OIdTDMA: Order Identification Time Division Multiple Access Switching and Routing for Wireless Sensor Networks

Maad Issa AL-TAMEEMI
Department of Computer Engineering
College of Engineering-University of Baghdad
Baghdad, Iraq
maadesaa@yahoo.com

***Abstract*—one of the major problems that cause a collision in WSN is Hidden and Exposed terminal problems.**
This work will concentrate on Time Division Multiple Access protocol (TDMA). TDMA is a very important protocol for WSN by minimizing the collision, avoiding idle listening by allocating time slots, and guarantee reliable communication. However, it couldn't resolve the hidden and exposed terminal problems, which occur when there is heavy communication in the network. We have tried to solve these problems with a new TDMA named OIdTDMA. In this protocol, the allocation of time slots determined by considering the identification order of each sensor node in the network. The results show that we have obtained a reduction of packet collision between sensors and thus, we avoid the hidden and exposed terminal problems. In addition, the use of the joint between this protocol which functions the MAC sublayer of the Data Link Layer and one of the Routing protocols which represent Network Layer, These layers exist in Wireless Sensor Networks.

Keywords- TDMA, hidden terminal, exposed terminal, OIdTDMA, network, sensor, base station.

1. INTRODUCTION

Wireless Sensor Network (WSN) is a group of sensors communicating via wireless networks. The power is supplied to these sensors through one battery for each sensor. WSN can be composed of hundreds or thousands of sensors. The goal of these kinds of networks is to collect useful information from their environment, compute simple tasks, and communicate data in a multi-hop manner in order to achieve some common objective such as managing battlefields, preventing forest fires, and monitoring seismic activity. The problems that can occur in WSN and can be resolved by minimizing energy consumption, increasing the lifetime or reducing the delay [1][2].
Generally, WSN is characterized by the fact that all nodes share the same transmission channel. Therefore, distributed Medium Access Control (MAC) mechanism is a key component of WSNs and it has obtained intensive research attention [6]. Indeed, the MAC protocol plays a very important role in handling errors and deciding when nodes could access the shared medium in order to transmit their data and trying to ensure that no collisions happen.
The MAC protocols can be classified into two groups: Contention-Based and Contention-Free protocols [11]. In contention-based protocols, all the sensor nodes access the channel in competition, so collisions can occur when more than two nodes access the channel at the same time (for example MACA [9], S-MAC [19]). In contention-free protocols, each sensor node has its own time interval or its own transmission frequency (for example TDMA [12], FDMA [15]). Nodes access the channel without competition, and there is no collision within the direct neighborhood.

One important approach in MAC protocols is to let nodes synchronize their active/sleep periods such that neighboring nodes are awake at the same time. A receiver only listens to a brief contention period at the beginning of the active phase, while senders contend during this period. Only nodes participating in data transfer remain awake after the contention period while others go back to sleep until the next active period.

One of the major problems that cause a collision in WSN is: Hidden and Exposed terminal problems.
This work is organized as follows. Section 2 Presents the WSN. Section 3 discusses the existing protocols used in the MAC sublayer and network layer. Section 4 explains the proposed OIdTDMA protocol and gives an analysis of different examples. The performance of protocol is evaluated in Section 5. Finally, Section 6 gives conclusion and suggestion for future work.

2. WSN PRESENTATION

Today, wireless Sensor Networks (WSN) is experiencing a very strong development. A WSN is a network consisting of sensors that can communicate with each other without any physical link.

A sensor is a small device that allows data acquisition, data processing, sending, and communicating with other devices. A sensor must satisfy certain constraints such as memory space, and energy consumption.
In this section, we present Wireless Sensor Network in general. Then, we describe the architecture and protocol layers of WSN.

2.1 Wireless Sensor Network(WSN)

A Wireless Sensor Network consists of hundreds or more sensors nodes which can either have a fixed location or are randomly deployed to monitor the environment [1]. Each sensor is capable of doing autonomously three complementary tasks: get the data and process it then communicate through the radio. In general, the flow of data is "multiple sources - single destination". The sensor nodes are numbered by an integer and the base station is identified by BS, where the base station may communicate with the task manager node via Internet or Satellite as shown in Figure 1.

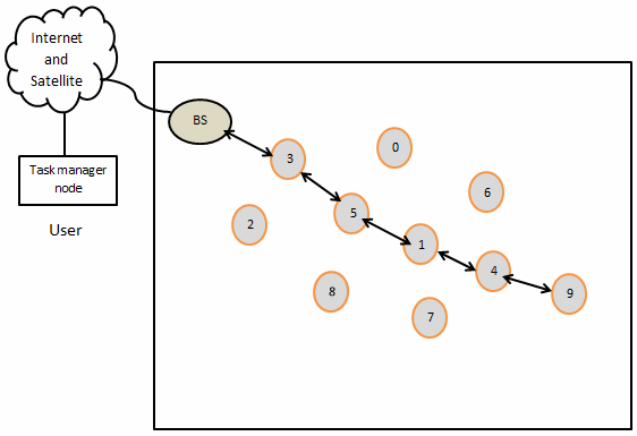


Figure 1: Wireless Sensor Network

The design of the sensor network influenced by many factors such as sensor network topology, hardware constraints, transmission media, and power consumption.
Wireless sensor networks are used in several domains such as military, medicine, and industry. The most common problems in the world of wireless sensor networks are lifetime, energy consumption, and the communication time delay between the sensor nodes and the base station in the network.

2.2 Protocol Architecture

In order to effectively establish a WSN, a layered architecture is adopted to improve the robustness of the network. In general, a protocol stack consists of five layers.
This protocol stack includes the application layer, the transport layer, the network layer, the data link layer and the physical layer as shown in Figure 2.

In addition, this stack has three planes (levels) of management: the task management plan which balances and schedules the sensing tasks given to a specific region, the mobility management plane which detects and registers the movement of sensor nodes in the routing phase and the sensor nodes can keep track of their neighbor sensor nodes are, the sensor nodes can balance their power and task usage, and the energy management plane which manages the use of energy for a sensor node. These management planes are necessary for the sensor nodes to work together in a power efficient way, to route data in a mobile sensor network, and to share resources between sensor nodes.

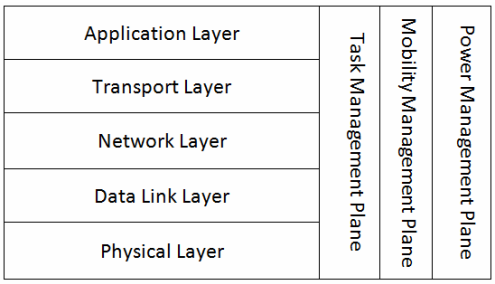


Figure 2: WSN Layers architecture

2.3 WSN Layers

The protocol stack of sensor nodes in a wireless sensor network consists of five layers as shown in Figure 2 and explained below:

- Applications Layer: depends on the sensing tasks that are different types of application software.
- Transport Layer: helps to maintain the flow of data if the application for wireless sensor networks requires it.
- Network Layer: takes care of routing the data supplied by the transport layer. This layer takes account of several challenges such as power saving, node deployment, scalability and the fact that sensors have not a global id and that sensors have to be self-organized.
  This layer aims to define a reliable path according to a certain unit of measure called metric which differs from protocol to protocol. There are several routing protocols applied for this layer. These protocols can be divided into several types such as flat routing (for example direct diffusion [8]), hierarchical routing (for example LEACH [7]), Location-based routing (for example Greedy Perimeter Stateless Routing [10]) and QoS-aware routing (for example Sequential Assignment Routing [14]).
- Data Link Layer: responsible for equitable distribution of resources and providing the multiplexing of the data stream, frame detection and error control. There is an important functionality supported by this layer which is the MAC sublayer. The MAC sublayer is responsible for handling errors which must be power aware and able to minimize collisions with neighbors broadcast. There are many protocols supported in this layer. These protocols can be classified into several types such as contention-based protocols (for example S-MAC [19], T-MAC [18] , D-MAC [13] , MACA [9]) and Contention-free Protocols (for example TDMA [12], FDMA [15], CDMA [20]).
- Physical Layer: addresses the needs of simple but robust modulation, transmission and receiving techniques.

3. EXISTING PROTOCOLS IN THE MAC SUBLAYER AND NETWORK LAYER

In WSN, minimizing delay and saving energy are the most important issues, by which we can increase the lifetime of the network. In this kind of networks, the MAC layer plays a very important role in handling errors and minimizing collisions with neighbors. So, in this section, we start to analysis the context, problems and the existing solutions for some of the protocols in this layer. Furthermore, the network layer also plays a very important role to find the reliable path between the source and the destination. Thus, in this work, we analyze and implement two types of protocols: Dijkstra and geographic routing.

3.1 The MAC Protocols

The main goal of MAC protocols for WSN is to minimize collisions in order to optimize energy consumption and to reduce the end-to-end time delay to prolong the lifetime of sensors.

Medium access in wireless networks is difficult because the transmission and the reception of data are very difficult at the same time.

Some problems may occur in the MAC sublayer that causes collision such as Hidden and Exposed terminal problems, as shown in [22], which increases the delay and then energy consumption. These situations occur when:

- Hidden terminal illustrated in Figure 3: Types of collisions in multi-hop wireless networks occurs when either the sensor node X is receiving a packet from more one node as shown in collision type 1, or for a node receiving and transferring a packet at the same time as shown in collision type 2.

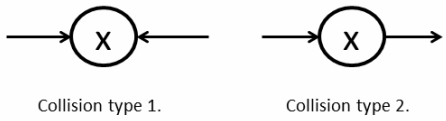


Figure 3: Types of collisions in multi-hop wireless networks

- A node is receiving a packet destined for another node (overhearing).
- A node is trying to listen even though there is no sensor node sending a packet (idle listening).
- The destination node is not ready to receive a packet through the transmission procedure and hence this packet is not correctly received (over emitting) [16].
- The exposed terminal problem occurs when the sensor node Y is transmitting to the sensor X, and sensor Z has a packet intended for the sensor K. Because the sensor Z is in the range of Y, it senses the channel to be busy and is not able to send as shown in Figure 4.

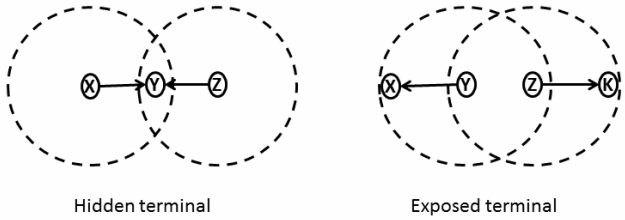


Figure 4: The hidden node problem and the exposed node problem

There are several protocols applied in the MAC sublayer which is trying to avoid all or some of these problems. Among these protocols, this work will concentrate on time division multiple accesses (TDMA) [12]. This protocol divides the bandwidth between all the nodes by dividing each logical channel into time slots to avoid collisions. This algorithm is efficient for small networks. Several approaches are trying to develop the TDMA protocol to get an optimal solution for large or small networks. These approaches operate individually in the MAC layer or working with another protocol in another layer to get the ideal solution for more than one layer at the same time. Some of them are briefly presented below:

- The authors submit the idea that TDMA is the desirable algorithm in WSNs for saving energy because it allows a sensor node to minimize idle listening. Moreover, TDMA has verified to be valid in converting existing distributed algorithms into a model which is consistent with WSNs. They show how TDMA can be used to provide efficient energy. The results show that the sensor nodes can save energy when the network is idle, and switch to active mode whenever the network detects an event. This work proposes a self-stabilizing, deterministic algorithm for TDMA in WSNs where a sensor node is aware only of its neighbors [3][4].
- The authors present SS-TDMA and explain how the time slots are assigned to the sensors so that the delay in the broadcast is reduced. Additionally, they proved that this algorithm is efficient for the end-to-end delay and showed how to use the SS-TDMA in the context of power management [3][4].
- The author proposes a reservation scheme in a dual frequency radio, which is Latency Minimized Energy Efficient MAC protocol (LEEM) to minimize the latency in the multi-hop path data transmission through a prior reservation of the next hop channel. Therefore, in a multi-hop sensor network, a packet can be forwarded to the next hop, as soon as it is received by a sensor node, which helps to eliminate the delay incurred for setting up the path [5].

MAC sublayer protocols can be classified into two groups, which are the contention-based and contention-free protocols [11]. We analyze these groups in the next section.

3.1.1 Contention-Based

In contention-based protocols, all the sensor nodes access the channel in competition, so collisions can occur when more than two nodes access the channel at the same time (for example S-MAC, MACA).

S-MAC protocol

The authors proposed S-MAC (Sensor-MAC) [19], S-MAC consists in locally managed synchronizations. The main idea of S-MAC is to divide the time of transceiver operation into two parts: active and sleep which is based on these synchronizations. When the transceiver is active, the sensor is ready to transmit or receive data, whereas, when the transceiver is in sleeping state, the sensor does not transmit or receive data.

Since the purpose of S-MAC is to reduce energy consumption, they tried to extend the sleeping time slot to reduce the energy consumption of sensors. However, when sensors are in the sleeping and active state periodically, a problem named deaf listening can occur. When sensors are working independently, a sensor can be active while its neighbors are in the sleeping state. If the active sensor transmits data frames to a sleeping sensor, this will raise a problem because the sensor in a sleeping state cannot receive data.

The solution of S-MAC is that the nodes must synchronize themselves to switch to an active state or sleeping state at the same time. Furthermore, RTS/CTS packet exchanges are used to unicast-type data packets as shown in Figure 5. Once these control packets are exchanged, the transmitter and receiver to remain in the active state during this period. Nodes that have no data to transmit or receive will turn off until the end of the period.

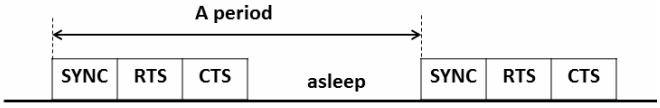


Figure 5: The period active and asleep in the S-MAC

This method allows sensors to avoid the non-active listening problem, but it will increase latency as nodes cannot transmit at the beginning of each period.
After receiving the data, the receiver cannot forward data frames immediately as the next node is already in the sleep state. It must wait until the start of the next period to access the channel. This protocol can reduce wasted energy by minimizing idle listening by making sleep schedules. A major drawback of S-MAC is that sleep and listen periods are predefined and constant, which decreases the efficiency of the algorithm under the variable traffic load.

MACA – Multiple Access Collision Avoidance

The basic idea of the MACA [9] is to use additional signaling packets which are:

- Request to send (RTS): the sender asks the receiver if it is able to receive a packet.
- Clear to send (CTS): the receiver agrees and sends out a clear to send(CTS).

Generally, Sender asks Receiver if it is able to receive a transmission Request to Send (RTS), Receiver agrees, he will send out a Clear to Send (CTS), the Sender sends the data and the Receiver then informs the Sender by sending the acknowledgment (ACK). The potential interference overhears RTC/CTS is by carrying the expected duration of the data transmission as shown in Figure 6.

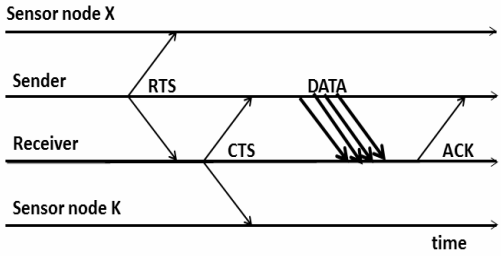


Figure 6: The request and clear to send in the MACA protocol

This protocol could not solve the hidden and exposed terminal problems because there are some collisions between sensor nodes as shown in
Figure 7.

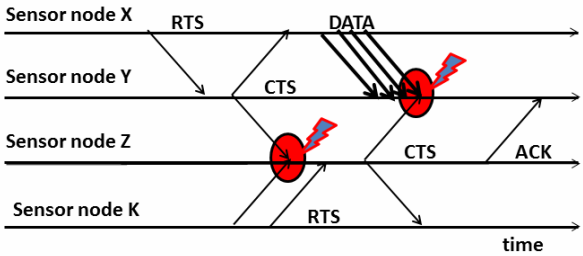


Figure 7: Collision case in the MACA protocol

If a packet is lost (collision), the node uses the binary exponential back-off (BEB) algorithm to back off for a random time interval before retrying. Each time a collision is detected, the node doubles its maximum back-off window.

3.1.2 *Contention-Free*

In contention-free protocols, each sensor node has its own time interval or its own transmission frequency (for example, TDMA, FDMA). Nodes access the channel without competition and there is no collision within the direct neighborhood.

TDMA protocol

Time Division Multiple Access (TDMA) is an important protocol for WSN. It runs the main role in saving energy by minimizing the collisions, avoid idle listening by allocating time slots, and guaranteeing reliable communication. In this protocol, each node has a time interval to access the channel as shown in Figure 8.

Figure 8: The TDMA Frame

A sensor node cannot access the channel during the interval reserved for another node. Hence, there is no collision. However, this protocol needs, at least, one coordinator node to allocate time slots to others. To operate, TDMA needs a good level of synchronization between nodes (a few microseconds). in other words, when the network topology changes, time slots must also change accordingly. In addition, the TDMA protocol cannot detect the two-hop neighbors: collisions, and cannot resolve the exposed terminal problem which occurs when a node is wanted to send packets to other nodes due to a neighboring transmitter.

FDMA protocol

In FDMA (Frequency Division Multiple Access) protocols, each node possesses several different frequency bands. A node knows the frequency of its neighbors. When it wants to transmit data frames to one of its neighbors, it chooses the neighbors frequency to transmit. Nodes can simultaneously make multiple transmissions within the scope of others without interfering.

We can imagine this type of MAC protocol as traffic on a highway with multiple channels: each channel represents a frequency band. The data frames can move on a path without disturbing others. FDMA has a large advantage because it can operate without interference and with high throughput. In fact, each transmission is performed on a different frequency band. This technique is used in the field of the mobile phone where the terminals size is quite large. This protocol requires a complex design of the transceiver with additional circuits. Therefore, FDMA is not the good candidate for sensor networks MAC protocol. This explains why there is not much work on this kind for sensor networks.

The author proposed a hybrid protocol between TDMA and FDMA [15]. The authors have shown that the use of pure TDMA or FDMA consumes more energy than a hybrid protocol that combines these two approaches. They have also proposed a method for determining the optimal number of FDMA channels to minimize the level of energy consumption. However, the main problem in FDMA, the complexity of the design of the transceiver, is still unresolved.

3.2 The Networks Protocols

There are many protocols applied in the network layer to find a reliable path between the nodes and the base station. In general, these protocols are designed to find the unfailing path to save energy and to minimize delay. This work will concentrate on two special types of protocols: Dijkstra algorithm to find the shortest path and Geographic Routing to discover the path between a sensor node and the base station.

3.2.1 Dijkstra Algorithm

Dijkstra algorithm is a graph search algorithm. This algorithm uses to find the shortest path problem for a graph with non-negative edge path costs producing the shortest path tree [17]. But also, it uses to find the shortest path for a wireless sensor network in term of the energy and reliability place. This algorithm calculates the least expensive path (i.e., the shortest path with respect to a cumulative cost metric) between the source and the target. Dijkstra algorithm can be used to find the route with the highest probability of success. Because Dijkstra algorithm is simple to implement, has a reduced complexity, and is proven to be extremely stable, it is widely used in network routing protocol and also in wireless sensor network.

Generally, in WSN, we need to use an energy efficient routing algorithm to calculate the shortest path from a sensor to the base station.

For this, the Dijkstra algorithm can be applied in a wireless sensor network to determine the shortest path between a pair of nodes i and j.

Where each node in the Dijkstra algorithm must respect the following rules.

- Each node can determine its own position.
- Each node knows the position of its neighbors.
- The position of the destination is known.

We analyze an example for this algorithm, where we suppose that all the links in this example are weighted. Each weight between two nodes represents the distance between them. This distance can be proposed as the time for sending one packet from one to another, or can be represented the euclidean distance between two nodes. Let the starting node be the sensor node 1 as shown in Figure 9. Dijkstra's algorithm will assign some initial distance values and will try to improve them step by step.

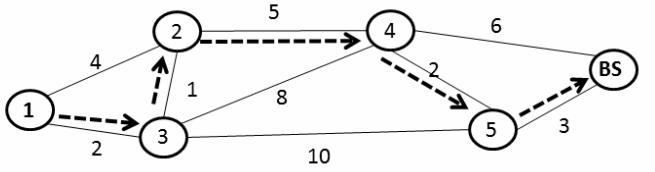


Figure 9: Example of Dijkstra Algorithm to find the shortest path

In Figure 9, Dijkstra Algorithm calculates the shortest path by depending on the distance of the source which is the sensor node 1 to the destination which is the base station As shown in the steps below.

1. For every node, we assign a tentative distance value: we set it to zero for our sensor node~1 and to infinity for all other nodes.
2. We mark all nodes unvisited. Set the sensor node 1 as current. We create a set of unvisited nodes called the unvisited set which consisting of all the nodes.
3. For the current node, we consider all of its unvisited neighbors and calculate their tentative distances. We compare the newly calculated tentative distance to the currently assigned value and assign the smaller one.
4. When we are considered all of the neighbors of the current node, we mark the current node as visited and remove it from the unvisited set. A visited sensor node will never be checked again.

5. If the base station has been marked visited or if the smallest tentative distance among the nodes in the unvisited set is infinity so the algorithm has finished.
6. We select the unvisited sensor node that is marked with the smallest tentative distance and set it as the new "current node" then go back to the third step.

4. The Proposed Protocol

In this section, we present in details the proposed OIdTDMA protocol and the basic conception of this protocol. We explain the algorithms created in this protocol and write its pseudo-code. In addition, we consider a case study example of a network to evaluate the efficiency of this algorithm in sublayer MAC. In addition, we will try to make joint between this protocol and some of exists Routing Protocols in Network Layer.

4. 1 The OIdTDMA Protocol

4.1.1 Basic conception

In this work, we study the OIdTDMA scheduling which aims to use an MAC scheduling with a routing algorithm in order to evaluate some metrics like delay, hops count, energy, and to check the quality of the routing solution. The OIdTDMA can avoid all collisions within the scope of our carried work, and in particular, the two-hops neighbor collision which leads to reducing the energy consumption. This protocol can minimize the end-to-end time delay.

We assume to be in a centralized context in which the base station receives all connectivity information for all sensor nodes in the network and uses existing routing algorithms, such as the Dijkstra algorithm and the Geographic routing protocol to calculate the paths from every sensor node to the base station. Therefore, this OIdTDMA protocol is responsible for managing the state of each sensor node either it is active or idle to have a successful packet circulation between the sensors in the network.

4.1.2 The notations for used algorithms in this protocol

In this section, we describe the notations of the algorithms used in this work as shown below.

- SN(V, E, bs): the sensor network with V the set of nodes (the sensors and base station), E the set of edges and bs the base station.
- M: the slots matrix of the sensor nodes in the network, the number of rows depends on the number of sensors, the number of columns depends on the maximum degree of a network.
- index(s): returns the column's index where s in placed in M.
- addSlot(M): adds an empty slot at the end of each line of M.
- Γ (s): predecessors successors of a sensor s.
- Δ (SN): the maximum degree of a network SN.

4.1.3 The Proposed Algorithm

In general, TDMA scheduling algorithm can be defined as the process of allocating time slots to the sensor nodes. Each node has a time interval to access the channel. TDMA ensure collision-free channel access for a sensor node with its neighbors only. The data transmission from a sensor node to the base station is received without collision by all its one-hop neighbors only. However, this algorithm can't avoid the hidden and exposed terminal problems.

The main goal of this work is to create a new TDMA scheduling named OIdTDMA, which can avoid the hidden and exposed terminal problems, avoid all collisions within the scope of our carried work.

In this work, the allocation of time slots is determined by considering the identification order of each sensor node in the network. We represent the time slots for all the sensor nodes as the matrix named M. The number of rows of M represents the total number of sensor nodes and the number of columns represents the maximum degree of a sensor in a network plus an extra column for this sensor as shown in Figure 10.

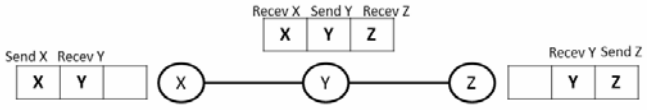


Figure 10

In Figure 10 the sensor X sends x data to the neighbor which is the sensor Y and the sensor Y receive x data, The sensor Y sends also y data to the neighbors which are X and Z and they receive y data, The sensor Z sends z data to the neighbor which is Y and it receive z data.

We have shown in the figure above, the sensor node Y has the maximum degree which equals two plus an extra slot for Y, which represents the number of columns of M and we show that there are three sensors, which represent the number of rows of M as shown in Figure 11.

The Matrix M			
Id. Sensor	The time slots		
X	x	y	null
Y	x	y	z
Z	null	y	z

Figure 11: A matrix M for the figure above

To achieve this goal, we indicate our algorithm of this work named algorithm (1). The objective of this algorithm is to find a suitable index for each sensor s of the network in order to add this sensor to the matrix. This column index is found by the algorithm (2).

Algorithm 1: Algorithm to place the sensors in the M

Data: SensorNetwork $SN(V, E, bs)$
Result: The slots matrix M
create M as an empty matrix of size $(|V| - 1) \times (\Delta(SN) + 1)$
foreach $s \in V - \{bs\}$ **do**
  $j \leftarrow getColumn(s, M, SN)$ `// algorithm 2`
  foreach $i \in \Gamma(s) \cup \{s\}$ **do**
    $M[i][j] \leftarrow s$
  end
end
return M

The algorithm (2) avoids the collisions by using constraints when choosing the column index of a sensor. A sensor s can be put in a column of index j if for each sensor s' which is in the column j, s is not a neighbor of s' and s doesn't have any common neighbor with s'. These conditions prevent collisions; indeed, for example if we have three sensors x, y, and z. y is the neighbor of x and z, the constraints prevent to put x and z in the same column in M because they could be active at the same time and cause a collision.
If there is no free slot or if constraints are not satisfied in all free slots, we have to add a column at the end of M to place the sensor.

Algorithm 2: Algorithm to find a slot index for a sensor in M

Data: Sensor i, Matrix M, Network SN
Result: The index to place the sensor s in M
for $j \leftarrow 0$ **to** $|M[i]| - 1$ **do**
  if $M[i][j] = null \wedge \forall i' \neq i, index(i') = j \Rightarrow (i' \notin \Gamma(i) \wedge \Gamma(i) \cap \Gamma(i') = \emptyset)$ **then**
    return j
  end
end
$addSlot(M)$
return $|M[i]| - 1$

4.1.4 An example

In this subsection, we demonstrate an example of our OIdTDMA algorithm explained in the previous sections. Consider the network in Figure 12, which consist of 10 wireless sensor nodes and the base station.

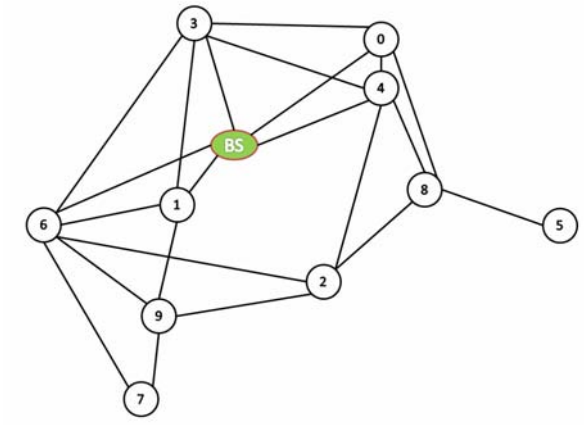


Figure 12: The sample network

The first step of the algorithm is to initialize data, as shown in Figure 13. We can see that M has 10 rows because there are 10 sensors in the network without counting the base station as shown in Figure 12, and 6 columns where the maximum number of columns represent by the maximum degree of a sensor node which is the sensor node 6 in this network and adding an extra column for this sensor. The algorithm starts by demonstrating 10 sensors in order based on Figure 13 as follows:

The Matrix M						
Id. Sensor	The time slots					
0	null	null	null	null	null	null
1	null	null	null	null	null	null
2	null	null	null	null	null	null
3	null	null	null	null	null	null
4	null	null	null	null	null	null
5	null	null	null	null	null	null
6	null	null	null	null	null	null
7	null	null	null	null	null	null
8	null	null	null	null	null	null
9	null	null	null	null	null	null

The Vector Index	
Id. Sensor	The Index

Figure 13: Initial values of M and index

Sensor 0 processed by calling the algorithm getColumn. As M is empty, the first empty slot is in the first column (index 0). The line of each sensor that has the sensor 0 as the neighbor is then updated as shown in Figure 14.
.

The Matrix M						
Id. Sensor	The time slots					
0	0	null	null	null	null	null
1	null	null	null	null	null	null
2	null	null	null	null	null	null
3	0	null	null	null	null	null
4	0	null	null	null	null	null
5	null	null	null	null	null	null
6	null	null	null	null	null	null
7	null	null	null	null	null	null
8	0	null	null	null	null	null
9	null	null	null	null	null	null

The Vector Index	
Id. Sensor	The Index
0	0

Figure 14: The sensor 0 is processed

The next processed sensor is sensor 1. The algorithm tries to place it in the first column of the second line because the cell is free. However, to do that, the constraints have to be satisfied. It is not the case because sensor 1 shares neighbors 3, 6, 9, and BS. But it can reach sensor 0 through sensors 3 and BS. The second column is then chosen. M and index are updated as shown in Figure 15.

The Matrix M						
Id. Sensor	The time slots					
0	0	null	null	null	null	null
1	null	1	null	null	null	null
2	null	null	null	null	null	null
3	0	1	null	null	null	null
4	0	null	null	null	null	null
5	null	null	null	null	null	null
6	null	1	null	null	null	null
7	null	null	null	null	null	null
8	0	null	null	null	null	null
9	null	1	null	null	null	null

The Vector Index	
Id. Sensor	The Index
0	0
1	1

Figure 15: The sensor 1 is processed

Sensor 2 is then processed. This sensor can't be placed in the first column because it shares neighbors 4, 8, 9 and 6 with sensor 0. But it can reach sensor 0 through sensors 4 and 8. It shares the same problem with sensor1 by sharing together sensors 6 and 9. The third column is then chosen to place sensor 2 as shown in Figure 16.

The Matrix M

Id. Sensor	The time slots					
0	0	null	null	null	null	null
1	null	1	null	null	null	null
2	null	null	2	null	null	null
3	0	1	null	null	null	null
4	0	null	2	null	null	null
5	null	null	null	null	null	null
6	null	1	2	null	null	null
7	null	null	null	null	null	null
8	0	null	2	null	null	null
9	null	1	2	null	null	null

The Vector Index

Id. Sensor	The Index
0	0
1	1
2	2

Figure 16: The sensor 2 is processed

The algorithm is applied to sensor 3. This sensor can't be placed in the same column as sensors 0 or 1 because it has a bi-directional link with them. It cannot place in the same columns as sensor 2 because they share sensors 6 and 4. The fourth column is then chosen to place the sensor 2 as shown in Figure 17.

The Matrix M

Id. Sensor	The time slots					
0	0	null	null	3	null	null
1	null	1	null	3	null	null
2	null	null	2	null	null	null
3	0	1	null	3	null	null
4	0	null	2	3	null	null
5	null	null	null	null	null	null
6	null	1	2	3	null	null
7	null	null	null	null	null	null
8	0	null	2	null	null	null
9	null	1	2	null	null	null

The Vector Index

Id. Sensor	The Index
0	0
1	1
2	2
3	3

Figure 17: The sensor 3 is processed

To place sensor 4, we have to choose a free column because the constraints cannot be satisfied with the occupied ones. The fifth column is then chosen as shown in Figure 18.

The Matrix M

Id. Sensor	The time slots					
0	0	null	null	3	4	null
1	null	1	null	3	null	null
2	null	null	2	null	4	null
3	0	1	null	3	4	null
4	0	null	2	3	4	null
5	null	null	null	null	null	null
6	null	1	2	3	null	null
7	null	null	null	null	null	null
8	0	null	2	null	4	null
9	null	1	2	null	null	null

The Vector Index

Id. Sensor	The Index
0	0
1	1
2	2
3	3
4	4

Figure 18: The sensor 4 is processed

Sensor 5 cannot be placed in the same column as sensor 0 because they share sensor 8. However, it can be placed in the same column as the sensor 1. Indeed, there is no link between them and they do not share any neighbor. Hence, the column chosen to place sensor 5 is the second one as shown in Figure 19.

The Matrix M

Id. Sensor	The time slots					
0	0	null	null	3	4	null
1	null	1	null	3	null	null
2	null	null	2	null	4	null
3	0	1	null	3	4	null
4	0	null	2	3	4	null
5	null	5	null	null	null	null
6	null	1	2	3	null	null
7	null	null	null	null	null	null
8	0	5	2	null	4	null
9	null	1	2	null	null	null

The Vector Index

Id. Sensor	The Index
0	0
1	1
2	2
3	3
4	4
5	1

Figure 19: The sensor 5 is processed

To place sensor 6, we have to choose a free column because the constraints cannot be satisfied with the occupied ones. The last column is then chosen as shown in Figure 20.

The Matrix M

Id. Sensor	The time slots					
0	0	null	null	3	4	null
1	null	1	null	3	null	6
2	null	null	2	null	4	6
3	0	1	null	3	4	6
4	0	null	2	3	4	null
5	null	5	null	null	null	null
6	null	1	2	3	null	6
7	null	null	null	null	null	6
8	0	5	2	null	4	null
9	null	1	2	null	null	6

The Vector Index

Id. Sensor	The Index
0	0
1	1
2	2
3	3
4	4
5	1
6	5

Figure 20: The sensor 6 is processed

Sensor 7 can be placed in the same column as sensor 0 because there is no link between them and they do not share any neighbor. Hence, the column chosen to place sensor 7 is the first one as shown in Figure 21.

.

The Matrix M

Id. Sensor	The time slots					
0	0	null	null	3	4	null
1	null	1	null	3	null	6
2	null	null	2	null	4	6
3	0	1	null	3	4	6
4	0	null	2	3	4	null
5	null	5	null	null	null	null
6	7	1	2	3	null	6
7	7	null	null	null	null	6
8	0	5	2	null	4	null
9	7	1	2	null	null	6

The Vector Index

Id. Sensor	The Index
0	0
1	1
2	2
3	3
4	4
5	1
6	5
7	0

Figure 21: The sensor 7 is processed

For sensor 8, the algorithm cannot find any suitable column because constraints are not satisfied with occupied columns and there is no free column anymore. Thus, the algorithm creates a new empty column at the end of M and place sensor 8 in it as shown in Figure 22.

The Matrix M

Id. Sensor	The time slots						
0	0	null	null	3	4	null	8
1	null	1	null	3	null	6	null
2	null	null	2	null	4	6	8
3	0	1	null	3	4	6	null
4	0	null	2	3	4	null	8
5	null	5	null	null	null	null	8
6	7	1	2	3	null	6	null
7	7	null	null	null	null	6	null
8	0	5	2	null	4	null	8
9	7	1	2	null	null	6	null

The Vector Index

Id. Sensor	The Index
0	0
1	1
2	2
3	3
4	4
5	1
6	5
7	0
8	6

Figure 22: The sensor 8 is processed

The case of sensor 9 is same as the previous one: it is necessary to create a new column as shown in Figure 23.

The Matrix M								
Id. Sensor	The time slots							
0	0	null	null	3	4	null	8	null
1	null	1	null	3	null	6	null	9
2	null	null	2	null	4	6	8	9
3	0	1	null	3	4	6	null	null
4	0	null	2	3	4	null	8	null
5	null	5	null	null	null	null	8	null
6	7	1	2	3	null	6	null	9
7	7	null	null	null	null	6	null	9
8	0	5	2	null	4	null	8	null
9	7	1	2	null	null	6	null	9

The Vector Index	
Id. Sensor	The Index
0	0
1	1
2	2
3	3
4	4
5	1
6	5
7	0
8	6
9	7

Figure 23: The sensor 9 is processed

5. Experimental Results

In this section, we assume that the network being created is centralized, in which all the nodes send the packets to the base station.

Generally, each layer in WSN works separately. This work aims to implement the proposed protocol named OIdTDMA explained in the third section to ensure that there are no collisions that result from hidden and exposed terminal problems in MAC sublayer, also the use of joint between the MAC sublayer and network layer. In addition, we implement routing protocol in the third layer.

The aim of this joint use is that the nodes recognize when to be active and find a suitable route to the base station, which helps to minimize the delay and, in addition, avoiding the problems in finding the correct path.

We will analyze the performance of OIdTDMA protocol by simulation in a JUNG simulator built in JAVA.

JUNG (Java Universal Network/Graph) is a free open-source software library that provides a common and extensible language for the modeling, analysis, and visualization of data that can be represented as a graph or network.

JUNG contains implementations of algorithms from graph theory, data mining, and social network analysis, such as routines for clustering, decomposition, optimization, random graph generation, statistical analysis, calculation of network distances, flows, and important measures (centrality, PageRank, HITS, etc.) [21].

Our evaluations are based on the simulation of different networks composed of 50, 80, 100, 150, and 200 sensor nodes uniformly and randomly distributed within an area of 100 x 100 m. In all simulations, the transmission range of all sensor nodes is set to 25m. Our simulation depends on a simple model of LEACH [7] for the radio hardware energy dissipation to calculate the energy. Using this radio model, to transmit a k-bit message at distance dist, we use equation 4.1:

$$E_{Tx}(k, dist) = E_{Tx-elec}(k) + E_{Tx-amp}(k, dist) = E_{elec} \times k + \epsilon_{amp} \times k \times dist^2 \quad (4.1)$$

To receive a k-bit message, we use equation 4.2:

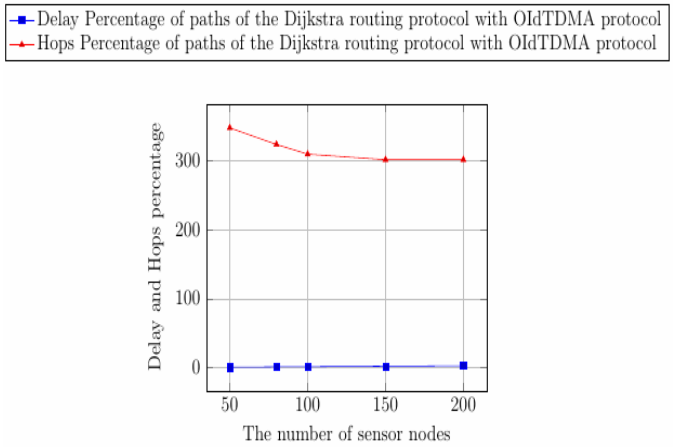
$$E_{Rx}(k) = E_{Rx-elec}(k) = E_{elec} \times k \quad (4.2)$$

The values of parameters used for simulations are shown in table 1.

No. Item	*Item*	*Item Description*
	Network Parameters	
1.	Simulation Area	100X100 m
2.	No. of sensor nodes	50,80,100,150,200
3.	Scope of connection	25m
	MAC Parameters	
	ID	Identifying for each sensor
4.	Slot Duration	1s
5.	Transmission time	1s
	Radio Parameters	
6.	Transmitter and receptor Ampifier Energy Dissipation: (a) Etx_amp (b) Erx_amp (c) Eelec (d) Eamp (e) Length	 Energy amplification Energy reception Energy transmission/Electronic reception(50nj/bit) Amplification factor(0.0013nj/bit/m4) The distance between transmitter and receiver
7.	Channel Type	Channel/wireless channel
8.	Communication model	Bi-direction

Table 1. Parameters used in Simulations

The communication model assumes that the radio link between two nodes is bidirectional, i.e, if node v can hear node u, then node u can hear node v. Each node is assumed to have a unique node ID.



Delay and Hops percentage of paths of the Dijkstra routing protocol with OIdTDMA protocol.

As shown in the figure above, we adopted the implementation of OIdTDMA protocol in the second layer and Dijkstra protocol in the third layer. We can see that initial results are successful for the two layers, one depends on the other. As described in the previous sections, the hidden terminal problem produces collisions preventing data to reach the base station and, therefore, implies to re-send it again. This additional sending causes a delay, which increases hops and power consumption.

We can see that OIdTDMA resolves this problem. Indeed, hops percentage shows that OIdTDMA protocol

permits to avoid collisions which cause hidden terminal. Hence, there is no more useless additional packet sending. Furthermore, we can see that hops percentage is decreasing for different wireless sensor network depending on the shortest path used to reach the base station. OIdTDMA protocol creates a number of slots to allocate time between sensors. These slots are defined using the maximum degree of the network. Each time a slot is active, the associated sensor is allowed to send and receive data.

Delay percentage increase very slightly with the number of sensors in the network, leading to an increase in the number of slots and thus an increase of the delay.

6. CONCLUSION

As explained in previous sections, this work aims to use MAC scheduling with a routing algorithm to avoid the collisions which occur in hidden terminal problems and evaluate some metrics like delay, hops count and quality of the routing solution.

This work was divided into several sections: first, we presented the wireless sensor networks and treated layers of it with the structure of the protocols. Then we made a state of the art for existing protocols in MAC sublayer and in the network layer. In particular, we focused on the work on their joint use. This research has led us to discover an unresolved problem with the TDMA protocol: hidden and exposed terminal problems, which occur when there are heavy communications in the network. In the third section of our work, we proposed the solution to this problem: OIdTDMA. We have tried to solve these problems with a new TDMA named OIdTDMA. In this protocol, the allocation of time slots is determined by considering the identification order of each sensor node in the network.

Finally, we presented the results obtained from the application of OIdTDMA using existing protocol in network layer such as Dijkstra routing protocol. The results show that we have obtained a reduction of packet collision between sensors and thus, we avoid the hidden and exposed terminal problems. For the future work, we will try to reduce delays which produce after processing the collisions.

ACKNOWLEDGMENT

Acknowledgment To: my dear parents, my dear brothers, my dear sisters, all my family, all my friends, all the people who helped me in any way.

REFERENCES

[1] Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., & Cayirci, E. (March 2002). Wireless sensor networks: A survey. Computer networks, vol. 38, no 4, p. 393-422.

[2] Arampatzis, Th, John Lygeros, & S. Manesis. (June, 2005). "A survey of applications of wireless sensors and wireless sensor networks." Intelligent Control. Proceedings of the 2005 IEEE International Symposium on, Mediterrean Conference on Control and Automation., pp. 719-724.

[3] Arumugam, M. a. (2005). Self-stabilizing Deterministic TDMA for Sensor Networks. Distributed Computing and Internet Technology, 69--81.

[4] Kulkarni, S. S. (2006). Infuse: A TDMA based data dissemination protocol for sensor networks. International Journal of Distributed Sensor Networks, 2, 55--78.

[5] Dhanaraj, M. a. (2005). A new energy efficient protocol for minimizing multi-hop latency in wireless sensor networks. Pervasive Computing and Communications, PerCom 2005. Third IEEE International Conference on, 117--126.

[6] Heidemann, J. a. (2006). Research challenges and applications for underwater sensor networking. Wireless Communications and Networking Conference (pp. 228--235). IEEE.

[7] Heinzelman, W. B. (2002). An application-specific protocol architecture for wireless microsensor networks}. Wireless Communications, IEEE Transactions on, 1, 660--670.

[8] Intanagonwiwat, C. a. (2003). Directed diffusion for wireless sensor networking. Networking, IEEE/ACM Transactions on, 11, 2--16.

[9] Karn, P. (1990). MACA-a new channel access method for packet radio. In ARRL/CRRL Amateur radio 9th computer networking conference}, 140, 134-140.

[10] Karp, B. a.-T. (2000). GPSR: Greedy perimeter stateless routing for wireless networks. Dans Proceedings of the 6th annual international conference on Mobile computing and networking (pp. 243--254). ACM.

[11] Kumar, S. a. (2006). Medium access control protocols for ad hoc wireless networks: A survey. Ad Hoc Networks, 4, 326--358.

[12] Kunert, K. (s.d.). TDMA-based MAC Protocols for Wireless Sensor Networks: State of the Art and Important Research Issues,2005.

[13] Lu, G. a. (2004). An adaptive energy-efficient and low-latency MAC for data gathering in wireless sensor networks. Dans Parallel and Distributed Processing Symposium, 2004. Proceedings. 18th International (p. 224). IEEE.

[14] Ming Lu, Y. a. (2007). An energy-efficient multipath routing protocol for wireless sensor networks. International Journal of Communication Systems, 20, 747--766.

[15] Shih, E. a.-H. (2001). Physical Layer Driven Protocol and Algorithm Design for Energy-efficient Wireless Sensor Networks. Dans Proceedings of the 7th Annual International Conference on Mobile Computing and Networking (pp. 272--287). Rome, Italy: ACM.

[16] Shukur, M. I. (2009). Wireless Sensor Networks: Delay Guarantee and Energy Efficient MAC Protocols. Proceedings of World Academy of Science: Engineering and Technology, 50.

[17] Skiena, S. (1990). Dijkstra's Algorithm. Implementing Discrete Mathematics: Combinatorics and Graph Theory with Mathematica, Reading, MA: Addison-Wesley, 225--227.

[18] Van Dam, T. a. (2003). An adaptive energy-efficient MAC protocol for wireless sensor networks. Dans Proceedings of the 1st international conference on Embedded networked sensor systems (pp. 171--180). ACM.

[19] Ye, W. a. (2002). An energy-efficient MAC protocol for wireless sensor networks. Dans INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE (Vol. 3, pp. 1567--1576). IEEE.

[20] Yener, M. C. (2006). Efficient Scheduling for Delay Constrained CDMA Wireless Sensor Networks.

[21]Jung-java universal network/graph framework, [en ligne]. jung.sourceforge.net (page consultée le 11 Juin 2013).

[22] Baicher, A. A. (2012). Wireless Sensor Network Architecture. Dans CNCS2012 (pp. 11--15). City Campus, Usk Way, NP20 2BP, Newport, U.K: IACSIT Press.