RTH-MAC: A Real Time Hybrid MAC protocol for WSN

Djalel ABDELI
LIA/UERI
EMP
Algiers, Algeria
{a_djalel}@yahoo.fr

Saoussan ZELIT, Samira MOUSSAOUI
LSI, Computer Science Department
USTHB
Algiers, Algeria
{zelit_saoussan, moussaoui_samira}@yahoo.fr

***Abstract* — Recently, Wireless Sensor Networks (WSNs) grow to be one of the dominant technology trends; new needs are continuously emerging and demanding more complex constraints, such as the real time communication. The MAC layer plays a crucial role in these networks; it controls the communication module and manages the medium sharing. In this work, we describe the Real Time Hybrid MAC (RTH-MAC) protocol for WSNs. It combines the advantages of both TDMA and FDMA in order to offer a soft real time communication to a converge-cast WSN randomly deployed. It schedules nodes communication in a central manner that eliminates collisions, minimizes interferences, and ensures a small bounded end-to-end delay. Moreover, this protocol offers a probabilistic end-to-end reliability guarantee using an acknowledgment mechanism, and offers to the end user the possibility of adapting the duty cycling ratio. Simulation results show that RTH-MAC presents good performances in term of the end-to-end delay, reliability and scalability.**

***Keywords* — WSN, Real time, MAC protocols, Multi-channel, TDMA, deadline, latency, reliability, end-to-end guarantee.**

I. INTRODUCTION

WSNs form a many-to-one network, where all sensor nodes cooperate to collect and forward data to one or a few nodes called sinks. These sinks might be at kilometers away from nodes, and cannot be reached in an only one hop, wherever the multi-hop communication will be required. Recently, WSNs ensure wide range of potential applications in different domains such as military, healthy and industrial area; and numerous emerging applications require more Quality of services (QoS) constraints such as the support of real time communication.

In this work, we are interesting by the MAC sub-layer because it plays a crucial role in WSN and especially for QoS provisioning, since it controls the medium access sharing, and all upper layer protocols are bounded to that. Thus, QoS support in the network or the transport layers cannot be provided without the assumption of a MAC protocol which solves the problems of medium sharing and supports reliable communication.

Since, several commercial sensor nodes (such as Imote2, MICAz and TelosB) are equipped by IEEE 802.15.4 **[1]** compatible radio transceivers, which have the operating frequency range of 2400- 2483.5MHz, and support up to 16 different channels, the use of multi-channel communication is considered as a novel tendency because it offers the ability of parallel transmissions over different channels within the same spatial domain, which minimizes interferences, collisions and contentions on the wireless medium, and improves the capacity of a network and maximizes the concurrency **[2], [3]**.

Therefore, this paper proposes the Real Time Hybrid MAC (**RTH-MAC**) protocol for wireless sensor networks. The RTH-MAC is a hybrid protocol based on multiplexing the Time Division Multiple Access (TDMA) method and the Frequency Division Multiple Access (FDMA) method in order to offer a hard real time communication to a converge-cast multi-hop WSN randomly deployed. It ensures a deterministic end-to-end delay by avoiding packet contention and collision. It minimizes the end-to-end latency by assigning time slots in a sequence that allows a continuous data flow from a source node to the sink in one super-frame duration. RHT-MAC increases the end-to-end reliability by implementing an acknowledgment/retransmission mechanism. Moreover, the RTH-MAC possesses an active/sleep mechanism for efficient energy usage with predefined duty cycle.

The rest of this paper is organized as follows: we present real time systems and theirs characteristics in Section II. In Section III, we discuss the related works. The network model and assumptions are presented in Section IV. The scheduling algorithm is detailed in Section V. We introduce an Acknowledgment/ Retransmission mechanism to improve the end-to-end reliability of the protocol in Section VI. Finally, we conclude with a summary of our findings and our future works.

II. REAL TIME IN WSNs

In real time system, the total correctness of each task depends not only on its logical result, but also on the time at which results should be performed. They should have the ability to predict whether the system can verify all critical timing requirements within a finite and specified period called the deadline. The deadline may change according to application needs and should be fast enough to preserve data pertinence. According to the application requirements, real time systems can be classified into two main categories: the soft real time system (SRT) and the hard real time system (HRT). In SRT system, the response time is specified as an average value, because the single late in message reception is not significant to the system functioning, but it might degrades its quality. However, in HRT system, deterministic communication should be defined, because the deadline defines the interval significance of the message. Thus, there is no message that can arrive late, because the system will consider it as a total failure **[4]**-**[6]**.

However, Real time WSNs differ significantly from classical real time system due to the large number of additive constraints such as the wireless communication nature, the limited available resources (power, processing and memory), the low reliability of nodes and the dynamism of the network. Real time WSN can also be defined as the study of hardware and software systems that are subject of real time constraints. These systems include all components of the operational system from event detection to system response, thus a real time WSN should include a real time operating systems, real time communication networks and real time software application. Moreover, the real time requirement should be supported at each OSI layer of the standard communication model.

III. RELATED WORKS

MAC protocols for real time applications have several constraints and requirements. They have the main constraints of WSN such as the finite source of energy, the limited computation capability, the characteristics of used radio transceiver, the variable density of the network, the unreliable links, etc. Moreover, they need to satisfy requirements of real time applications wherever the most important ones' can be summarized in: respecting deadlines, minimizing the transfer delay, improving transfer reliability and offering the best tradeoff between Energy consumption, delay and reliability **[4], [5]**.

A large amount of WSN MAC protocols that intend to reduce latency and increase reliability have been proposed in the literature and an exhaustive list is presented in **[3]**-**[7]**. However, only few of these protocols offer a guaranteed bounded delay and a guaranteed reliability without topological assumption, because most of them are topological dependent.

For real time topology dependent WSNs, it exists several MAC protocols in the literature. Some protocols are based on Personal Area Network (PAN) networks, such as the **GTS of IEEE 802.15.4, i-GAME, ART-GAS, AGA, Knapsack, GSA [7]**, this category is characterized by the low data rate, the reduced number of nodes, and the limited size of the networks (few meters). Moreover, there are some protocols that support only the one-hop communication, such as **Alert**, **LPRT** and **TOMAC [6]**, this category is well adapted to 1-hop star network, and can be used especially in Body Area Network (BAN). Nevertheless, there are some independent protocols, where each one of them is designed to a well specific topology. Thus, the **I-EDF[6]** protocol is based on a cellular backbone, and each cellule has an hexagonal form. The **Dual Mode Mac [6]** protocol offers real time guarantee to linear WSN. The **AREA-MAC [6]** protocol organizes the network in a grid manner, and each node is identified by its position coordinates in the grid. The **PEDAMACS [6]** protocol is not based on a specific topology but it assumes that the sink can communicate directly with all nodes of the network. The **RT-Mac [6]** protocol is not based on a specific topology; however it is limited to an only single data stream from the source node to the sink. Moreover, there are the **GinMAC** and **Burst [4]** that require a careful planning and deploying.

MAC protocols for randomly deployed WSNs can be divided into two subcategories: those offering single hop guarantee and those offering an end-to-end guarantee. The first subcategory contains several protocols such as **FMAC, FTDMA, VTS, SUPPORTS, CR-SLF, QoS-MAC [4]**. This subcategory is not well suitable to multi-hop WSN, because it does not cover the entire path of data; thus it can be used in one-hop star WSN, in applications that tolerate some deadline missing or in combination with a real time routing protocols.

The other subcategory contains **RRMAC [8], RT-Link [9], HyMAC [10]** and **PRISM [11].** They ensure an end-to-end bounded delay guarantee by dividing time to temporal slots, organized in an efficient manner in order to reach the sink in one super-frame. The HyMAC exploits additional mechanisms, such as the multi-channel communication, to maximize parallel transmission and minimize the end-to-end delay. However, HyMAC is not reliable, it requires height latency in dense network, and it does not resolve the traffic fluctuation problem.

IV. RTH-MAC PROTOCOL

A. Network model and assumptions

The network is composed of one base station, and a set V of homogenous sensor nodes distributed randomly on the region of interest, and forming an ad hoc WSN. The network is assumed to be completely connected and synchronized, thus there is no subset of nodes totally isolated from the network. Moreover, nodes are supposed to be statics after the deployment; however the network has the ability to insert new nodes and remove exhausted ones.

The set V includes a subset S of sink nodes, this subset should contain at least one node at any time, and all nodes $s \subseteq S$ are connected directly to the base station. Each node $n \subseteq V$ has a unique identifier, and is equipped with a low-power radio compatible to the IEEE 802.15.4 standard (as the CC2420 **[12])** that offers the possibility to communicate in 16 different channels. We assume that communication is bidirectional between node, thus if a node $n1 \subseteq V$ can communicate with $n2 \subseteq V$ then n2 can communicate with n1. Moreover, each node $n \subseteq V$ can communicate with a subset $Nn \subseteq V$ of nodes determined by the range of the radio transmission, this subset is named ***neighbors*** of node n.

B. RTH-MAC description

The Real Time Hybrid MAC (RTH-MAC) protocol for WSNs is a scheduled synchronized protocol; it is destined to converge-cast multi-hop WSN, where nodes and sinks are randomly deployed in the region of interest.

The RTH-MAC protocol combines the advantages of the TDMA and FDMA access methods in order to offer a hard real time communication. Thus, it ensures a bounded deterministic end-to-end delay (by avoiding packet contention, collision and minimizing interferences). It minimizes the end-to-end latency by assigning time slots in a sequence that allows data to flow continuously from rallel transmissions without generating conflicts or increases the end-to-end reliability by implementing the (acknowledgment/ retransmission) mechanism. Moreover, the RTH-MAC possesses an active/sleep mechanism for efficient energy usage with predefined duty cycle. The RTH-MAC offers the possibility of transmitting several samples in the same time slot in order to deal with the traffic fluctuation problem. Finally, include an update mechanism in order to tolerate some topological changes such as the insertion of new nodes in the network, or the departure of exhausted ones.

Our contributions consist of adapting the MRPM **[14]** protocol in order to be used in the initialization phase to minimize the energy consumption and the phase duration. Secondly we have improved the scheduling algorithm of the HyMAC protocol in order to minimize the number of used time slots and channels and to render it more flexible to topological changes. Moreover, we have added an acknowledgment/retransmission mechanism in order to improve the end-to-end reliability; and we have modified the super-frame form to minimize the energy consumption of nodes.

The RTH-MAC functioning is organized in four different phases as illustrated in Fig.1. After nodes deployment, all nodes turn-on their radio transmitter, and wait until receiving a Hello message. When each node receives a Hello message, it begins its discovery step by saving the source address of the message in its neighbor list, and broadcasting another Hello message with its proper ID as source address. After the collection of direct neighbor's information, all nodes forward their neighbors lists to the base station. After that, the base station construct a fully connected graph G(V,E) that represents the network.

The Slot/channel attribution phase consists of constructing the schedule plan, basing on the graph G, by attributing to each node n⊆ V the appropriate slot time and frequency to communicate. Therefore, the base station will diffuse the schedule to nodes, and only after this step that nodes can send data packet.

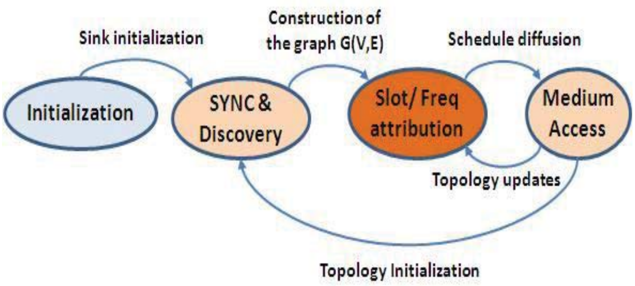

Fig.1. RTH-MAC functioning

C. *The discovery phase*

The discovery phase of RTH-MAC aims to initialize the network, collect information's on neighbors of each node, and gather these information's in the base station in order to construct a graph of nodes. Thus, it is an important step in the protocol functioning, and it should be determined prudently in order to minimize the energy consumption and reduce the duration of this phase.

Several protocols in the literature ignore this phase or use a simple contention method, such as the CSMA/CA, which is the most known contention method in wireless network, but it consumes a lot of energy in idle listening. Therefore, enhanced contention methods consist of using slotted time contention period in order to plan sleep period and reduce the energy consumption of nodes. The most known protocol of the latest category is SMAC **[15]**, but it suffer from height overhead and from non optimized contention period, which consumes more energy and introduces more loss time. We have found in the literature the MRPM MAC protocol **[14]**, which is a synchronized contention based protocol, it reduces the energy consumption comparing to CSMA/CA and SMAC by minimizing overhead and contention periods, and it reduces end-to-end latency. Therefore, we have proposed a discovery mechanism based on an adaptation of the MRPM protocol in order to minimize the energy consumption and reduce the duration of the discovery phase.

The discovery phase is composed of three different steps; it begins by collecting information about neighbors of each node, after that all nodes diffuse their neighbor lists until reaching a sink. Finally the base station constructs a graph of sensor nodes from the received information.

Therefore, All nodes are programmed to turn-on their radio transmitter after the deployment, and to wait in the LPL state until receiving the first Hello message, which initializes the node, synchronizes it with the discovery frame, and indicates that it is a neighbor of the receiving nodes. The user start the initialization phase of the network from the base station by sending a Hello message to all sinks nodes (s⊆ S); it contains the initialization time slot size and the timeout of the collection phase, and it is illustrated in Fig.2. After that, each sink contend (in the contention period) in the beginning of each time slot in order to broadcast its Hello message to its adjacent nodes. Therefore, when each node n⊆ V-S receives a Hello message, it begins its discovery phase by synchronized itself with the discovery frame, it saves the source address from the Hello message in its neighbor list, and it decreases the phase collection by 1. Therefore, it contends to diffuse its proper Hello message with its proper ID as source address and its switch its radio transmitter to the receive state in listening periods in order to receive Hello's of neighboring node and to update its neighbor list. When a node succeeds in sending a Hello it initializes a local counter at x, this counter will defined by the end user, and it will be decremented in each time slot until reaching 0, and in this time node tries to retransmit another Hello message with the appropriate timeout. Moreover, in order to diminish the priority of resending a new Hello message comparing to the priority of sending the first Hello message, each node increases the size of its contention window (CW) after each contention win, until reaching the maximum CW allowed.

Length	Type	Dest @	Src @	seqn	Sleep time	Timeout	Duration

Fig.2. Packet structure of the Hello message

After the timeout of the collection phase, all nodes contend to broadcast their neighbors list in order to forward them (basing on the flooding mechanism) to the base station until receiving the schedule from the sink. Moreover, the access is organized similarly to the previous step, and nodes diffuse their lists using the MRPM frame structure.

After gathering the topology information of the whole network, the base station construct a fully connected graph G(V,E), where the vertices of the graph denote the nodes (n⊆ V) and the (n1,n2) edge denotes that n1 is a direct neighbor of n2 which means that n2⊆ Np of the n1 node, thus each one of them can communication directly with the other.

D. Super-frame structure of the medium access phase

The protocol is a centralized scheduled MAC protocol. Thus, after constructing the schedule, time will be divided to fixed length frames, and each frame will be divided to a fixed number of time slots. Moreover, slots are organized in two periods: the contention free period and the contention period. The form of the super-frame is illustrated in Fig. 3.

The number of slots in the contention free period is calculated during the slot/freq attribution phase, and it depends on the expected node density, the number of available channel and the number of sinks in the network; and our work tries to minimize this number. However, the number of slots in the contention period should be fixed by the end user according to topological changing frequency.

Scheduled slots are used to transmit useful data. They are composed of two main parts, the first part is used to transmit data, and the second one permits to switch-off the radio transmitter. The data part duration T_{Data} can be done by:

$$T_{Data} = L_{maxdata} / D \quad (1)$$

Where $L_{maxdata}$ is the maximum length of data that can be transmitted to sink including the ACK and the retransmission possibility, and D represents the throughput supported by the radio transmitter.

However, the sleep part is used to minimize the energy consumption of each node. The amount of the sleep part is defined by the duty cycling ratio (**DCR**), which is defined by the end user. Thus, the scheduled time slot duration T_{sched} can be calculated as:

$$T_{sched} = (100/DCR) * T_{Data,} \quad (2)$$

Since the protocol is designed for real time application, and to ensure meeting the deadline. Thus, the respect of packets deadline should be satisfied using the following equation:

$$T_{sched} * (Ncs + Nss) < \text{deadline} \quad (3)$$

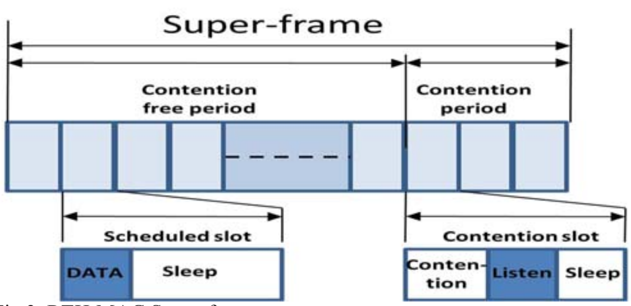

Fig.3. RTH-MAC Super-frame

E. Topological updates

The contention period is used to update the topology of the networks, thus new nodes and leaving ones send their notification message in this slots. At the beginning of the contention period, all nodes of the network switch their channel to the defined contention channel (by default the first one). Thus, new nodes in the networks content to send their Hello message, and those have exhausted batteries (less than 5%) content to send their Quit message (inspired from **RT-Link**). All other nodes are in receiving state, if they receive a notification message, they save it and send it to the base station in the next frame; otherwise they switch-off their radio until the next slot. Therefore, the base station reschedules the network communication if it is necessary.

V. RTH-MAC SCHEDULING ALGORITHM

A. Design modeling

This work tries to minimize the number of scheduled slots by exploiting the different available channels. Thus, this schedule should maximize parallel transmissions without generating conflicts or collisions by removing unneeded links between adjacent nodes, and basing on forming different broadcast domain on the same time slot, each broadcast domain is defined on a different channel. As demonstrated in **[9]**, the use of this principle reduces the degree of the topology and can reduce the average energy consumption by more than a factor of 3.

Without the use of the multi-channel capability, the construction of the minimum delay schedules is similar to the NP-complete distance-2 graph coloring problem, and many practical heuristics should be applied **[9], [10], [13]**.

However, multiplexing TDMA and FDMA complicates the schedule of communication. Thus, we reformulate the schedule construction problem as the distance-2 multi-coloring graph problem. Each node n⊆ V should be colored by two different colors (u,v), where u⊆C1 and v⊆C2. C2 is limited and it represents the set of available channels, but C1 is unlimited and it represents the set of time slots which should be minimized. Therefore, if *n* is colored by the couple (u,v), the couple (u,v) should not be present on the 2-hop neighbors.

B. Scheduling algorithm

The RTH-MAC scheduling heuristic is an improvement of the HyMAC heuristic, where slots attribution is performed according to the height of the treated node independently from the other branch's, instead of attributing slots according to all nodes of the same level. Thus, RTH-MAC aims to provide the minimum number of slots for scheduling the network using the appropriate (time slot, frequency) to the nodes. Moreover, it organizes the schedule in a hierarchic manner that permits to reach one sink in an only one frame. Additionally, it supports several sinks randomly deployed in the network.

The algorithm presented in Algorithm. 1 is based on the Breath First Search (BFS) methods, it constructs a set of trees, and each one has one different sink as a root. It has three inputs: the G(V,E) graph of sensor nodes constructed in the discovery phase, the number of available channels, and the set S of sinks present in the network, this set should contain at least one sink. The algorithm results on the number of used slots and a multi-colored graph G’. Each vertex (node) of G’ is colored by a time slot and a frequency. The BS value in the algorithm represents the base station, Q contains the explored vertex, and NT is a set to save temporary data from neighbor’s nodes.

The algorithm began by attributing default value to different nodes. Thus, unexplored vertex will be marked, and sinks will have the couple of communication (1,1), unless if there is two adjacent sinks. After that, it initiates the construction of neighbors of the explored node basing on the exploration order.

Therefore, for all explored nodes (beginning by sinks), the next unexplored nodes are considered as child’s, thus each child *p* will be colored by a different time slot, and a default frequency (slot(*p*), channel(*p*)). If this couple is used in the 2-hop neighbors, the algorithm tries to find an unused channel; otherwise it increments the time slot and repeats the 2-hop checking test of used couples.

Algorithm 1: RTH-MAC Scheduling algorithm

Input: - The G(V,E) graph of sensor nodes
- The number of available channel

Output: - The colored graph G’

Scheduling:

```
while Q is not empty do
  n←Dequeue (Q); CurrentSlot←slot(n)+1
  for all 1-hop neighbors p of n do
    if parent(p)= 0 then
      parent(p) ←n; channel(p) ← 1
      slot(p) ← CurrentSlot; Enqueue (p,Q)
      Construct NT contain all 1-2-hop explored neighbors v of p
      i← 1
      while i<= length (NT)
        if slot(p)= slot(v) then
          if parent(p)= parent(v) then
            slot(p) ← slot(p)+1; channel(p) ← 1
          else
            if parent (p)< Nbr-ch then
              channel(p) ← channel(p) +1
            else
              slot(p) ←slot(p)+1; channel(p)←1
            end if
          end if
            i← 1
        else
          i← i +1
        end if
      end while
    end if
  end for
end while
```

Once the algorithm explores all nodes, all used time slots will be inverted in order to ensure data traveling from the source to the sink in a consecutive time slots, and thus the sink will be reached in an only one frame. This inversion is similar to the HyMAC inversion, and it is done as following:

$$\text{Slot (n)} = \text{MaxSlot} - \text{Slot (n)} + 1 \qquad (4)$$

Fig. 4 illustrates a sample run of the RTH-MAC scheduling algorithm on a graph of sensor networks that contains two different sinks.

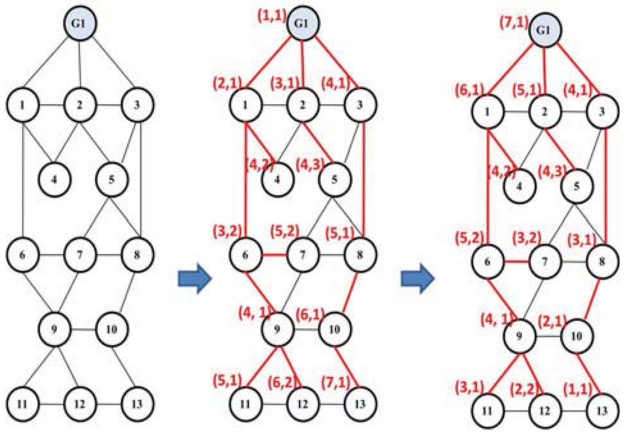

Fig.4. Attribution of the couple (Slot, Freq) to the graph G using the scheduling algorithm.

We begin the first iteration of the algorithm after attributing the couple (1,1) to the sink G1. Thus, for all 1-hop neighbors of the sink (nodes: 1, 2 and 3), we attribute the default value (2,1) and we verify if it is used in the 1-2-hop explored neighbors (explored nodes are those having a parent). Therefore, node 1 will be colored by (2,1) because it is the first explored node. However, when node 2 is colored by (2,1) it will find that the node 1 is colored by the same colors, and since both of them have the sink as parent (same parent) the node 2 should increases its slot number to (3,1). Using the similar approach, the node 3 will have the (4,1) colors after testing the (2,1) and the (3,1) couples of colors. In order to pursuit nodes exploration, we insert each colored node in the queue Q. Thus, we continue our algorithm by visiting the first node in the queue, which is the node 1 here, and we find its 1-hop unexplored neighbors which are 6 and 4 nodes. The node 6 should have the default (3,1) colors, but these colors exists in node 2 which is a 2-hop neighbor node of the node 6, and since nodes 2 and 6 have not the same parent, and there is more available channels, the node 6 will be colored with the couple (3,2). The same approach will be used until coloring all nodes of the network.

C. *Performance evaluation of the scheduling algorithm*

The performance of the scheduling algorithm has been studied through numerical simulations. Thus, we have implemented the HyMAC and RTH-MAC algorithm in MATLAB to perform numerical analysis for the relationship among the number of nodes, the node density, the number of required hops to reach the sink from the farthest node of the WSN, and the resultant number of slots. After that we study the effect of the sink position on the region of interest, and the effect of using several sinks on the number of required time slots. Finally, we study the consequences of minimizing the number of used channel on the number of required time slots.

We began by generating a random distribution of nodes in a limited region, we define the connectivity between nodes according to their separating distance, thus two nodes are considerer connected if the distance between them is less than 45, and we generate the graph G(V,E).

1. *The number of required slots according to the number of nodes and nodes density in the network*

In the first simulation, we vary the region size from 160*160 to 600*600 in order to ensure nodes connectivity and maintain nodes density around 10 neighbors max, and we place the sink in the (0,0) position. Thus, we calculate the number of required slots according to the number of used channels (1 and 16) and the number of nodes in the network.

When we use 16 channels, RTH-MAC provides very small number of used slots compared by HyMAC as illustrated in Fig. 5. However, when we use an only one channel, RTH-MAC provides a number of channels higher than those provided by HyMAC with one channel (Which is similar to RT-Link) when the number of nodes is not important, but when incrementing nodes number, RTH-MAC becomes more robust. Thus, RTH-MAC is more scalable than HyMAC with one channel; nevertheless their results are generally nearest.

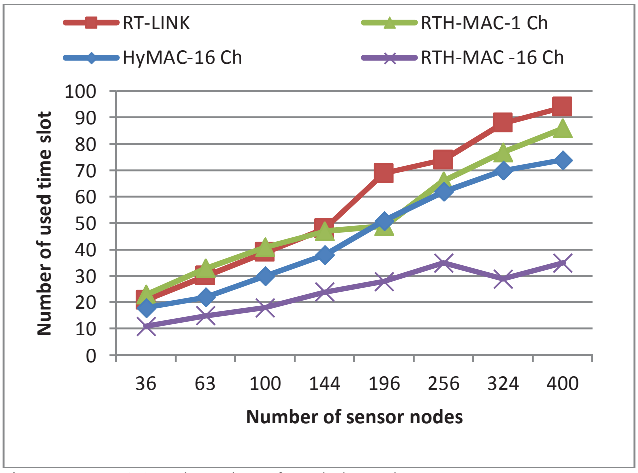

Fig.5. RTH-MAC and number of used channel

2. *The effect of the sink position and the number of used sinks in the WSN on the number of required slots*

As seen in the previous simulation, the number of required time slots is proportional to the number of hops to reach the sink, thus if the latest number increases the first number increase consequently. Therefore, we try to minimize the number of scheduled time slots by changing the position of the sink. Thus, in the simulation illustrated in Fig. 6, we have placed the sink in three different manners, wherever in the first scenario we place the sink randomly in the deployment region. In the second scenario we place the sink in the center of the deployment region. The last scenario is similar to previous simulations where the sink is placed in the (0,0) position.

Therefore, we have calculated the number of required time slots according to the sink position and the number of nodes in the network, we have used 16 channels for both protocols (RTH-MAC and HyMAC), we have varied the size of the deployment region from 350*350 to 1000*1000, and the number of used sensor nodes from 144 to 900 nodes. As illustrated in Fig. 6, for both protocols, the use of a sink positioned at coordinates (0,0)

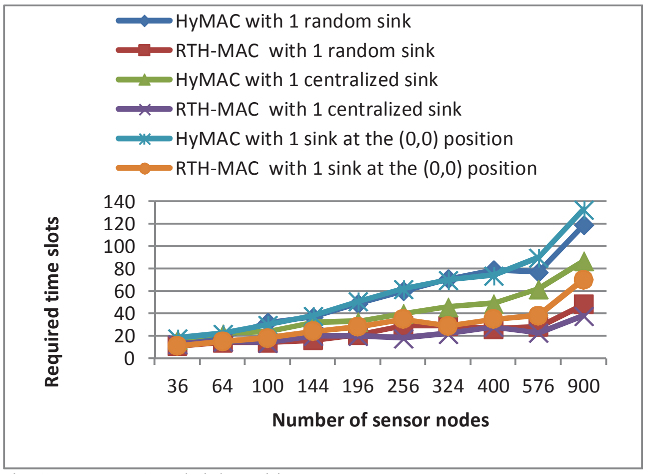

Fig.6. RTH-MAC and sink position.

gives bad results comparing to other positions, such the HyMAC requires 133 time slots and the RTH-MAC requires 70 time slots. This result can be explained by the increase of the number of hops to reach the sink, and if the number of required time slots is very important it renders impossible to meet shortest deadline.

However, the use of randomly deployed sink can reduce the number of required time slots, because it can minimize the number of required hops to reach the sink. However, the random deployment cannot decreases the number of required time slots for all positions, and it cannot be considered as a reference of improvement, because it can use poor position than (0,0).

The centralized position of the sink gives generally best results comparing to fixed position at the (0,0) for both HyMAC and RTH-MAC, and especially when increasing the size of the deployment region and the number of user sensor nodes. We can see in Fig. 6 that the number of used time slots shoots down from 70 to 38 in RTH-MAC, and from 133 to 87 in HyMAC.

After testing the effect of the sink position on the number of required time slots, we have testing the effect of using several sink in the WSN. Thus, we have first modified our initialization algorithm as shown in Algorithm 2 which consists of attributing time slots and frequency to sinks. Thus, all sinks communicate on the first channel, and all of them have the first time slot as a default one. However, if a sink has another sink in its one or two neighbors it attribute to the sink a new time slot not used by neighbors sinks.

After that, as illustrated on Fig. 7, we have calculated the number of required slots according to the number of sinks and the number of nodes in the network, we have used 16 channels, and we have used the same sizes of networks used in the previous simulation. Moreover, we have deployed sinks randomly in the region of interest in order to study the effect of using several sinks without introducing the effect of sinks positions. As illustrated in Fig. 7, the use of several sinks randomly deployed in the WSN, does not always improve the protocol results, especially in small networks. Thus, the use of several sinks is more advantageous on an extended WSN such seen in the Fig. 7 where the use of 2 sinks basing on RTH-MAC for a network of 900 nodes decreases the number of required time slots from 48 time slots to 35.

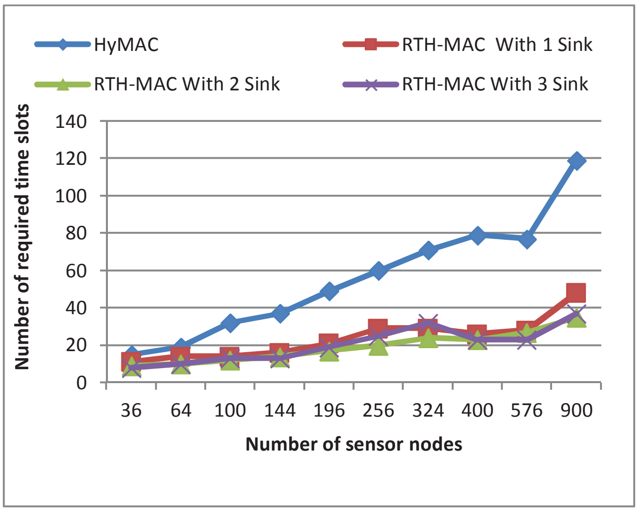

Fig.7. RTH-MAC with several random sinks

3. The effect of the number of available channels on the number of required slots

Radio transceivers compatible to Zigbee (IEEE 802.15.4) such as the CC2420 are used in commercial motes as micaZ, Imote2, and Tmote. They communicate in the RF frequency range of 2400 MHz to 2438.5 MHz, having steps of 5 Mhz. Thus, they have the ability to communicate in 16 different channels, these channel are numerated from 11 to 26. As mentioned in **[16]**, the use of the multichannel communication decreases significantly the Packet Error Rate (PER), however it does not eliminate definitively interferences. Moreover, in **[1]** we find that the adjacent channel rejection (±5MHz channel spacing) is at (45dB and 30 dB), and the alternate channel rejection (±10MHz channel spacing) is at (54dB and 53 dB), however the other rejection (±15MHz channel spacing) are at (62dB and 62 dB). Thus, the use of alternate channel should be more efficient than the use of adjacent channel. Moreover, the use of channel spacing by 15 MHz or more is the efficient manner.

Therefore, we study in the next simulations the effect of using alternate channel only and the use of channel spacing by 15 MHz on the number of required slots.

In the simulation illustrated in Fig. 8, we have varied the region size from 160*160 to 1000*1000 and the number of sensor nodes from 36 to 900 in order to ensure nodes connectivity and maintain nodes density around 10 neighbors max, and we have placed the sink in the (0,0) position.

Thus, we have calculated the number of required slots according to the number of used channels (2, 4, 6 and 8) and the number of nodes in the network. As, seen in the previous simulation, 6 or 8 channels are sufficient for our algorithm, thus the use of mire channels gives the same results, for that reason we have limited this simulation to 8 channels.

Fig. 8 shows that the use of 4 channels is sufficient to give the best results in small or large network if the node density is around 10 nodes. It shows also that RTH-MAC that uses 2 channels outperform HyMAC uses 8 channels.

D. Summury of finding

RTH-MAC is designed to use several channels, thus its performances go better than performances of similar protocols (such as HyMAC) when using multiple channels. Comparing to HyMAC, it decreases the number of required time slots in the majority of cases, it is more scalable, and it necessitate fewer number of channels.

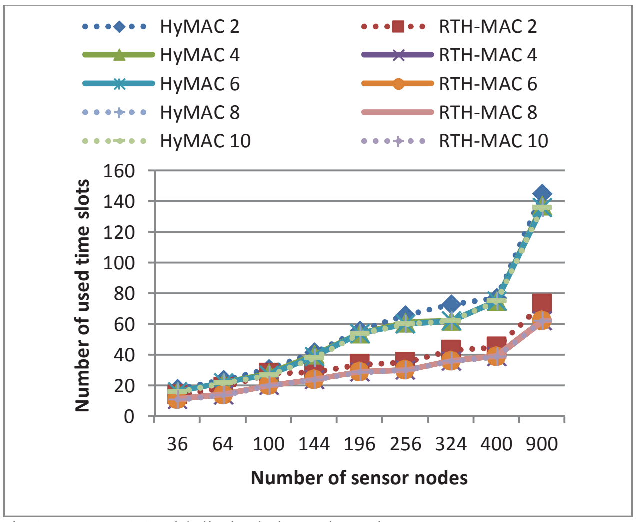

Fig.8. RTH-MAC with limited channel number

Since increasing the number of deployed nodes involves the need of more time slots, raising node density can decrease the needed number of slots because the path of data will be reduced. Moreover, we had found that the number of required hops to reach the sink influence negatively the number of the required time slots, thus we recommend to place the sink in the central position of the deployment region, or in another position that minimize the number of required hops to attain the sink in order to optimize the number of required time slots. In large WSN and when we have the possibility to insert several sinks, it can be more efficient to insert additional sink in the network, but in pre-studied position.

In order to minimize interference and increase reliability our algorithm tolerates the use of minimum number of communication channels. Thus, we can use only 4 channels (which are 11, 16, 21 and 26) in network with feeble density (around 10 neighbors), and we can use 6 different channels (which are 11, 14, 17, 20, 23 and 26) at network with huge density (around 33 neighbo rs), which permits the use of non-adjacent channels.

VI. IMPROVING THE END-TO-END RELIABILITY

In order to support real time applications for WSN, communication protocols should increase the transfer reliability and ensure a probabilistic or deterministic guarantee of successfully deliverance. The end-to-end message transfer reliability is measured by the rate of packet loss or by the packet delivery ratio (PDR) from the source node to the sink.

$$PDR = 1 - PER \quad (5)$$

Where, PER represents the Packet Error Rate of the radio transmitter.

Several techniques are used to improve the transfer reliability and they are presented in **[4]**; the main solutions can be summarized in: eliminating collisions, minimizing interferences, the use of multiple paths, the use of the Automatic repeat request (ARQ), and the use of redundant time slots for each node.

The ARQ mechanism is used to improve the transfer reliability. It consists of sending data and waiting for an acknowledgement (ACK) during a predefined timeouts. If the sender receives the ACK within the attending period it concludes the successfulness of the packet transmission. Otherwise, it concludes that there is a transmission failure, and the protocol should attempt to retransmit the packet.

Moreover, the ACK can be explicit or implicit. The explicit one consists of sending a packet named ACK message to inform the sender by the successful reception. However, the implicit ACK can be concluded if the receiver forwards the same packet to other nodes. The latest mechanism improves the transfer reliability and reduces potentially the end-to-end delay and the energy consumption.

HyMAC protocol and our scheduling protocol are based on a scheduling mechanism that eliminates the collisions and minimizes the interferences, which permits to improve links reliability and the end-to-end deliverance ratio. However, they do not improve the reliability of weaken links. Thus, we try to introduce additional mechanism to our protocol in order to improve its reliability. In this contribution we study the use of the ACK and redundant slots mechanisms and their consequences on the end-to-end reliability of our protocol.

In multi-hop converge-cast communication, the explicit ACK mechanism could be implemented in several ways; it can be implemented at each hop or by end-to-end ACK where sinks acknowledge to the source node the successful reception. Moreover, the implicit ACK can be used only if work on path aware MAC protocols. Furthermore, the number of retransmissions should be defined carefully to avoid the end-to-end deliverance degradation.

Therefore we should study three propositions applicable to our scheduling algorithm. The first one is the use of scheduling without ACK, and this scenario is similar to HyMAC protocol. The second solution is to introduce an explicit ACK and retransmission on each time slots. The last solution is to use the implicit ACK and redundant free contention period.

A. The Scheduling without ACK

The HyMAC protocol does not implement a retransmission mechanism because it indicates that the protocol is free collision, and that it eliminates interferences. However, collision can occur from communication of other networks, and interference may be happen due to the adjacent channel communication. Moreover, weaken links have always a ratio of packet error, as mentioned in **[1]**, the PER is at 1% in communication of -82 dBm.

The probability of the successful n hops deliverance Pb(snd) and it is calculated as:

$$Pb(sed)= (1\text{-}R)^n \tag{6}$$

The packet loss ratio Pb(pl) of a packet in the end-to-end path is then calculated by:

$$Pb(pl)=1\text{-} Pb(sed) \tag{7}$$

B. The use of explicit ACK and retransmission at each slot

As aforementioned the use of the explicit ACK and waiting a timeout for retransmission is a mechanism used in several protocols, it improve the packet transfer reliability because it includes more chances to retransmit the packet in each time slots. However, this mechanism increases the end-to-end latency since it introduces more overhead time such as the Transmit/Receive/Transmit switching time, and the ACK timeout, which could be no efficient for real time application that necessitate small deadline. For that reason we have use an only one retransmission possibility in each time slot.

The successful one hop deliverance probability Pb(s1d) is equal to the probability of one hop deliverance in the first tentative, which is equal to (1-R), or the probability of one hop deliverance in the second tentative, which is equal to the probability of one fail and one success in the one hop deliverance of the packet. Thus, the Pb(s1d) can be calculated as:

$$Pb(s1d)= (1\text{-}R) + R*(1\text{-}R) = (1\text{-}R)(1+R) \tag{8}$$

Thus, the successful two hops deliverance probability can be calculated as the probability of the one hop successful deliverance (first hop) and the probability of the one hop successful deliverance (second hop). And so on until reaching the n hop which is the sink. Thus the successful end-to-end deliverance probability equal to:

$$Pb(sed)= = [(1\text{-}R)(1+R)]^n \tag{9}$$

C. The use of implicit ACK and redundant free contention period

The use of the implicit ACK reduces the time slot size because it eliminates the Transmit/Receive switching time, and the ACK timeout. It consist of putting the source node in the listening state when the next hop (destined) node perform a transmission, if it forwards the same packet, the source node consider that it has receive the packet successfully, otherwise it consider that there is a transmission error, and it should retransmit the packet in the next redundant slots. For this reason, the redundant slot should not be adjacent to the emitting slot because it cannot conclude the implicit ACK. Thus, we have placed the redundant slots in a new free contention period similar to the first contention period and named the redundant free contention period. However, the last hop requires that the sink performs an explicit ACK in its proper time slot (because it is the last time slot in each period). We have used an only one redundant free contention period in order to meet near deadlines.

The Pb(sed) is equal to the probability of successful end-to-end deliverance in the first free contention period, or the probability to fail in the first hop and succeed in the end-to-end deliverance in the redundant contention free period, or the probability to succeed in the first hop, fail in the second hop and succeed in the rest end-to-end deliverance in the redundant contention free period. And so on.

Thus, Pb(sed) can be done by:

$$Pb(sed)= (1\text{-} R)^n * (1+n*R) \tag{10}$$

D. Numerical analyses of the three proposed scenarios

We can see from previous formulas that the Pb(sed) of the first scenario is less that the Pb(sed) of the third scenario, which is less than the Pb(sed) of the second scenario, because:

$$(1\text{-}\ R)^n < (1\text{-}\ R)^n * (1+n*R) < [(1\text{-}R)(1+R)]^n \quad \text{when } R\neq 0 \text{ and } n> 1 \text{ hop} \tag{11}$$

Thus, the use of explicit ACK with retransmission in the same slot is more reliable than the use of implicit ACK with redundant free contention period, which is more reliable than do not using any retransmission mechanism.

Since end-to-end transfer reliability is proportional to the number of hops to reach the sink, and to the PER, we have varied these parameters and calculated the end-to-end successful deliverance noted Pb(sed). As shown in Fig. 9, the ignorance of implementing a reliability mechanism (such as in HyMAC) reduces the end-to-end successful packet deliverance to less than 55% if the sink is far away from the source node by 60 hops even if the PER is reduced to 1%. Thus, a protocol that tolerates 45% of packet loss is not considered as a good protocol for real time application. However, when we use the redundant slots with implicit ACK we improve the end-to-end reliability to 87% where the number of required hops to reach the sink is 60, and PER is 1%, but the increase of the PER to 5% reduces the Pb(sed) to 18% which is not acceptable. Then, if we are in WSN with stable links ensuring a 1% PER we can use the redundant free contention period to improve reliability, otherwise it is preferable to avoid this mechanism and especially when the number of required hops to reach the sink is important.

As aforementioned, the use of the explicit ACK with the intra-slot retransmission gives the best ratio comparing to the two other mechanisms. Moreover, we can see in the simulation that when we use this mechanism in a network that necessitate 60 hops to reach the sink from the source node, and a PER of 1% the Pb(sed) is at 99,4% which is considered as optimal for real time application, and when we increase the PER to 5% the Pb(sed) decreases to 86%, which can be also considered as optimum for 60 hops of 5% PER.

However, in order to test the effect of the previous mechanisms destined to improve the end-to-end reliability on the end-to-end latency, we have performed a numerical simulation in order to calculate the necessary size of the frame to deliver the packet to the sink, because the frame size represents the minimum deadline supported by the protocol in the given condition. Thus, we have used characteristics of the CC2420 radio transmitter as parameters of our simulation, which are detailed in table 1.

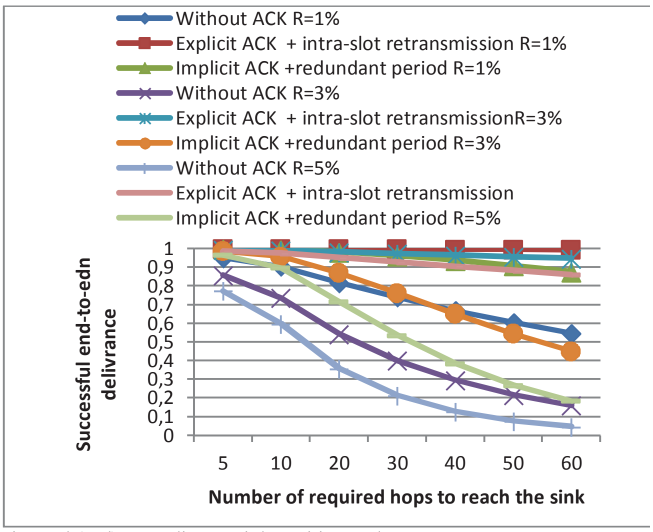

Fig.9. Pb(sed) according to sink position and PER.

We have calculated the size of the frame for the three mechanisms as following:

- Mechanism 1: Size= (Dp/ D) * number_of_time_slots
- Mechanism 2: Size= ((Dp/ D)+Ts+Tk+Ts+(Dp/ D)+ Ts+Tk) * number_of_time_slots
- Mechanism 3: Size= 2*(Dp/ D) * number_of_time_slots

Table.1. Parameters of the simulation that evaluates the size of the frame

Throughput (D)	250 Kbit/s
Rx/Tx switching time (Ts)	392 µs
Max_data_packet_size (Dp)	20 Bytes
ACK_packet_size (Dk)	8 Bytes
ACK timeout (Tk)	150 µs

Fig. 10 illustrates that the use of the second mechanism increase considerably the minimum required time to reach the sink comparing to the two other mechanisms, since at 60 hops the second mechanism requires at least 165 ms, however the third mechanism necessitate only 76 ms. However , 165 ms still sufficient for the majority of WSN applications.

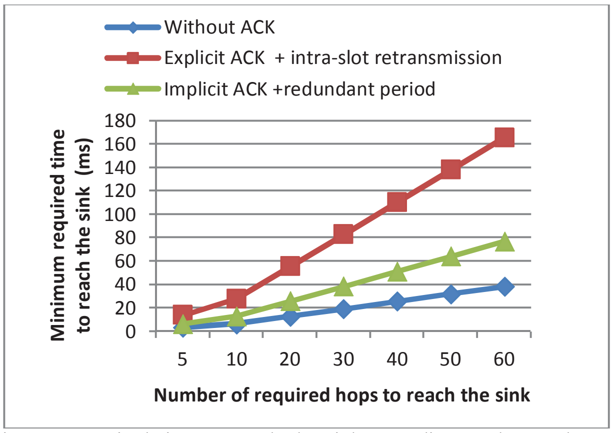

Fig.10. Required time to reach the sink according to the number of required hops to reach the sink.

E. Adopted approach

We have tested two different methods to improve the end-to-end transfer reliability, the first consists of using explicit ACK with intra-slot retransmission, the second consists of using redundant free contention period with implicit ACK, and we have compared them with the situation where there is no reliability mechanism. The first method improves significantly the protocol reliability, but it introduces more time to the end-to-end latency. However, the end-to-end latency still acceptable such it is less than 166 ms in huge network. Thus, in the situation where we have large network and an important PER, we recommend the use of the first method. However, we recommend the use of the second method (implicit ACK with redundant free contention period) in small network with negligent PER and necessitating little deadline.

VII. CONCLUSION

In this paper, we have proposed the RTH-MAC protocol which is designed for soft real time applications destined to randomly deployed WSNs. It is a hybrid TDMA/FDMA

MAC protocol that ensures a guaranteed bounded end-to-end delay, minimizes the end-to-end latency and increase communication reliability. It aims of scheduling nodes communication in an efficient manner that maximizes the parallel transmission, and ensures the deliverance of each packet in an only one super-frame. The comparison of the RTH-MAC with the HyMAC protocols shows that RTH-MAC outperforms HyMAC in term of number of used slots, number of used channel, density and scalability. Moreover, the use of multiple sinks can improve the RTH-MAC performance.

Future works are focalized on performing a test bed implementation in order to study the energy consumption of the protocol, and to validate the communication scheduling, the reliability mechanism and the initialization phase.

VIII. REFERENCES

[1] IEEE Standard 802.15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs), 2003.

[2] A. Bachir, M. Dohler, T. Watteyne, and K. K. Leung, “MAC Essentials for Wireless Sensor Networks,” IEEE Communications Surveys and Tutorials, vol. 12, no. 2, pp. 222-248, 2010.

[3] O.D. Incel. “A survey on multi-channel communication in wireless sensor networks”. Elsevier, Computer Networks (55), pp. 3081–3099, 2011.

[4] P. Suriyachai, U. Roedig, and A. Scott “A Survey of MAC Protocols for Mission-Critical Applications in Wireless Sensor Networks”, IEEE COMMUNICATIONS SURVEYS & TUTORIALS, VOL. 14, NO. 2, SECOND QUARTER 2012.

[5] M.A. Yigitel, O.D. Incel, and C. Ersoy, “QoS-aware MAC Protocols for Wireless SensorNetworks: A Survey”, Comput. Netw., Elsevier North-Holland, Inc., Vol.55, pp. 1982-2004, 2011.

[6] Z. Teng, and K. Kim, "A Survey on Real-Time MAC Protocols in Wireless Sensor Networks", Scientific Research journal, Communications and Network, 2010, Vol. 2, pp. 104-112.

[7] F. Xia, R. Hao, Y. Cao, and L. Xue. “A Survey of Adaptive and Real-Time Protocols Based on IEEE 802.15.4”. Hindawi Publishing Corporation, International Journal of Distributed Sensor Networks, Article ID 212737, 2011.

[8] J. Kim, J. Lim, C. Pelczar and B. Jang, “RRMAC: A Sensor Network MAC for Real Time and Reliable Packet Transmission”, Proceedings of International Symposium Consumer Electronics, Vilamoura, 2008, pp. 1-4.

[9] Rowe, R. Mangharam, and R. Rajkumar, “RT-Link: a timevsynchronized link protocol for energy constrained multi-hop wireless networks,” in Proceedings of IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON ’06), 2006.

[10] M. Salajegheh, H. Soroush, and A. Kalis. “HYMAC: HYBRID TDMA/FDMA MEDIUM ACCESS CONTROL PROTOCOL FOR WIRELESS SENSOR NETWORKS”. The 18th Annual IEEE International Symposium on PIMRC, 2007.

[11] E.Goldoni, and P. Gamba. “PRISM: A Novel Protocol for Real-Time Synchronous Acquisitions in WSNs”. International Journal of Sensors, Wireless Communications and Control, Vol. 1, No. 2, pp. 80-87, 2011.

[12] Datasheet, CC2420, 2.4 GHz IEEE 802.15.4/ZigBee-ready RF Transceiver, Chipcon products from Texas Instruments

[13] H. Balakrishnan, C. Barrett, V. S. Anil Kumar, M. Marathe, and S. Thite. “The distance 2-matching problem and its relationship to the mac layer capacity of ad-hoc wireless networks”. IEEE J. Selected Areas in Communications, 22(6), August, 2004.

[14] Sthapit, P. Pyun, J,Y. “Medium reservation based sensor mac protocol for low latency and high energy efficiency”. Springer Telecommunication Systems pp. 1-9 (2011)

[15] Ye,W. Heidemann, J. and Estrin, D. “Medium access control with coordinated adaptive sleeping for wireless sensor networks”. IEEE/ACM Transactions on Networking (pp. 493-506). (2004).

[16] Xiaorui, W. Xiaodong, W. Xing, F. and Guoliang, X. “MCRT: Multichannel Real-Time Communications in Wireless Sensor Networks”. ACM Transactions on Sensor Networks, Vol. 8, No. 1, Article 2, Publication date: August 2011.