed FTDMA protocol while Section 4 presents the comparative performances of different protocols. Finally, Section 5 concludes the paper.

2. Related Works

There are two major requirements, in general, for collecting the data sensed and transmitted by the sensor nodes in an MSN: the discovery of the current network topology and the protocol to transmit the messages.

To use a TDMA scheme, cluster-based topology (or a variant) is usually used. Different methods are proposed to form the clusters as well as to select the cluster leaders [9, 10]. However, the scope of this work does not include the formation of clusters. In this paper, we adopt the approach proposed by Kothapalli et al. [9].

The IEEE 802.11 protocol based on the carrier sense multiple access with collision avoidance (CSMA/CA) is currently the most popular transmission technique. In CSMA/CA, a node transmits a packet if it finds the medium to be clear. Otherwise, it follows an exponential back-off scheme and tries to transmit later. A receiver sends acknowledgment packets (ACK) to confirm the successful receipt of the packets. Two other messages, (Request To Send RTS) and (Clear To Send CTS) are also used before transmission for negotiation. Woo and Culler [11] use carrier sensing to avoid collisions. However, fairness may not be guaranteed because some nodes may always find the channel busy with other transmissions.

In TDMA protocol, a period of time called frame is divided into a fixed number of timeslots where every timeslot is dedicated to exactly one node to send its packet(s). In traditional TDMA protocols, the number of timeslots in a frame is equal to the number of nodes in a cluster when a cluster-based topology is used. This ensures collision-free message transmissions.

Traditional TDMA protocols work well for fixed networks. However, in MSNs, the membership of the sensor nodes in clusters may change due to the mobility of nodes and/or the switching of nodes between the so-called “sleep”

and “awake” modes. If some nodes move out of the current cluster and/or some node goes to the “sleep” mode (because it has nothing to transmit) for energy conservation, the slots allocated to them are still reserved but wasted. On the other hand, if some nodes move into the cluster and/or some nodes wake up from “sleep” mode, there may be shortage of slots to support their transmissions. Focusing on these issues, Herman and Tixeuil [12] propose a TDMA approach. However, it may not adapt over time. It also requires a big overhead of managing the information of extended neighborhood of a node, which makes it difficult to implement [2].

Bruhadeshwar et al. [2] propose self-stabilizing TDMA (STDMA), adopting the proposal by Kothapalli et al. [9] to form the overlay network, a TDMA approach where the slots are divided in a hierarchical manner. First, blocks of timeslots in a frame are divided among the cluster heads of the clusters. Cluster heads then assign their allocated timeslots among the member nodes. Doing so, this approach prevents the possible interferences among the transmissions of nodes in different clusters. To assign the blocks of timeslots to the cluster heads, this approach directly follows the clustering in the overlay network. However, the STDMA approach uses a fixed number of timeslots and hence may fall in short of slots when the number of member nodes increases. It may not make efficient use of unused slots too, making unwanted delay in the network. Moreover, during the contention period of STDMA, there remain high chances of collisions, and the leader needs to send a status message regarding the slot usage which adds up as an overhead.

Overhead tolerant TDMA (OLT-TDMA) [13] is another approach where a frame has a fixed number of timeslots. A newly joined node first scans the whole frame to sense and make a list of the unused slots. It then randomly picks one slot and starts transmission in the following frame. However, if more than one node selects the same slot, their messages may collide. The leader then marks the slot in the “collision map” and sends this information to all the nodes. The nodes then avoid the marked slots. Hence, it may underutilize the available timeslots. When the number of nodes becomes higher than the available slots, the OLT-TDMA allows selecting a victim node, and then both the nodes contend for the same slot. This approach may not always guarantee the collision avoidance and good performance.

D-TDMA [14, 15] is a dynamic TDMA scheduling approach addressing collisions and interferences. It also applies strategies, such as allocating contiguous slots to a node for transmission and reception, to minimize switching overhead of the nodes. However, like other TDMA protocols having a fixed number of slots in a frame, D-TDMA may also underutilize the available timeslots when network density is low. Again, when the number of requested (sender, receiver) pair becomes too high to be allocated in a frame, the D-TDMA may not always find an appropriate slot to choose without hampering other nodes’ transmissions.

We have proposed a flexible TDMA (FDTMA) mechanism, which successfully overcomes the shortcomings of the other existing proposals in different membership conditions (a preliminary version of this proposal can be found in [16]). A new sensor can always be allocated a timeslot

without making any interference with other transmissions. The FTDMA allocates more slots to support more sensors when the number of sensors in a cluster grows higher. This minimizes the contentions, the number of packet losses, and the queuing delay, which ensures very good network performance. On the other hand, it decreases the number of timeslots if unused. This also minimizes unnecessary waiting time, which increases the throughput.

3. Proposed Method

3.1. Statement of Problem. In this work, we focus on a cluster-based protocol for propagating messages in MSNs. We can think of two issues related to such problems: discovering a so-called overlay network allowing efficient routing and the medium access control (MAC) protocol. The former one handles the problem of clustering the sensors in different clusters and selecting a leader for each of the clusters. It is much studied for the static networks. Some works are also found for the mobile nodes as presented earlier. In this work, however, we focus on the second issue—the transmission protocol.

For the discovery of the overlay network, we adopt the model proposed by Kothapalli et al. [9]. In this model, a backbone structure (the so-called overlay network) is established by efficiently making some clusters on sensors and selecting the cluster leaders. This model is based on the concept of *dominating set*. It is proved to be locally self-stabilizing and can converge quickly.

Using this overlay network, we focus on achieving a TDMA protocol that ensures interference-free transmissions of the nodes in a cluster. At the same time, a good protocol should also meet the desirable performance requirements in a network such as fairness, adaptability and throughput. No node should starve for timeslots to be assigned to it. A good overall network throughput should be maintained. In other words, the duration of control messages should be much less than the data transmissions. If some nodes leave the cluster due to mobility, energy conservation, or any other reason, the protocol should discover their unused timeslots and make use of them. On the other hand, the protocol should also welcome newly joined nodes in a cluster by assigning timeslots for their necessary transmissions. The scheme should also be self-stabilized, which can work without any central control.

Hence, the allocation of timeslots directly influences the network performance. Furthermore, proper timeslot allocation also ensures collision-free communication. So, an efficient way of scheduling the timeslots among the members of a cluster is required to maximize the network performance.

3.2. The Proposed Approach. TDMA protocols usually consider a “frame,” which is a period of time split into some equal intervals called timeslots. Each timeslot is designated to a transmitting node (sensor) to transmit its data. The frame structure is repeated over time. Hence, the nodes get the chances to transmit in a round-robin fashion.

The problem of the most of the TDMA protocols is that they use fixed number of timeslots in a frame. Hence, the

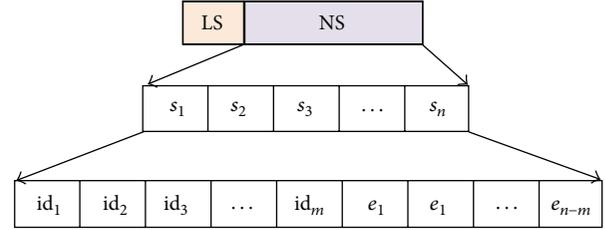


FIGURE 1: The structure of an FTDMA frame.

protocol may fall in shortage of timeslots to allocate to the newly joined nodes so that they may also perform their transmissions. Moreover, if some nodes are not transmitting, because of reasons like having no data to transmit, the timeslots allocated to them may also remain unutilized. Focusing on these shortcomings, we propose a flexible time division multiple access (FTDMA) protocol, which can make a better use of the timeslots and provide guaranteed support to the newly joined nodes’ transmissions.

3.2.1. The Frame Structure. The basic structure of the frame used in the proposed FTDMA is much similar to most of the conventional TDMA protocols. An FTDMA frame has two parts: the leader segment (LS) and the node segment (NS). The leader of a cluster broadcasts the schedule information in the LS. The NS is composed of n timeslots of equal duration, and the duration of a slot is set to transmission/reception time of one (or more) data packets along with the ACKs. According to the scheduling information broadcasted by the leader during the LS, the sensor nodes in that cluster may transmit their data during their allocated timeslots in the subsequent NS.

It is noteworthy to mention here that a message may sometimes take longer time in MSN (than the static network) to reach the receiving sensor due to the increase in relative physical distance between the sender and the receiver caused by mobility of the sensors. When the transmission ranges of the nodes in a network are small, which is true for the low-power devices such as sensors, this delay is usually negligible. However, if such propagation delays become significant, the cluster leader can handle it by increasing the duration of the timeslots. In such a case, slot duration can be included in the message broadcasted by the leader during LS.

Unlike the traditional TDMA approaches, the number of timeslots, n , in the NS of FTDMA frame may vary based on the cluster’s membership condition. FTDMA increases (or decreases) the value of n if some sensor nodes join (or leave) the cluster. This provides with the opportunity to successfully allocate timeslots to the newly joined nodes and reduce the possibility of idle slots, and thus FTDMA enhances the network performance.

Among the n timeslots in the NS, m slots are scheduled to m sensor nodes’ transmissions, and the rest $n - m$ slots are kept empty to be used by the newly joined nodes (to whom no slot has been assigned). The scheduling is described later in detail. The structure of the proposed TDMA frame is presented in Figure 1.

3.2.2. The Frame Usage

(1) *Role of Leader.* In the LS of a frame, the leader of a cluster may broadcast its message containing the scheduling of the timeslots to be used by the sensors in the following NS of that frame. Simply stating, a leader allocates a slot in the subsequent NS for each *willing* (i.e., willing to transmit) sensor in its cluster. A sensor is considered *willing* if the leader has received at least one message during the previous three (this value may be preset by the network administrator depending on the mode of application) frames. If there are m *willing* sensors in the cluster, the leader puts n (where $n > m$) timeslots in the subsequent NS of the frame. The first m slots are allocated to the *willing* sensors, and the rest $n - m$ slots give the room to the newly incoming sensors (see Figure 1). The format of the message broadcasted by the leader contains the information shown in Figure 2. It contains the values n and m followed by m IDs of the sensors, which are allocated a timeslot in the subsequent NS.

Here r represents the number of rounds that a frame will repeat without any further LS from the leader. This is especially helpful when a small number of nodes transmit in a cluster. In such cases, the LS becomes an overhead if there is no change in cluster membership. The use of r thus improves the channel utilization avoiding the LS in the subsequent $r - 1$ rounds of the frame. As the next LS message (i.e., allocating slots to any newly enjoined sensor in the cluster) may happen after at least r rounds of the frame, r should be set at a value that does not notably degrade the QoS for the newly joined nodes. If the leader senses a collision (among the REQUEST messages as explained later) during any of the empty slots, it may set $r = 1$ in its next broadcast in LS so that the newly joined nodes may try to send REQUEST messages in every frame.

Upon successful receipt of a REQUEST message from a sensor, a leader sends an ACK to it assuring that the corresponding sensors will be allocated a timeslot in the upcoming frame. If another sensor having the same ID as this new sensor already exists in the cluster, the leader also assigns and sends a new ID together with the ACK.

(2) *Role of Member Sensors.* Sensors in a cluster may be categorized as *unwilling* (having no data to send), *waiting* (a newcomer sensor who has not yet tried to get a timeslot), *willing* (a newcomer sensor who has been trying to get a timeslot but not received ACK from the leader yet), and *active* (sensor that has been assigned a timeslot or received an ACK from the leader) state sensors.

An *active* sensor scans (in listening mode) for a broadcast message from the leader during the LS of a frame. Upon getting the message, it searches its ID in it. If it finds the ID in the i th position of the broadcasted message, it transmits its data during the s_i slot of the NS. This ensures no collision with the transmission of other sensors.

A *waiting* sensor keeps scanning (because it has no idea about the frame as well as the LS in the newly joined cluster) for a broadcast message from a leader. After receiving a broadcast, it first gets the time information and synchronizes its clock with that of the leader. It then changes its status to

r	n	m	id_1	id_2	id_3	\dots	id_m
-----	-----	-----	--------	--------	--------	---------	--------

FIGURE 2: The information included in the message broadcasted by a leader during LS of a frame.

willing. Note that it can send a REQUEST message in the same frame.

According to the message broadcasted by the leader, r round(s) of $n - m$ empty (free) slots are going to appear. A *willing* sensor randomly (uniform random) selects a slot e_j from the $r(n - m)$ empty (free) slots and transmits a REQUEST message during the slot e_j . If it gets an ACK from the leader, it changes its status to *active* at the end of the r th round of frame. It also changes its ID if there is such instruction with the ACK.

An *unwilling* sensor may keep its radio off (some application, however, may require the radio to be on in order to receive necessary control signals) to conserve energy (a sensor at any other status performs only in one slot of NS. In the rest of the time it may also switch off its radio to preserve energy). When it has some data to transmit, it switches to the *waiting* state.

Note here that there is chance of collision of the transmissions if more than one sensor chooses the same empty slot e_j . The procedure to handle such situations is described in the next subsection.

Another point is that a sensor may also send a data packet instead of the REQUEST message. This may increase the network throughput a little, though at the cost of energy, if the transmission is unsuccessful due to collision.

3.2.3. Handling Membership Changes

(1) *Incoming Nodes.* To support the newly joined nodes, the leader keeps $n - m$ empty slots in the NS of a frame. In normal situation, we propose to put one empty slot on NS, that is, $n = m + 1$. This is because use of more empty slots may decrease the network throughput. When a leader successfully receives a REQUEST (or data packet) from a sensor in an empty slot without any collision, the leader allocates a slot for the sensor in the next frame (meaning that the sensor has joined the cluster successfully). At this point, it also decreases the value of n (ensuring the constraint $n > m$, of course). Otherwise, if a leader detects a collision during an empty slot, it increases the value of n (i.e., the number of empty slots) in the next frame. By increasing empty slots, the leader tries to minimize the probability of collisions in the subsequent frame. The general modification of the value of n is done using

$$n = m + \max(1, \text{LSES} * C_f), \quad (1)$$

where C_f is the number the previous frames where no newly joined sensor could be allocated slots due to collisions, and (Last Successful Empty Slots LSES) represents the value $\lceil (n - m)/2 \rceil$ calculated in the last frame when the leader could successfully receive REQUEST message(s) from the newly joined sensor(s). The LSES gives an estimate of the number of empty slots that may suffice to entertain the newly joined sensors. The value $\lceil (n - m)/2 \rceil$ ensures the increase

(or decrease) of empty slots based on the network conditions, especially the rate of incoming sensors. When there is no collision, using one empty slot suffices to welcome newly joined nodes. On the other hand, the number of empty slots is increased to resolve collision(s), if any.

The proposed FTDMA thus handles the joining requests of incoming nodes much better than the other available approaches. While most of the other approaches dedicate one portion of the frame for the newly joined nodes so that they may send the REQUEST messages based on traditional contention and back-off [2, 5], the leader increases the room for the new nodes. With more empty slots, it is more likely that more sensors can be facilitated facing less number of collisions. It effectively minimizes the joining time of a sensor to a cluster in FTDMA. This leads to a very simple and easy way of management.

When the number of nodes becomes too high (very dense network) in a cluster, the performances of the different existing protocols highly degrade which is also true for the proposed FTDMA. In such a network, the allocation of a slot for every sensor in FTDMA may lead to a very long frame. This may cause delay on the channel access for the sensors. This may be critical in a network where a delay-sensitive application is deployed. In such cases, a limit on the cluster size can be imposed so that a cluster is broken into smaller clusters. In an application where packet losses are not critical, a small frame may be used where a leader schedules different subsets of the sensors in different cycles of the frame. In this work, however, we assume a moderately dense network which is not delay sensitive.

(2) *Outgoing Nodes.* A member sensor of a cluster may be disconnected from the leader for a number of reasons such as turning the radio off for a while (entering the *unwilling* state), dying (battery power exhausted), and moving out of the cluster. In FTDMA, a “going-to-disconnect” sensor sends a DISCONNECT message (to the leader) in its allocated slot, and the leader does not allocate any slot for it in the following frames. However, such cases may not always be predicted by the sensors nodes, and hence a node may be disconnected without sending the DISCONNECT message. To handle such situations, we propose the leader to consider a sensor node as “disconnected” if it does not receive any message from the sensor in k number of contiguous frames ($k = 3$ works well in general). The value of k can be predetermined by the network administrator according to the characteristics and availability of sensory data.

When a sensor is disconnected, the leader does not further allocate any slot for it in the following frames. It also decreases the value of m that in turn decreases n to shrink the frame size. Thus FTDMA gets rid of unused timeslots. A smaller frame also iterates quicker, which provides the *willing* sensors to have their chances to transmit more frequently. This increases the overall throughput.

4. Performance Evaluation

We have evaluated the performance of the proposed FTDMA in comparison with the IEEE 802.11, the overload tolerant

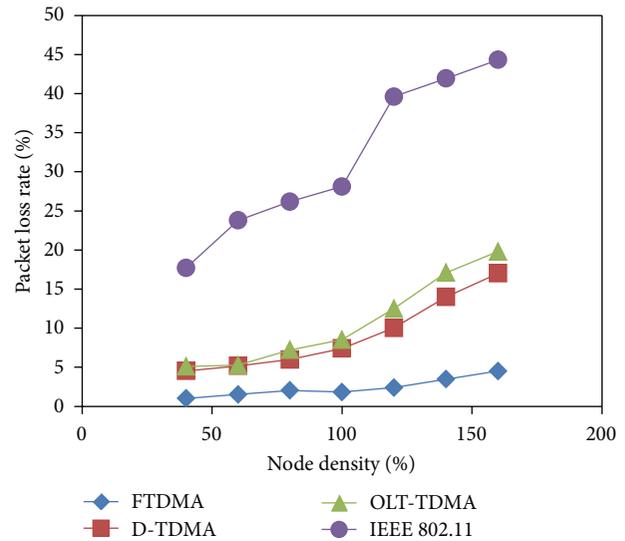


FIGURE 3: The packet loss ratios of the methods for different node densities.

TDMA (OLT-TDMA) [13], and the D-TDMA [14, 15] through simulation. In our simulation scenario, we have constructed a $400\text{ m} \times 300\text{ m}$ rectangular field, where the sensors can move freely in any direction. A location in the field is represented as a 2D vector (x, y) , where $x \in [0, 400]$ and $y \in [0, 300]$. Sensors have been deployed at different randomly selected locations on the field. The movement directions of sensors are also picked randomly from 0 to 360 degrees. After the start of simulation, no sensor goes out of the field. If any sensor approaches the boundary, it changes its direction following the law of reflection with respect to the boundary. Sensors move at a constant speed (our simulations, however, include different speed scenarios too).

We adopt the “Constant density spanners” approach proposed by Kothapalli et al. [9] to form the clusters and leaders and applied all the TDMA methods on the same cluster. For all the methods, the duration of the control segment (e.g., LS) of a frame is set to 14.8 ms while each timeslot assigned to a sensor is of 3.52 ms. Hence, a slot can accommodate the data transfer of 20 packets, each having the size of 44 Bytes. The number of slots in a frame is set to 10 for both OLT-TDMA and D-TDMA. A simulation has been run for 50 s. Each simulation scenario is repeated 30 times and the presented results show the average of these 30 runs.

For the first experiment, we have allowed the sensors’ speeds to be selected randomly from the range [1, 10]. In our simulations, we have considered seven different node densities (the average number of nodes in a cluster) ranging from 40% to 160%. The approximate node densities have been achieved by increasing/decreasing the total number of sensors deployed on the field. Since D-TDMA and OLT-TDMA follow the TDMA technique with fixed number of timeslots, the node density can be considered 100% if there are approximately 10 sensors per cluster.

Figure 3 demonstrates the comparative performances of the three mechanisms in terms of packet loss ratio. It shows

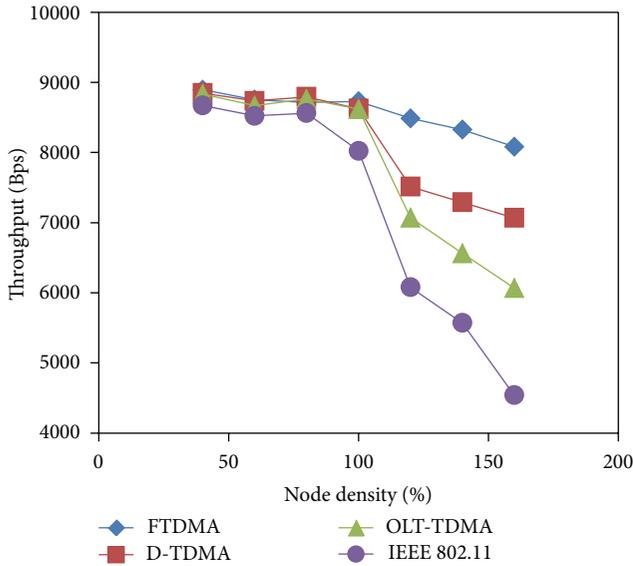


FIGURE 4: Throughput achieved by the methods for different node densities.

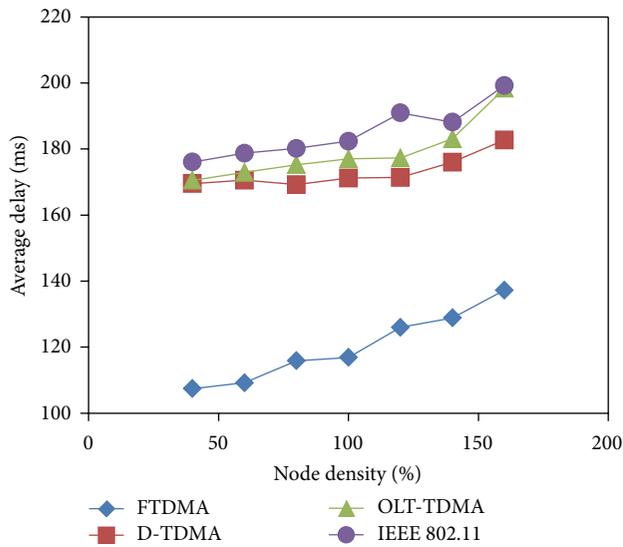


FIGURE 5: The average delays of the methods for different node densities.

that the packet loss rises with traffic load. However, the packet loss rate does not grow that much for the proposed FTDMA mechanism while it grows faster in the other protocols. The throughputs shown in Figure 4 also demonstrate the better performances of the FTDMA.

Figure 5 shows the average delay of all the sensors with different methods. The increase of traffic loads also increases the average queuing delay. However, with the FTDMA, the delay is significantly lower. The maximum delays shown in Figure 6 also show the similar performances.

In another experiment, we have set the node density to 100% and allow the sensors to have different speeds. In every scenario, we allow the sensors to randomly pick a

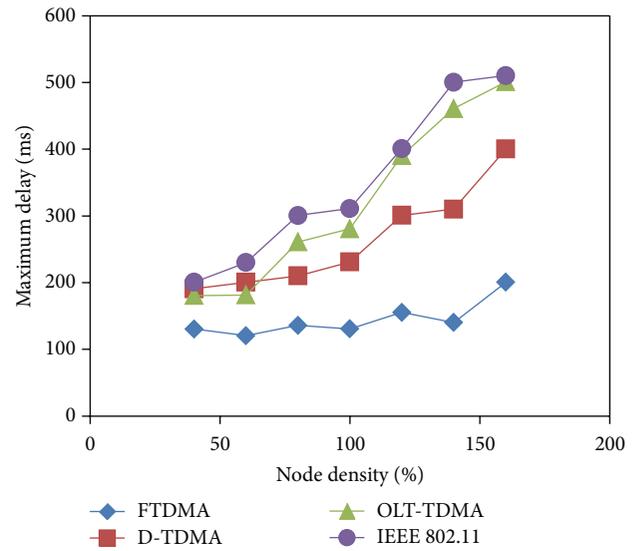


FIGURE 6: The maximum delays of the methods for different node densities.

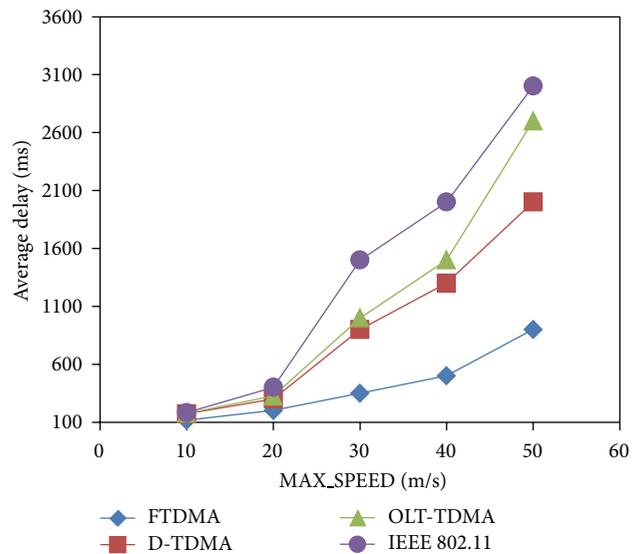


FIGURE 7: The average delays of the methods for different speeds of sensors.

speed from the range $[1, \text{MAX_SPEED}]$, and we vary the MAX_SPEED in different scenarios. Figure 7 shows that the average delay increases with the increase of speeds of the sensors. However, the proposed FTDMA faces significantly lower delay compared to the other methods.

With the OLT-TDMA, a new node must listen to one whole frame to find an empty timeslot for it. However, with the proposed FTDMA, a node needs to listen only to the LS part of a frame, and it can choose an empty slot from the $n - m$ slots in that very frame for its transmission. If more than one sensor node selects the same slot in OLT-TDMA, a collision is detected by the leader and it marks the corresponding timeslot in the collision map in the subsequent frame.

The new joining nodes avoid this slot as there was a collision. This makes underutilization of the slots where some collisions occurred. Another delay occurs to execute the LIF algorithm for choosing a victim. And sharing a slot with the victim node requires a contention-based channel access, which may incur collisions and subsequent backoff and delay. The D-TDMA approach also faces similar shortcomings when the number of sensors per cluster increases. Furthermore, in a light-load scenario, many slots are likely to remain unused. On the other hand, all these problems are overcome by the proposed FTDMA. It minimizes the number of slots in low traffic conditions, resulting in the reduction of unnecessary slots, and maximizes the number during high node densities, ensuring less collision. All these desired points add to the better performance of the proposed FTDMA mechanism over the other existing ones.

5. Conclusion

In this paper, we focus on overcoming the shortcomings of the existing TDMA approaches and propose a flexible TDMA (FTDMA) approach, which works similar to that of the traditional TDMA approaches but with the exception that the number of timeslots in an FTDMA frame may vary depending on the cluster membership conditions. The protocol is scalable in clusters and can solely be controlled by the cluster leaders in a distributed manner. As future work, we plan to develop a general TDMA protocol that may work in different network topologies. We also have a plan to work on highly dense and delay-sensitive networks in future.

Acknowledgments

The authors would like to express their deep gratitude to the anonymous reviewers for their constructive comments that have helped to correct the flaws in the proposal. This work was supported by Hankuk University of Foreign Studies Research Fund of 2013.

References

- [1] Md. Abdul Hamid, M. Abdullah-Al-Wadud, and I. Chong, "A schedule-based multi-channel MAC protocol for wireless sensor networks," *Sensors*, vol. 10, no. 10, pp. 9466–9480, 2010.
- [2] B. Bruhadeshwar, K. Kothapalli, and I. R. Pulla, "A fully dynamic and self-stabilizing TDMA scheme for wireless ad-hoc networks," in *Proceedings of the 24th IEEE International Conference on Advanced Information Networking and Applications (AINA '10)*, pp. 511–518, April 2010.
- [3] W. Crowther, R. Rettberg, D. Waldem, S. Ornstein, and F. Heart, "A system for broadcast communications: reservation ALOHA," *Proceedings of the 6th Hawaii International Conference on System Sciences*, 1973.
- [4] F. Borgonovo, A. Capone, M. Cesana, and L. Fratta, "RR-ALOHA, a reliable R-ALOHA broadcast channel for ad-hoc inter-vehicle communication networks," in *Proceedings of the Med-Hoc-Net*, Baia Chia, Italy, 2002.
- [5] A. Jhumka and S. Kulkarni, "On the design of mobility-tolerant TDMA-based Media Access Control (MAC) protocol for mobile sensor networks," *Proceedings of the 4th International Conference on Distributed Computing and Internet Technology*, vol. 4882, pp. 42–53, 2007.
- [6] G. K. Wong and X. Jia, "An efficient scheduling scheme for hybrid TDMA and SDMA systems with smart antennas in WLANs," *Wireless Networks*, vol. 19, no. 2, pp. 259–271, 2013.
- [7] F. Shad, T. D. Todd, V. Kezys, and J. Litva, "Dynamic slot allocation (DSA) in indoor SDMA/TDMA using a smart antenna basestation," *IEEE/ACM Transactions on Networking*, vol. 9, no. 1, pp. 69–81, 2001.
- [8] R. Kuehner, T. D. Todd, F. Shad, and V. Kezys, "Forward-link capacity in smart antenna base stations with dynamic slot allocation," *IEEE Transactions on Vehicular Technology*, vol. 50, no. 4, pp. 1024–1038, 2001.
- [9] K. Kothapalli, C. Scheideler, M. Onus, and A. Richa, "Constant density spanners for wireless ad-hoc networks," in *Proceedings of the 70th Annual ACM Symposium on Parallelism in Algorithms and Architectures*, pp. 116–125, July 2005.
- [10] M. Demirbas, A. Arora, V. Mittal, and V. Kulathumani, "A fault-local self-stabilizing clustering service for wireless ad hoc networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 17, no. 9, pp. 912–922, 2006.
- [11] A. Woo and D. E. Culler, "A transmission control scheme for media access in sensor networks," in *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking*, pp. 221–235, July 2001.
- [12] T. Herman and S. Tixeuil, "A distributed TDMA slot assignment algorithm for wireless sensor networks," *Proceedings of the 7th International Symposium on Algorithms for Sensor Systems, Wireless Ad Hoc Networks and Autonomous Mobile Entities*, vol. 3121, pp. 45–58, 2004.
- [13] R. Wang, Z. Wang, Z. Chen, and L. Zhang, "A 3G-802.11p based OLT-TDMA mechanism for cooperative safety in a dense traffic scenario," in *Proceedings of the 73rd IEEE Vehicular Technology Conference (VTC '11)*, pp. 1–5, Budapest, Hungary, May 2011.
- [14] Y. Wang, P. Shi, K. Li, and Z. Chen, "D-TDMA: An approach of dynamic TDMA scheduling for target tracking in wireless sensor networks," in *Proceedings of the IEEE/ACM International Conference on Green Computing and Communications (GreenCom '10), IEEE/ACM International Conference on Cyber, Physical and Social Computing (CPSCom '10)*, pp. 546–553, December 2010.
- [15] Y. Wang, P. Shi, K. Li, and Z. Chen, "An energy efficient medium access control protocol for target tracking based on dynamic convey tree collaboration in wireless sensor networks," *International Journal of Communication Systems*, vol. 25, no. 9, pp. 1139–1159, 2012.
- [16] M. Abdullah-Al-Wadud, "A TDMA scheme for vehicular Ad Hoc networks," *Green University Review*, vol. 3, no. 2, pp. 53–56, 2012.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

