Middle East Technical University
Department of Computer Engineering
Wireless Systems, Networks and Cybersecurity (WINS) Laboratory

Recursive Clustering Time Division Multiple Access Scheduling

CENG513 Wireless Communications and Networking
2017-2018 Spring
Term Project Report Group B

Prepared by
Mahnoor Yaqoob , Mehmet Cagri Kose
Student ID: e2331460, e2319648

Computer Engineering
22 May 2018

Abstract

Wireless sensor networks have inherent challenges of designing their communication schedule like absence of point-to-point communication, low power capacity and their ad hoc nature. But wireless sensor networks also have the advantage of having single data direction, towards the common sink, even with various numbers of hops. It is important to take wireless sensor networks advantages and disadvantages to account. Although it is important to keep transmission number at minimum, some overhead of command messages beside data messages is necessary to fine tune the communication. While data link will have single direction towards common sink, command link will be both in child's and parent's direction. We propose recursive clustering TDMA scheduling, which tries to provide dynamic topology discovery for resistance to changes, minimum number of total transmissions for information exchange.

Table of Contents

List of Figures

List of Tables

1 Introduction

A wireless ad hoc network is formed by set of finite number of nodes that are geographically distributed. The network is called ad hoc because it does not depend on pre planned or fixed infrastructure. Wireless ad hoc networks can dynamically self organize into temporary topology to form a network.

Ad hoc networks pose many design challenges; one of the most important design issues is Medium Access Control (MAC), which allows avoiding or resolving collisions. Wireless sensor networks have even more design challenges. One of them is battery life that will be mainly used for wireless transmissions and the other is difficulty of maintenance of numerous nodes placed in various locations. Solving these essential problems will allow cheaper WSN implementation thanks to improved battery life and reduced need of human intervention.

Classic ad hoc network solutions (as an example, between mobile computers) fail to address challenges introduced by wireless sensor networks. They have infinite energy source and they are in fewer numbers, easier for manual human intervention. Wireless sensors however can be placed in hard to reach locations throughout the physical location (for example factories) that maintenance of them is not practical.

In this paper we will target the issue of resource allocation by implementing the most common MAC protocol; time division multiple access (TDMA), by using TDMA we can insure the delay bounds. Using TDMA is also energy efficient as compared to other approaches as it can identify easily when to listen to the channel and when to transmit. In this paper we will provide a node based TDMA scheduling algorithm which can efficiently allocate the resources and keeps track of newly added nodes and inactive links. We were able to create a network model that prevents collisions, thus removing the necessity of various retransmission mechanisms. The network model was also resistant to topology changes in a simple and deterministic manner with small convergence delays.

2 Background and Related Work

2.1 Background

The scheduled MAC protocol in this paper is based on Time Division Multiple Access (TDMA). In TDMA the transmission channel is divided into frames and frames consists of time slots, depending on the topology and in each slot only one node is allowed to transmit. The ***frame length*** is defined by the number time slots with in a frame. Nodes are arranged in clusters. Cluster head is responsible for communication inside that cluster. One of the main goals is to reduce collisions or avoid them this can be achieved by using TDMA. There are two types of collisions that can occur when the nodes are in range of each other, ***primary collision***; occurs when a particular node transmits and starts receiving in the same time slot, ***secondary collision***; occurs when a node starts receiving a transmission which was not intended for it. In our case we have assumed that the channel is collision free. A set of time slots is assigned to each link in order to transmit all the packets to the corresponding receivers while implementing TDMA link scheduling algorithms. Successful reception of transmission depends on multiple parameters such as interference, signal strength etc. There are two interference models in TDMA link scheduling:

Protocol interference model: all nodes are assumed to lie in a planar region. A transmission by a node v_i is successfully received by a node v_j if and only if the intended destination v_j is sufficiently apart from the source of any other simultaneous transmissions.[1]

Physical interference (or path-loss) model: the signal quality perceived by a receiver is measured by the Signal to Interference and Noise Ratio (SINR). More specifically, in this model, the transmission from node v_i is successfully received at node v_j if and only if the received SINR is at least the minimum SINR threshold required by node v_j. [1] Our approach is based on the Protocol interference model.

2.2 Related Work

Previous works on TDMA includes two types of assignments, either node is given the transmission rights or a link is assigned with the transmission rights. In this section we will review the link based scheduling techniques proposed in the literature.

Cheng and Yin in [2] have proposed an edge coloring algorithm based on three criterias; minimum number of slots in a directed graph, considering unicast and broadcast traffic both and avoiding hidden terminal and exposed terminal problem. Their edge coloring algorithm ECDiG guaranteed the solution to above three conditions.

In [3] Ali et al have focused on the problem of adaptation of TDMA link scheduling to dynamic topologies. They have proposed two distributed algorithms to adjust TDMA slots in mobile networks. They have tackled the two major design issues, distributed scheduling of the link and identification of changes during connection in their algorithm. One of the algorithms adjusts the slot assignment in case a new link is added and the other algorithm updates and adjusts the slot assignment in case a link is no longer active or deleted. The results of their simulations showed that the adaptive algorithms function as basis for suite of algorithms for implementing TDMA.

In [4] Djukic and Valaee have proposed a method to find a link schedule with a min-max delay. The delay in TDMA scheduling occurs when packets are forwarded from inbound to outbound link. They used the network conflict graph to find the min-max delay for subsets of paths in the network. They have used Bellman-Ford Algorithm to minimize the delay and proposed a heuristic method to solve the spatial reuse problem of the transmission.

In [5] Behzad and Rubin argued that the traditional graph based scheduling algorithms consider the primary and secondary conflicts, but they do not take into account the effect of interference. This causes the performance of such algorithms to suffer as compared to physical interference models. In their paper they have proposed a Truncated Graph based scheduling algorithm which same as other graph based algorithms except it calculated the optimal number of simultaneous transmissions and does not let number of transmissions to exceed this value. The results of their simulations showed that their algorithm does not guarantee the increase in performance.

In [6] the authors have used an interference based model for STDMA scheduling task. In order to find out best suitability of their model to node or link assignment, they compared best results of both link and node assignments. Their method showed that for high traffic loads, link assignment is better than node assignment due to higher reuse efficiency. And for low traffic loads, node assignment is preferable due to lower average time between transmissions.

The authors in [7] have proposed a column generation method for optimization problem of STDMA scheduling. Their schedule consists of number of time slots, one of the several networks units are allowed to transmit in each slot. Their results showed that link oriented assignment achieve higher spatial reuse as compared to node based assignment.

In broadcast scheduling transmission rights are assigned to the nodes and in link scheduling the transmission rights are assigned to the links therefore link scheduling is suitable in unicast traffic. In [8] Graph-based algorithms for point-to-multipoint broadcast scheduling in Spatial reuse Time Division Multiple Access (STDMA) is proposed and their results showed that their model outperforms the existing algorithms. They developed an algorithm for STDMA multihop wireless ad hoc networks under the physical interference model, namely MaxAverageSINRSchedule.

3 Network Model

The proposed methodology includes a wireless ad hoc network with N static Nodes. In a particular time slot a node can transmit, receive or stay idle. We are assuming that all nodes have same receiving sensitivity, transmission power and thermal noise power; they are homogeneous. In a graph based link scheduling algorithm for any two nodes (v_i, v_j) where v_i is the transmitting node and , we define the signal-to-noise ratio (SNR) as follows:

$$SNR = \frac{P_i G(i,j)}{N_r}$$

P= transmission power of node v_i

N_r = total noise at receiver

$G(i,j)$= the link gain between nodes v_i and v_j

In order for our model to satisfies the properties of protocol interference model we define that for a pair of nodes (vi, vj) form a link (i,j), if the signal-to-noise ratio (SNR) is not less than a communication threshold then the node vi is in communication range of node v_j. Therefore the set of links L is to transmit successfully below conditions must be satisfied: For a set of links, LL, we define the transmitting nodes:

$$V_T(L) = v_i : (i,j) \in L$$

$$L = (i,j) : SNR \geq \gamma_c$$

A schedule S is defined as the sets X_t, for $t = 1, 2...T$, where T is the period of the schedule. The sets X_t contain the links assigned time slot t. Therefore if above equations are satisfied we can say that links in L can transmit successfully. The algorithm we have proposed restricts the nodes from transmitting simultaneously.

Graph based link scheduling algorithm is defined as the above network is modeled by a graph:

$G(V, E_C, E_I)$

Where V = set of nodes in a graph

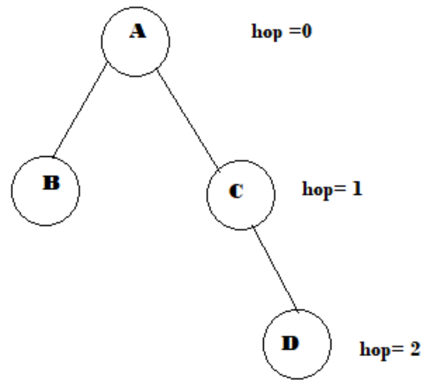


Figure 1 . Topology of network in a graph structure

E_C= set of communication edges

E_I= set of interference edges

Node i is connected to node j with a communication arc if and only if node j is in communication range of node i. Node i is connected to node j with an interference arc if and only if node j is in interference range of node i, but not in the communication range of node i. Most graph based algorithms does not differentiate between E_C and E_I and equal values for them are set. Therefore we have also assumed both values to be equal in our system. Figure 1 shows a graph of scheduled nodes.

4 Proposed Recursive TDMA link scheduling Algorithm

4.1 Node Rights

Every node can transmit and receive data. Depending on the place of hierarchy, a node can have the role of child and/or parent at any given time. But there are limits for this hierarchic structure. A node can have many child nodes but can have only one parent. This allows the topology to be represented in a tree structure.

Parent nodes can have multiple children. Parent nodes keep a list of children nodes. This list is created and maintained with parent requests. Responsibilities of parent nodes are; 1. Sending command messages to their children (Assigning time frames for example) 2. Receiving the data sent by children and passing it to its parent. (Refer to Image 1)

Child nodes can have single parent. They have the id of their parent and this relationship is created by acknowledging parent requests. Responsibilities of child nodes are; 1. Making adjustments according to the commands they received (transmitting only during their assigned time frame) 2. Sending data (own data and data from its children) to their parent during their assigned time frame

4.2 Topology Discovery

The network model will implement wireless sensor network model where there are sensor nodes and there is common sink in the network. All data gathered from sensors will end up in common sink to be processed.

The physical topology of the network is assumed to be nodes randomly distributed over 2D plane. However, on the software side, topology is represented as tree of nodes. Topmost node is common sink node, where all data ends up. Common sink has children and its children have their own children and so on. The number of connections between a node and common sink is defined as hop number. Common sink has the hop number 0 and its children has hop number of 1 and so on.

There are two types of command messages in the system for maintenance and discovery of topology;

1. Parent Request: The topology will be represented as a tree structure. Parent requests are special type of command messages that broadcasts the ID of the node and distance(hop number) of the node to the common sink. When common sink sends the initial parent request, it will be sending it with its ID and hop number equal to 0.
2. Parent Acknowledgement: The nodes other than common sink, at the start of operation, will wait for parent requests. When a node receives parent request, it compares the hop number in the parent request to its current (if exists) parent's hop number. If new parent request comes from a node closer (in terms of hope number) to the common sink, the node accepts new parent parent acknowledgement message, otherwise ignores it. Parent acknowledgement message also has data length field that represents the length of total data this node is responsible for.

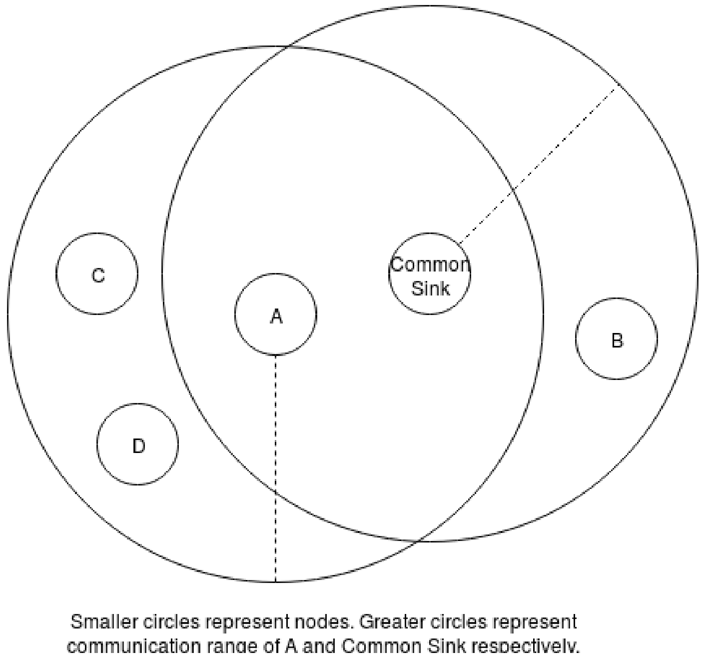
Topology discovery mechanism is applied recursively, starting from the common sink to the farthest nodes. At the very start, common sink starts topology discovery. Once common sink discovers its children nodes, the children nodes in turn start to discover their children. Refer to figure 2a, 2b and 2c for further information.

4.3 TDMA Scheduling

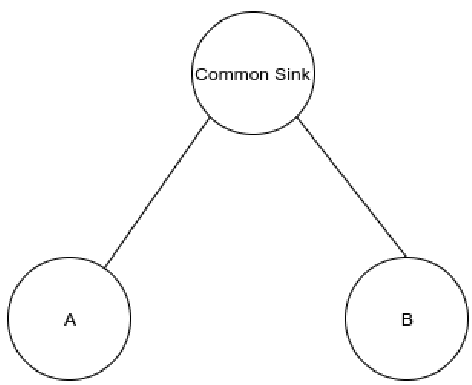
Scheduling algorithm works after topology discovery is done. We mentioned topology discovery starts with common sink at the section before. Lets start with how scheduling algorithm works for common sink.

After common sink discovers its children, it calculates durations of sub-frame it must allocate to each child and and itself. We defined superframe as the largest time frame that is repeated through networks lifetime. At first, whole superframe is allocated to the common sink. The common sink allocates sub-frame from the superframe. The rationale for common sink allocating a sub-frame to itself is simple, common sink is also a parent and it is assumed common sink produces data and relays its childrens data.

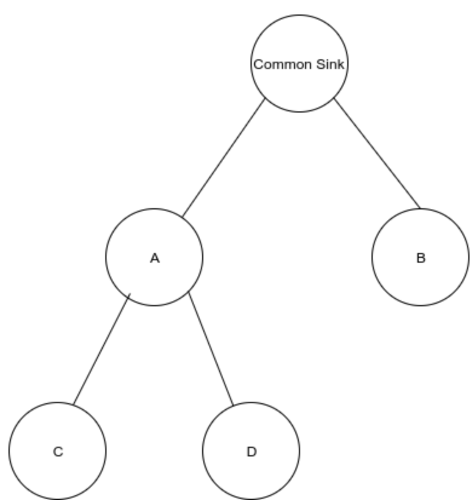
The duration of each subframe allocated to each node is proportional the total data amount that node is responsible for. For this reason, the subframe allocated by common sink to



(a) Physical representation of node placement on 2D plane



(b) Common Sink Discovers the child nodes



(c) Child nodes further discover their children

Figure 2 . Topology Discovery

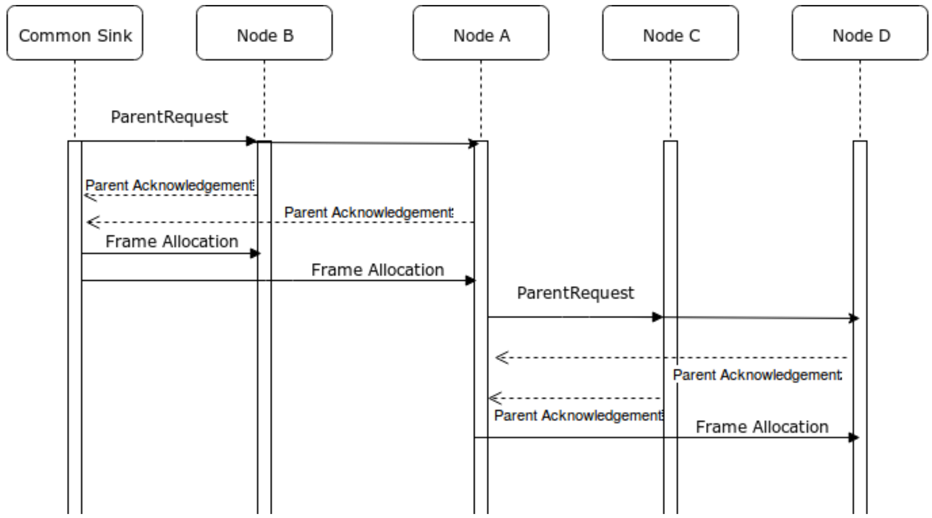


Figure 3 . Sequence Diagram of TDMA scheduling

itself(or any other parent node) is larger than all children nodes separately.

Topology discovery and frame allocation mechanisms work hand in hand. After the completion of scheduling for common sink, topology discovery for child nodes starts.

4.4 Clock Synchronization

We mentioned both topology discovery and scheduling algorithms are applied recursively from the common sink to edge nodes. Clock synchronization is also done in a recursive manner. Once the topology is understood, every parent node is responsible providing clock signals to their children nodes. Clock signals are sent at the start of time frames allocated to the parent nodes. Remember that, only parent nodes can send clock signals. Also, clock signal has id field which help children nodes to accept clock signals from the parent node and ignore clock signals received from non-parent nodes. Children nodes calculated their allocated time frames using offset and frame duration values provided in the frame allocation messages. Offset value is added the time mark which is clock received and frame duration value tells how long they can use the channel.

4.5 Data Transmission

After both initial discovery and scheduling operations are completed, nodes start to transmit data in their allocated time frames. As we mentioned before child nodes transmit only their data and parent nodes are responsible for transmitting their own data and relaying their children's data to the higher level parent nodes. When a node transmits its own data, it uses data transmission frame structure with their own id as source and their parent's id as destination id. When a node relays their children's data, it uses data relay frame structure to reframe the child node's data. The difference between data relay frame structure and data transmission structure is; data relay frame structure has the relaying node's unique id as source id and the unique id of original owner of the data in origin id field. This way, when relaying the data from children nodes, the original creator is not deleted from the records. Figure 5 shows the flow of data transmission.

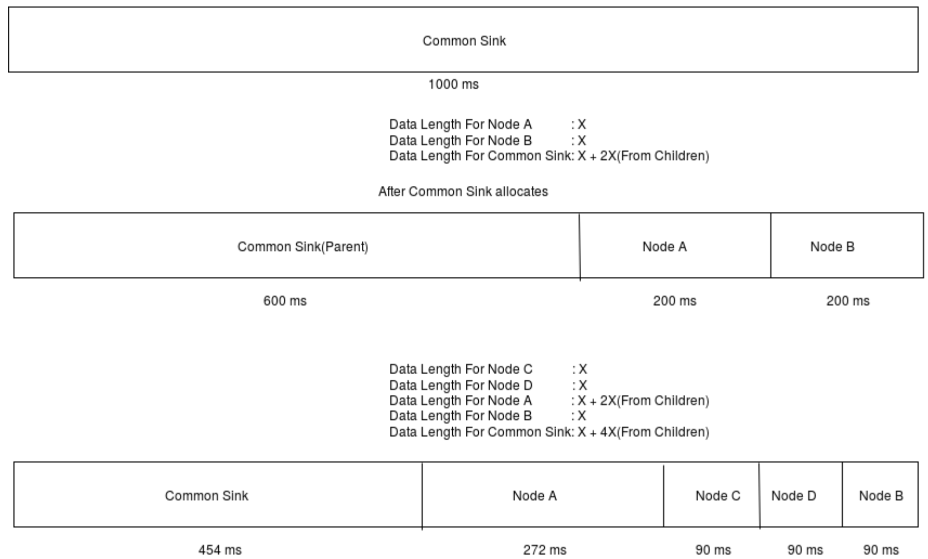


Figure 4 . Frame Allocation Represented in time line

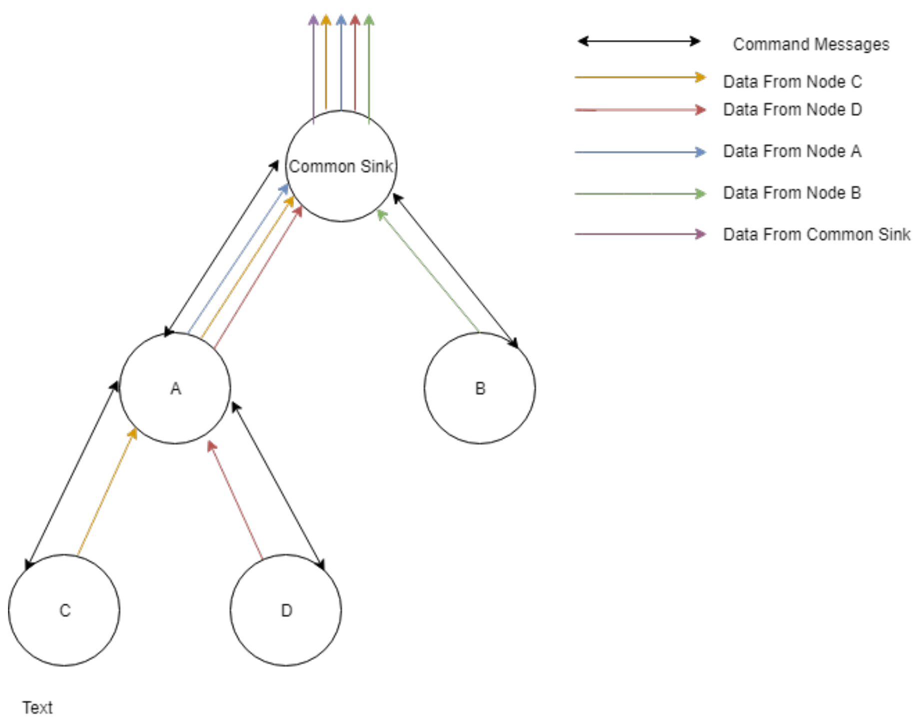


Figure 5 . Data Flow

Parameters	Values
Frequency	433 MHz
Bandwidth	200 KHz
Modulation	Frequency Shift Keying
Data Rate	9600 Kbps
Output Power	20dBm

Table 1. Simulation Parameters

5 Methodology

5.1 Lab Equipment

Because of the failures of BladeRF lab in the department, we have chosen another approach. We have used single Arduino Nano demonstration board as main processing units and single Dorji DRF4432D20 as transceivers as replacement for BladeRF modules. We used total 5 sets to simulate BladeRF lab. Arduino and Dorji units are connected to each other over serial connection. Arduino Nano boards are programmed with Arduino IDE through USB connection. Dorji units are connected only to Arduino units over data and power points. Dorji DRF4432D20 communicates over simple UART connection. Relevant simulation parameters are given in table 1. For debugging, serial connection over USB is used. Arduino IDE can be used to display serial output of Arduino boards. Since the transmission range of Dorji units exceeds the possible distance we can create in our lab, every node could hear each other. To test multi-level topology, we used flags in some nodes to prevent them to be child of common sink directly

5.2 Frame Structure

Every frame starts with Start byte *$* and ends with an end byte #, to distinguish each frame from each other. Different frame structures for command and data messages are given below. X's represent each bit in the figure 6a,b,c,d,e and f.

Parent Request

- Type Code:Command type code for parent request (0xF7)
- Own ID: This field has unique id of the node
- Hop number: This field has distance to the common sink. It is 0 for common sink itself.

Parent Acknowledgement

- Type Code: Command type code for parent acknowledgement (0xAC)
- Own ID: This field has unique id of the node
- Parent ID: This field has unique id of new accepted parent.
- Data Length: This field has total data length that will come from this node. Every node transfers its own data and its children's data to its parent. This field is used to allocate necessary time frame to children.

Frame Allocation:

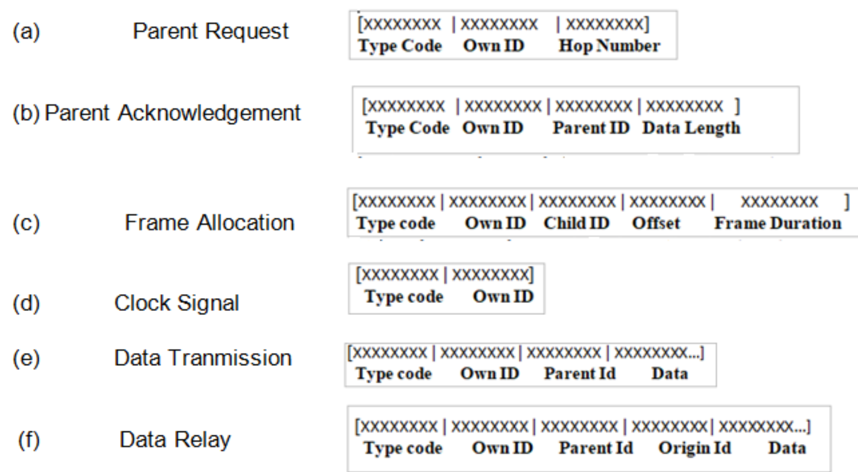


Figure 6 . Different Frame Structures

- Type Code: This field has the command type code for frame allocation (0xFA)
- Own ID: This field has unique id of the node
- Parent ID: This field has unique id of child node.
- Offset: This value sets the start of child's time frame, starting point is clock signal
- Frame Duration: This value sets the duration of frame allocated to child for channel use.

Clock Signal

- Type Code: This field has the command type code for clock signal (0xC5)
- Own ID: This field has unique id of the node. This is used to prevent children nodes from synchronizing with another parent node even if they receive clock from them.

Data Transmission

This frame structure is used to transmit data from child node to parent node. This message is used to transmit data created in this node.

- Type Code: This field has the command type code for data transmission (0xDA)
- Own ID: This field has unique id of the node as source address.
- Parent ID: This field has unique id of parent node as destination address.
- Data: This field carries the data from the current node.

Data Relay

The parent node, after receiving data from its children nodes, relays the data of children nodes to its own parent. To distinguish data retransmission of childrens data from the parents own data, childrens data is reframed with this frame structure.

- Type Code: This field has the command type code for data relay (0xD7)
- Own ID: This field has unique id of the node as source address.
- Parent ID: This field has unique id of parent node as destination address.
- Origin ID: This field has unique id of the origin node which created data.
- Data: This field carries the data from the child node.

Data Length	Latency
9	84
13	94
15	98
20	119
25	131
30	154

Table 2. Latency Values with respect to different data size

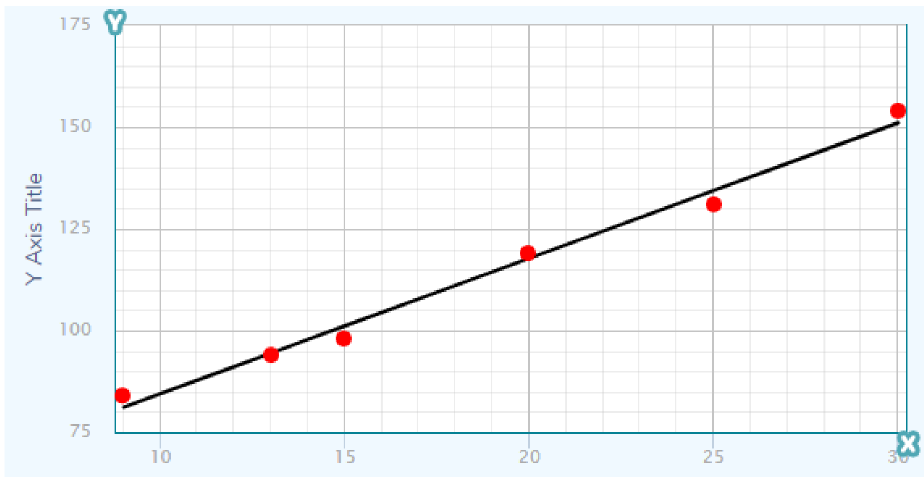


Figure 7 . Inherent Latency and Transmission Latency plotting

6 Results and Conclusion

6.1 Results

Since wireless network nodes have low processing power, we tried to measure inherent latency that is with the smallest transmission and transmission latency which changes with respect to data length and transmission functions between Dorji modules. According to the measurements and latency values given in the table 2, we have applied curve fitting technique to find inherent latency and transmission latency.

Once we apply curve fitting to the plot created by the values shown in figure 7, we acquired the following equation

$$Y = 3.328664 * X + 51.19828$$

As we can see, inherent lag is 51 milliseconds. Transmission delay is 3.3 milliseconds for every byte we sent. Inherent lag can be solved by subtracting inherent latency value from the time frame offset values sent to each node.Transmission delay is also explainable. Dorji transceiver module has a UART interface that works at 9600 baudrate which is used to transfer data from Arduino modules. The transmission rate of the Dorji module was also 9600 baudrate during our trials. 9600 baudrate can be converted into 960 byte per second since there is a start bit and stop bit before and after the byte we send(total 10 bits is used to send 8 bits). So, for each byte we can expect latency close to 1 milliseconds. Since there are two 9600 baudrate interfaces, total latency injected by them is close to 2 seconds. The remaining 1.3 milliseconds latency for each byte probably comes from the low processing power of both Dorji and Arduino modules.

Since we can consistently predict the latency for known size of data, nodes can apply an algorithm to decide when to call their transmission functions. Consider the example below; A node has 20 bytes of data to send within 50 milliseconds time frame. By using equation, he calculates the expected latency as 3.32*20 + 51 = 117 milliseconds. The actual transmission rate of(not UART interface between Arduino and Dorji) transceiver is 9600 baudrate. So the transmission of 20 byte will occupy the channel approximately 20 milliseconds. Therefore node can start calling transmission functions anywhere between 117 milliseconds and 87 milliseconds before the time frame starts. With these adjustments, transmitting nodes use the channel within their allocated time frame without infringing on other timeframes.

6.2 Future Work

In future, we will develop compensating mechanisms that takes into account the latency caused, not only by transmission rate but also various hardware interfaces, limitations and software implementations. Time synchronization problem hinders the performance of networks which has deep multi-level topologies.

To diminish effects of the latency on the real-time quality for multi-hop networks, effects of shifting time frames back in time will be analyzed

6.3 Conclusion

In this paper we have proposed an algorithm for TDMA link scheduling that ensures lightweight frame structure This results in reduced transmission time and increased battery life. Designed protocol is lightweight and there are no excessive acknowledgement mechanisms. However the proposed model does not reuse spectrum amongst non-interfering links.

Static time frames allow nodes to use sleep mode more efficiently because there is no need for channel sensing. This also has positive effect on battery life.

References

[1] A. Sgora, D. J. Vergados, and D. D. Vergados, ``A survey of tdma scheduling schemes in wireless multihop networks,'' *ACM Computing Surveys (CSUR)*, vol. 47, no. 3, p. 53, 2015.

[2] M. Cheng and L. Yin, ``Transmission scheduling in sensor networks via directed edge coloring,'' in *Communications, 2007. ICC'07. IEEE International Conference on*. IEEE, 2007, pp. 3710--3715.

[3] F. N. Ali, P. K. Appani, J. L. Hammond, V. V. Mehta, D. Noneaker, and H. Russell, ``Distributed and adaptive tdma algorithms for multiple-hop mobile networks,'' in *MILCOM 2002. Proceedings*, vol. 1. IEEE, 2002, pp. 546--551.

[4] P. Djukic and S. Valaee, ``Link scheduling for minimum delay in spatial re-use tdma,'' in *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*. IEEE, 2007, pp. 28--36.

[5] A. Behzad and I. Rubin, ``On the performance of graph-based scheduling algorithms for packet radio networks,'' in *Global Telecommunications Conference, 2003. GLOBECOM'03. IEEE*, vol. 6. IEEE, 2003, pp. 3432--3436.

[6] J. Grönkvist, ``Novel assignment strategies for spatial reuse tdma in wireless ad hoc networks,'' *Wireless Networks*, vol. 12, no. 2, pp. 255--265, 2006.

[7] P. Björklund, P. Värbrand, and D. Yuan, ``A column generation method for spatial tdma scheduling in ad hoc networks,'' *Ad hoc networks*, vol. 2, no. 4, pp. 405--418, 2004.

[8] K. Papadaki and V. Friderikos, ``Robust scheduling in spatial reuse tdma wireless networks,'' *IEEE Transactions on Wireless Communications*, vol. 7, no. 12, pp. 4767--4771, 2008.