RESEARCHERID
THOMSON REUTERS

[ICRTCET-2018]

ISSN 2348 – 8034
Impact Factor- 5.070

GLOBAL JOURNAL OF ENGINEERING SCIENCE AND RESEARCHES

M - CAMILS BASED OPTIMIZED COLLISION-FREE LINK SCHEDULING IN WIRELESS SENSOR NETWORKS

Naveen K V[*1] & Anand M[2]
[*1&2]Member, IEEE

ABSTRACT

In the 5G era, Internet of things (IoT) will find wide spread applications in all domains of human life. Wireless Sensor network (WSN) becomes a very important topic of research for successful deployment of IoT. Within wireless sensor network, the scheduling algorithm within the MAC protocol determines the time slot allocation of WSN nodes activity. There are many scheduling algorithms proposed in the literature. One such protocol which has gathered immense attention is the Distributed Degree-Based Link Scheduling (DDLS) algorithm which works by identifying a Collision Avoidance Maximal Independent Set (CAMILS) of WSN nodes. This paper proposes Maximal - CAMILS (M-CAMILS) based optimized collision - free link scheduling scheme to further optimize the time slot allocation. The M-CAMILS algorithm will identify links which are two hops away from the time slot allocated links such that scheduling length can be decreased further and transmission delay and duty cycle can be reduced. The comparison of time slot allocation pattern between the proposed M-CAMILS based link scheduling and the DDLS algorithm for static symmetric networks deployed in a grid arrangement were made. The proposed algorithm achieved an average 10% improvement in time slots required for the same amount of data transmissions.

Keywords: *Distributed wireless sensor networks, link scheduling, collision avoidance*.

I. INTRODUCTION

5th Generation wireless communication systems (5G) are expected to be a reality in few years and Internet of Things (IoT) is going to be a major use case of these networks by bridging the gap between digital and physical systems to strive towards ever more automation and autonomous decisions in the environment like smart grids, smart factories, smart buildings, smart cities, smart farming, connected vehicles, medical IoT etc. Device to device communication and wireless sensor networks (WSN) will take the center stage of this ultra

connectivity paradigm. Huge numbers of IoT based products are already in offering and this number is expected to grow to billions within next 4 - 5 years. Many of the existing network technologies aren't fit to enable this hyper - connected environment, leveraging IoT on a real-time and massive scale. Researchers, scientists and engineers are facing the challenges of designing new communication technologies to meet this ubiquitous computing and connectivity requirement [1].

MAC layer is the main responsible for achieving this ubiq-uitous computing and connectivity requirement at low energy consumption. In MAC protocol, interference and collision hin-der efficient communication by causing retransmission.

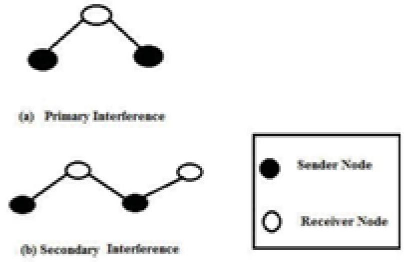

Fig. 1. Primary and Secondary Interference in WSN

There is primary interference where a node receives messages from multiple neighbouring nodes at the same time and secondary interference where communication between a pair of nodes interferes with a neighbouring pair as shown in Fig. 1. Hence, scheduling of transmission between neighbouring nodes has to be done correctly to reduce interference and improve efficiency [2].

Many CSMA and TDMA based scheduling algorithms have been proposed for medium access in wireless sensor networks [3]–[7]. CSMA based techniques, requiring constant listening of the medium, increase energy consumption and reduce life span of nodes. So, TDMA based algorithms seem to be more practical. TDMA based schemes can either schedule communication of nodes (node scheduling) [7], [8] or of links between the nodes (link scheduling) [9]–[11].

Among the link scheduling algorithms, DCLS scheme [10] has received great attention in the recent past. This algorithm avoids interference by assigning different time slots to con-tinuously connected three links using graph colouring theory. However, since the algorithm depended heavily on random factors, the time to finish link scheduling, the network delay time and the energy efficiency varied from network to network even under the same parameters.

DDLS scheme with low latency and low duty-cycle [11] came as a big improvement over the DCLS scheme. DDLS scheme reduces scheduling length by assigning the same time slot to multiple links without collision by identifying a Collision Avoidance Maximal Independent Set (CAMILS). CAMILS is basically a set of links which do not interfere with each other even while communicating at the same time.

CAMILS is identified based on the ”largest sum of degree first”. Sum of degree is calculated as the sum of degrees of the node and its one hop neighbours. The node with the largest value of sum of degree among its one hop neighbours or node having highest node id among nodes with the same sum of degree among one hop neighbours will get the opportunity to allocate time slot to a link. Time slot allocation using DDLS scheme for a 4x4 WSN is shown in Fig. 2. Initially, all nodes calculate their sum of degree and collect sum of degree of neighbours. Then, nodes compare their sum of degree with that of one hop neighbours and node with the largest value of sum of degree among its one hop neighbours or node having highest id among nodes with same sum of degree will get the opportunity to allocate time slot to its link with the neighbour node having next largest sum of degree. In the figure, node ids are represented as 1, 2, 3 etc and sum of degree are shown inside parenthesis near the corresponding nodes. Nodes 6, 7, 10 and 11 have the same sum of degree 16. Hence, node 11, having the highest node id among nodes having the same sum of degree value, can allot the time slot. So, node 11 will communicate to node 10 in the first time slot as per the DDLS algorithm. Thus, link between nodes 11 and 10 is a CAMILS for the time slot and can communicate without any primary and secondary interference.

However, links between nodes 5 and 1, and 8 and 4 can also communicate in the same time slot without causing any interference to link between nodes 11 and 10. Thus, there are links which can be included in the CAMILS to get a maximal - CAMILS (M-CAMILS) such that they can be allotted the same time slot without causing any interference thereby optimizing the scheduling length. Hence an algorithm to identify an M

CAMILS to optimize collision - free link scheduling in wireless sensor network is presented in this paper. Only static symmetric grid based sensor networks are considered. Grid based sensor networks are useful in applications where exact physical position of nodes is pre-determined like smart grid, industrial IoT, smart farming, outdoor environment monitoring etc. The proposed algorithm assigns only one time slot to one link to ensure fairness similar to the DDLS method.

Remaining paper is arranged as four sections. Section II discusses preliminaries of DDLS algorithm and Section III presents the proposed scheme. Section IV presents simulation results and performance comparison of the proposed and DDLS algorithm and finally, section V conclusion.

RESEARCHERID
THOMSON REUTERS
[ICRTCET-2018]
ISSN 2348 – 8034
Impact Factor- 5.070

II. PRELIMINARIES

This paper considers wireless sensor nodes deployed in a symmetric grid arrangement. All the nodes have an omni-directional antenna with uniform transmission range. Nodes which are within the transmission range can communicate with each other. Nodes communicate by message broadcast which is received by all one hop neighbours within the transmission range. Collision happens when two nodes within the transmission range communicate simultaneously (primary interference) or when communication between a pair of nodes unintentionally interferes with another pair (secondary inter-ference).

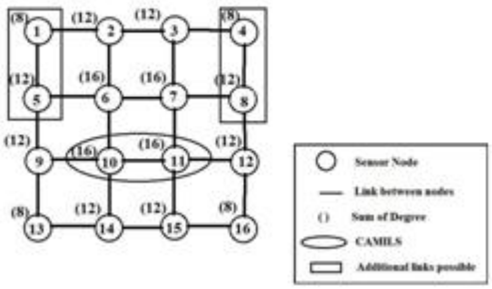

Fig. 2. Time slot allocation based on DDLS algorithm for a 4x4 WSN

All nodes are synchronized in time and can have bidirec-tional communication in the allotted time slot. Each node can be in any of the following states at any given instant: READY, WAIT, BLOCKED, PAIRED and COMPLETE. State of the node changes when it receives message from another node or internal message from within. Each node will try to make a transmission pair with one of its neighbour nodes and assign a time slot to the link between them during the ’work period’. At the end of each work period, every node sends an internal message to itself, analyses link information and changes state to READY if links are still to be allotted time slot else changes to COMPLETE state. In COMPLETE state, all the links of the node have been assigned time slots. Nodes in COMPLETE state ignore all received messages.

Initially, each node has its id and degree information. Each node broadcasts its degree information to its neighbours and collects their degree information. Nodes receiving degree information will get information about its two hop neighbours also. Thus sufficient information of three continuous links is obtained.

Each node receiving the degree information of its neigh-bours will then calculate the sum of degree *DS*. *DS* is sum of the degrees of the node and its one hop neighbours. Then, each node will send its *DS* to all neighbours and collects their *DS* and stores them in a lookup table with respect to the corresponding node ids.

III. PROPOSED SCHEME

In this section, detailed description of the proposed M-CAMILS based optimized collision - free link scheduling scheme is presented.

Let V represents the set of all the nodes in the WSN and v the individual nodes. All neighbours of the node v are represented as u. Nodes can be in READY, WAIT, BLOCKED, PAIRED or COMPLETE state at any given instant. Nodes change the state upon receiving messages from other nodes or internal message from within.

This paper introduces a SLOT ALLOT property of nodes to identify links which are two hops away from the time slot allocated links. The identified link and time slot information are stored as an *extra slot msg* in the message queue of the node. _ _

The algorithm runs on all the nodes at the same time. Initially, SLOT ALLOT and message queue are empty. All nodes enter READY state after receiving the internal message *internal msg contend*. In READY state, nodes compare their sum of degree (*DS*) with that of one hop neighbours to check whether they can assign time slot. If *DS* is largest among one hop neighbours or *DS* are equal with node id v greater than u, node v sends a *join req msg* to node u and moves to the WAIT state.

Node u receiving join req msg checks whether DS of v is larger than its DS or node id v is greater than it's id with DS being equal. If the condition satisfies, node u will perform make pair procedure directly instead of sending join accept msg as in DDLS algorithm thereby improving scheduling time. In make pair procedure, node assigns the time slot to the link, decrease its DS by 2 and sends pri-mary join msg to its one hop neighbours with the updated

DS value and node ids of one hop neighbours other than v. If node u doesn't find node v to be eligible, it will simply discard join req msg and stay in READY state id with *DS* being equal. If the condition satisfies, node u will perform make pair procedure directly instead of sending *join accept msg* as in DDLS algorithm thereby improving scheduling time. In make pair procedure, node assigns the time slot to the link, decrease its *DS* by 2 and sends *pri-mary join msg* to its one hop neighbours with the updated

DS value and node ids of one hop neighbours other than v. If node u doesn't find node v to be eligible, it will simply discard *join req msg* and stay in READY state.

One hop neighbours of node u other than v receiv-ing *primary join msg* perform primary update procedure and transits to BLOCKED state. In primary update procedure, nodes will decrease their *DS* by 1 for each received *pri-mary join msg*, update the *DS* of sender node in their lookup table and changes their state to BLOCKED from READY. Nodes in WAIT state receiving *primary join msg* from node other than the expected node will also perform primary update procedure and transit to BLOCKED state.

At the end of primary update procedure, nodes send *sec-ondary join msg* to their one hop neighbours to update their *DS*.

In order to avoid interference caused by all nodes send-ing *secondary join msg* together, M-CAMILS algorithm pro-poses an "update period". "update period" is the time taken by one hop neighbours to send *secondary join msg* one after the other in the order mentioned in the received *primary join msg*. "update period" duration increases linearly with increase in the maximum degree of the net-work. Nodes receiving *secondary join msg* perform sec-ondary update procedure. Nodes update the *DS* of sender nodes in their lookup table and increase their SLOT ALLOT value by 1 for each received *secondary join msg*.

Node v in WAIT state, upon receiving *primary join msg* from u performs make pair procedure and wait during "up-date period" to avoid interference with one hop neigh-bours sending *secondary join msg*. Then, node sends *pri-mary join msg* with updated *DS* value. One hop neigh-bours of node v in READY or WAIT state will perform primary update procedure after receiving *primary join msg*.

Nodes in PAIRED or COMPLETE state discard the mes-sages. Nodes receiving *secondary join msg* performsec-ondary update procedure and increase their SLOT ALLOT TABLE I

COMPARISON OF DDLS AND M-CAMILS ALGORITHMS

DDLS Scheme	M-CAMILS Scheme
Assumption	
Nodes have STATES (READY, WAIT, BLOCKED, PAIRED or	Nodes have STATES (READY, WAIT, BLOCKED, PAIRED or

ISSN 2348 – 8034
Impact Factor- 5.070

COMPLETE)	COMPLETE), SLOT ALLOT (to look for links two hops away of time slot assigned links)and a message queue (to store messages).
Algorithm	
Nodes check whether their sum of degree is largest among their one hop neighbours. Node with largest sum of degree or highest node id among nodes with same sum of degree will send *join req msg*.	Nodes check whether their sum of degree is largest among their one hop neighbours. Node with largest sum of degree or highest node id among nodes with same sum of degree will send *join req msg*.
Corresponding node receiving *join req msg* verifies the condition and sends *join accept msg*, if true.	Corresponding node receiving *join req msg* verifies the condition and if true, performs make pair procedure, changes state to PAIRED and sends *primary join msg*.
Above sender and recipient nodes perform make pair procedure, update their STATE as paired and send *primary join msg* to their one hop neighbours.	Upon receiving *primary join msg*, node which send *join req msg* performs make pair procedure and one hop neighbours performs primary update procedure and send *secondary join msg* in the order as mentioned in received message to avoid interference during the "update period".
One hop neighbours receiving *primary join msg* performs pri-	Node in PAIRED state now sends *primary join msg* to its one hop

ISSN 2348 – 8034
Impact Factor- 5.070

mary update procedure and send *secondary join msg.*	neighbours and nodes which received *secondary join msg* perform secondary update procedure and update their SLOT ALLOT by 1.
Nodes which received *secondary join msg* perform secondary update procedure.	One hop neighbours which received *primary join msg* performs primary update procedure and send *secondary join msg* in the order mentioned in the received message to avoid interference.
At the end of "work period", nodes send *internal msg contend* and analyse link information. Nodes with links yet to be assigned time slot will go to READY state and others to COMPLETE state.	At the end of "work period", nodes send *internal msg contend* and analyse link information. Nodes having SLOT ALLOT value 2 identifies the node with next largest sum of degree and didn't send *secondary join msg* and stores the link information and node id as *extra slot msg* in the message queue (extra slot procedure).Nodes with links yet to be assigned time slot will go to READY state and others to COMPLETE state.
Nodes in READY state again tries to identify time slots for their links.	Nodes in READY state again tries to identify time slots for their links. Nodes which per-

ISSN 2348 – 8034
Impact Factor- 5.070

	formed extra slot procedure will send the *extra slot msg* along with the next message and performs make pair procedure. Nodes re-ceiving the *extra slot msg* along with other message will assign the corresponding time slot to the link and update the sum of degree.

value by 1.

M-CAMILS algorithm proposes an extra slot procedure at the end of each work period. Nodes send an in-ternal message *internal msg workperiod* to indicate the end of "work period". Then, each node updates the in-formation and performs extra slot procedure. During ex-tra slot procedure, each node will check their SLOT ALLOT value. SLOT ALLOT value two indicates that the node is two hops away from the time slot allocated link and can communi-cate in the same time slot. Hence, nodes with SLOT ALLOT value two will check for node with largest sum of degree among its neighbours which didn't send *secondary join msg* and link with which is not assigned time slot. The identified link and time slot are then stored as *extra slot msg* in the message queue. Nodes then transits to READY state if still links are present to be allocated time slot else transits to COMPLETE state.

Algorithm proceeds to next time slot and again, nodes in the READY state try to assign time slot to links with their neighbours. Node v which performed extra slot procedure in the previous time slot will have to assign the time slot to the link with the identified node u and update the neighbours regarding the slot assignment. Hence, when v gets opportunity next time to send message, it will decrease its *DS* by 2 and send *extra slot msg* and updated *DS* also along with it. Upon receiving *extra slot msg*, node u will assign the time slot in the message to the link, decrease its *DS* by 2 and updates *DS* of the sender node in it's lookup table. All other neighbours of node v will decrease their *DS* by 1 and update sender *DS* in the lookup table. Nodes which are neighbours to both v and u will decrease their *DS* by 2. The updated *DS* are send to neighbours in the next messages so that link scheduling for the time slot will proceed normally.

Algorithm continues to run till all the links in the network are assigned time slots. M-CAMILS algorithm ensures fairness in link scheduling by assigning one link only one time slot. This also means that the time taken by the same sender node to send a packet to the same receiver node will be same as scheduling length. Thus, by achieving optimized link scheduling, M-CAMILS is also optimizing network latency.

IV. SIMULATION AND RESULTS

This section presents the simulation results of the proposed M-CAMILS scheme and compares the time slot allocation pattern with DDLS scheme under the same simulation envi-ronment. Only networks in a symmetric grid arrangement M x M were considered. All sensor nodes are connected and the maximum degree for the network is 4. The simulations are done using MATLAB R2015b.

M-CAMILS algorithm identifies a Maximal - Collision Avoidance Maximal Independent Set by identifying links two hops away from the CAMILS for the same time slot. Tables II and III show the time slot allocation pattern of

ISSN 2348 – 8034
Impact Factor- 5.070

DDLS and M-CAMILS schemes for a 4x4 WSN. Links which communicate in the same time slot are shown in the same row as the time slot. In the first time slot, only link 11 - 10 is scheduled in

Table II time slot allocation pattern for ddls scheme

Time Slot	CAMI LS		
1	11- 10	-	-
2	7- 6	-	-
3	10 - 6	-	-
4	9- 5	11 - 7	-
5	3- 2	12 - 8	15- 14
6	6- 5	8- 4	14- 13
7	2- 1	10 - 9	16- 15
8	5- 1	7- 3	14- 10
9	4- 3	12- 11	9 - 13
10	6- 2	15- 11	-
11	8- 7	-	-
12	12- 16	-	-

Table III time slot allocation pattern for m - camils scheme

Time Slot	M - CAMILS		
1	11- 10	5- 1	8 - 4
2	7- 6	9 - 13	1 2- 16
3	10- 14	7- 3	-
4	6- 2	11- 15	-
5	10 - 9	12 - 8	3- 2
6	6- 5	15- 14	3- 4
7	9- 5	11 - 7	-
8	2- 1	14- 13	7- 8
9	10 - 6	16- 15	-
10	12- 11	-	-

Table IV scheduling length of ddls and m-camils scheme for different wsn sizes

WSN Size	No. of Nodes	DDLS Time slots	M - CAMILS Time Slot	Percentage Improvement
4x4	16	12	10	16.7
6x6	36	16	13	23
8x8	64	18	16	11.11
10x10	100	20	19	5
12x12	144	21	20	4.7
14x14	196	22	20	9
16x16	256	24	23	4.2
18x18	324	27	24	11.11
20x20	400	27	26	3.7

CAMILS (table II) where as links 11 - 10, 5 - 1 and 8 - 4 are scheduled in M-CAMILS (table III). Thus, comparing tables II and III, it is found that M-CAMILS algorithm assigns the same time slot to more links compared to previous DDLS algorithm. M - CAMILS finished link scheduling in 10 time slots compared to 12 time slots for DDLS giving a 16.7% improvement in scheduling length where scheduling length is the total number of time slots required to assign all the links of the network.

Table IV and Fig. 3 both shows further comparison of scheduling length for the two schemes for different network sizes. From table IV, it is found that M - CAMILS out performs the DDLS scheme for all the network sizes. The maximum improvement achieved is 23% for 6x6 WSN and average improvement is 10% against DDLS scheme.

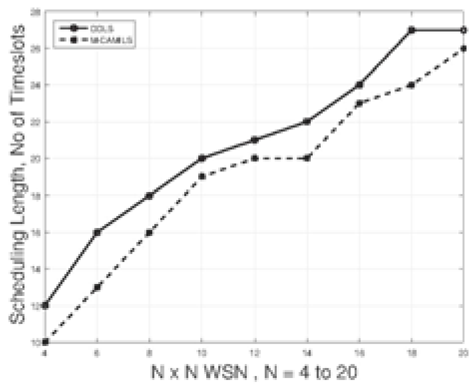

Fig. 3. Scheduling length comparison

V. CONCLUSION

This paper presents the M-CAMILS based collision - free algorithm for optimizing link scheduling in wireless sensor networks. M-CAMILS optimize link scheduling by assigning the same time slot to links which are two hops away from the time slot assigned links. Selecting links two hops away ensures collision - free communication by avoiding primary and secondary interference. M-CAMILS demonstrated an average 10% improvement in comparison with the DDLS scheme. Thus by improving the scheduling length without compromising on interference, M - CAMILS algorithm boost efficiency and shorten transmission delay.

In this paper, only sensor networks deployed in symmetric grid arrangement were considered. In the future work, the algorithm would be extended to randomly deployed network.

REFERENCES

1. W. Ejaz et al., "Internet of Things (IoT) in 5G Wireless Communications," in IEEE Access, vol. 4, pp. 10310-10314, 2016
2. B. S. Krishnan, M. Ramaswamy, and A. Nachiappan, "Optimizing energy delay metric for performance enhancement of wireless sensor networks," Int. J. Eng. Sci. Technol., vol. 2, no. 5, pp. 1289-1297, 2010.
3. M. Avvenuti, P. Corsini, P. Masci, and A. Vecchio, "Increasing the efficiency of preamble sampling protocols for wireless sensor networks," in Proc. IEEE MCWC, vol. 1. Sep. 2006, pp. 117-122.
4. M. Buettner, G. V. Yee, E. Anderson, and R. Han, "X-MAC: A short preamble MAC protocol for duty-cycled wireless sensor networks,"in Proc. 4th Int. Conf. Embedded Netw. Sensor Syst. ACM SenSys, vol. 4.pp. 307-320, 2006
5. L. Shi and A. O. Fapojuwo, "TDMA Scheduling with Optimized Energy Efficiency and Minimum Delay in Clustered Wireless Sensor Networks," in IEEE Transactions on Mobile Computing, vol. 9, no. 7, pp. 927-940, July 2010

ISSN 2348 – 8034
Impact Factor- 5.070

6. Venkatesh Rajendran, Katia Obraczka, and J. J. Garcia-Luna-Aceves. 2003. Energy-efficient collision-free medium access control for wireless sensor networks. In Proceedings of the 1st international conference on Embedded networked sensor systems (SenSys '03). ACM, New York, NY, USA, 181-192.

7. S. C. Ergen and P. Varaiya, "TDMA scheduling algorithms for wireless sensor networks," J. Wireless Netw., vol. 16, no. 4, pp. 985 997, 2010.

8. Y. Wang and I. Henning, "A Deterministic Distributed TDMA Scheduling Algorithm for Wireless Sensor Networks," 2007 International Conference on Wireless Communications, Networking and Mobile Computing, Shang-hai, 2007, pp. 2759-2762.

9. S. Gandham, M. Dawande and R. Prakash, "Link scheduling in sensor networks: distributed edge coloring revisited," Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies., 2005, pp. 2492-2501 vol. 4

10. C. Wang and K. F. Ssu, "A Distributed Collision-Free Low-Latency Link Scheduling Scheme in Wireless Sensor Networks," 2010 IEEE Wireless Communication and Networking Conference, Sydney, Australia, 2010, pp. 1-6.

11. B. Kang, S. Myoung and H. Choo, "Distributed Degree-Based Link Scheduling for Collision Avoidance in Wireless Sensor Networks," in IEEE Access, vol. 4, pp. 7452-7468, 2016.