An Efficient Zone-Based Routing Protocol for WSN

Kamal Beydoun[1], Khodor Hammoud[2],
Kamal.Beydoun@ul.edu.lb, khodorhammoud94@gmail.com
[1] Department of Computer Science, Lebanese University, Beirut, Lebanon
[2] Department of Computer Science, Lebanese University, Beirut, Lebanon

ABSTRACT

A Wireless Sensor Network is a collection of sensor nodes that cooperate with each other to send data to a base station. These nodes have limited resources in terms of energy, memory, and processing power. Energy conserving communication is one of the main challenges of wireless sensor networks. Several studies and research are focused on saving energy and extending the lifetime of these networks. Architectural approaches, like hierarchical structures, tend to organize network nodes in order to save energy. Most of these protocols need background information on the network for them to be efficient. In this paper, we describe a new approach for organizing large sensor networks into zones, based on the number of hops, to address the following issues: large-scale, random network deployment, energy efficiency and small overhead. This network architecture enables a hierarchical network view, with the purpose of offering efficient routing protocols based on zone partitioning. Simulations undertaken demonstrate that our approach is energy-efficient; this is highlighted by the reduction of traffic overhead.

Keywords
Wireless Sensor Network — Hierarchical Routing.

1 Introduction

Technological advances in microelectronic and wireless communications have enabled environmental monitoring using small sensor devices grouped in new types of networks called wireless sensor networks (WSNs). Wireless Sensor Networks (WSNs) are dense wireless networks made up of small, low-cost sensors (nodes), distributed randomly along a designated area. These sensor nodes collect and disseminate environmental data. Wireless sensor networks facilitate the monitoring and controlling of physical environments from remote locations with good accuracy. Sensor nodes have limited resources: small batteries, small memory and small processing power. They are equipped with both sensory devices allowing data sensing, and wireless transceivers that help them communicate. When detecting a stimulus, sensor nodes (called sources) generate data packets and transmit them through the network to one or several special nodes (called sinks). Direct communications would be possible if large transmission power on the transmitter node is used. However, in largely deployed networks, high transmission power wouldn't be enough to reach the sink and would consume a lot of energy. This problem can be overcome by multi-hop communication; the number of hops should be minimized in order to save energy. Sensors thus play a double role: data generator and data router.

When a node wants to send a data packet, it sends it along a route. The process of sending data from source to destination is called Routing. Routing protocols used in WSNs construction should fill some requirements concerning energy efficiency, distributed-based algorithmic and scalability. In order to address the challenge of energy-efficient, scalable SWN communication, most existing research use hierarchical architectures such as cluster-based topologies. Clustering builds up groups of nodes, named clusters, according to some metrics. Each cluster has an elected Cluster Head. Its role is to assure membership management and routing, by communicating the collected data via nodes to the base station. Despite the advantages of clustering protocols, most of them require further information on the network (e.g. node energy, connectivity, geographical position). That leads to an overload in the network due to the number of sent packets. Consequently, both the energy and lifetime of the network decrease.

In wireless, mobile and multi-hop networks, routing protocols should be able to deal with random node deployment. It means that even though sensors' positions are known (manually deployed), no particular hypotheses

Manuscript received February 5, 2025
Manuscript revised February 20, 2025
https://doi.org/10.22937/IJCSNS.2025.25.2.9

concerning its neighbors can be done due to the large scale of the network (neighborhood discovery protocols need to be implemented). Therefore, sensor networks are considered a subclass of ad hoc networks because of the absence of an infrastructure. Thus, ad hoc networking may influence some routing approaches in wireless sensor networks, with respect to their topologies. In hierarchical structures, topology control can be applied to minimize the set of active nodes (switching off some of them to preserve energy) or to define coordination tasks for some particular nodes. We are interested in hierarchical structures because flat architectures generally depend on the size of the network, which makes routing approaches difficultly scalable. In either approach, an important issue that needs to be addressed is the most crucial aspect: energy efficiency.

In our work, we propose a new approach of node grouping (ZHRP[1]) in zones for large WSNs, where zone construction uses the number of hops as the metric. No other information on the network is needed. For this purpose, an inexpensive neighborhood discovery algorithm is proposed. The idea is to distribute routing roles between nodes inside a zone, avoiding cluster management (including cluster head election and rotation, cluster construction as in classical hierarchical approaches). The zone topology we propose does not intend to give a management role to specific nodes: the nodes on the zone border will help routing between the zones, and all nodes of a zone have the same function inside their zone. Moreover, it does not need prior information on the network.

2 Related work

Energy consumption is one of the main challenges in wireless sensor networks. Energy saving assures a long lifetime for the system. Another main goal is reducing the size of the stored data (e.g. routing table) in each node of the network. Recent researches in Wireless Sensor Networks are focused on increasing the lifetime of the system by decreasing energy consumption of each node in the network ([2], [3], [4]). Because of the importance of energy consumption optimization, a particular interest is oriented towards routing protocols.

2.1 WSN Routing Protocols

Ad hoc routing protocols (AODV [5], DSR [6], and DSDV [7]) may be used as network protocols for sensor networks. However, such approaches will generally not be good candidates for sensor networks because of the main following reasons ([8]): (i) sensors have low battery power and low memory availability; (ii) the routing table size scales with the network size. According to the structure of the network, the routing protocols in WSN are classified as follows ([9]):

2.1.1 Flat-based routing

In flat networks, each node typically plays the same role and sensor nodes collaborate to communicate the sensed data. Due to the large number of such nodes, it is not feasible to address each node. This consideration has led to data centric routing, where the BS (base station) sends queries to certain regions and waits for the data from the sensors located in the selected regions. Early works on data centric routing, e.g. SPIN [2] and directed diffusion [10], were shown to save energy through data negotiation and redundant data elimination.

2.1.2 Location based routing

In this type of routing, sensor nodes are addressed by means of their locations. The distance between neighboring nodes can be estimated on the basis of incoming signal strengths. Relative coordinates of neighboring nodes can be obtained by exchanging such information between neighbors like in GEAR [11] and SPAN [12] protocols. Alternatively, it may be possible to obtain location information using existing infrastructure, such as the satellite-based GPS (Global Positioning System), if the nodes are equipped with a low power GPS receiver like in GAF protocol [13].

2.1.3 Hierarchical routing

Hierarchical routing (Table 1), originally proposed in wired networks, is a well-known technique that has advantages related to scalability and efficient communications. The concept of hierarchical network architecture is also used to perform energy-efficient routing in wireless sensor networks. That is because in a hierarchical architecture, higher energy nodes can be used to process and send information while lower energy nodes can be used to perform the sensing in the proximity of the target. LEACH [14] and HPAR [15] are two known hierarchical routing protocols. Hierarchical architectures are efficient ways to lower energy consumption performing data aggregation and fusion in order to decrease the number of transmitted messages to the BS.

Table 1. Some Available Hierarchical Routing Protocols

SOP	• Path having the minimum energy consumption per bit • Path having the maximum capacity
PEGASIS	• With CDMA: linear chain • Without CDMA: 3-level chain-based scheme
HPAR	• Intra-zone : max-min zPmin algorithm • Inter-zone: Bellman-Ford
energy*delay	• Intra-cluster : Dijkstra • Inter-cluster: CHs form a chain
HEED	• Inter-cluster: one hop between CH and the cluster member • Intra-cluster : Ad-hoc routing protocol like DSR, DD.
LEACH	• Nodes communicate data to their CH • CHs communicate directly to the BS
Top-Down	• Hierarchy made of member nodes and CHs • Member nodes send data to their parents (CH or not)
APTEEN	• Hierarchy is made of CHs • Nodes send data to CHs which forward it to the BS via other CHs
TTDD	• Geographical forwarding
Hier Clustering	• An existing routing infrastructure is assumed
VGA	• Routing metrics are not specified
CMLDA	• Routing metrics are not specified

2.2 WSN Clustering:

Nodes are gathered in several groups, generally disjointed, which are named clusters. Each cluster has a Cluster Head (CH). The sensors collect data and send it to the CH. CHs can communicate with the Base Station (BS) directly or via other CHs. In some networks, the CHs, referred to as gateways, will perform data aggregation, and send only relevant information through long haul radio communication to the BS. For that purpose, CHs will have specialized processing and telecommunication capabilities, and fewer energy constraints. If the CH is elected only once, we describe these networks as “static” in terms of change of CHs ([16], [17]). On the contrary, in “dynamic” networks, nodes exchange the role of CH (re-election) according to several metrics like remaining energy, connectivity with other nodes ([18], [14], [19]).

There are many existing clustering protocols. LEACH [14] is a distributed clustering-based protocol that uses randomized rotation of the CHs to evenly distribute the energy load among the sensors in the network. LEACH assumes that the fixed sink is located far from the sensors and that all sensors in the network are homogeneous and battery-constrained. Lin’s protocol [20] is a distributed clustering technique for large multi-hop mobile wireless networks. The cluster structure is controlled by the hop distance. In each cluster, one of the nodes in the cluster is designed as cluster head. Other nodes join a cluster if they are within a predetermined maximum number of hops from the cluster head. HEED [21] is a distributed clustering protocol that periodically selects cluster heads according to a hybrid function between their residual energy and a secondary parameter, such as node proximity to its neighbors or node degree.

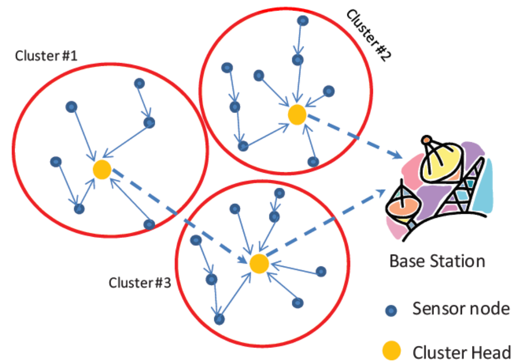

Figure 1: Clustering in WSN

Most topologies based on clusters assume that cluster heads are high-energy nodes and their transmission power can be adapted in order to reach the base station at far distances and to communicate directly to other cluster heads. Another assumption is that nodes within a cluster can directly communicate to the cluster head. The transmission defines the set of neighbors for a sensor node, those able to receive the transmitted signals. Because variation of the transmission range consumes more resources, virtual topologies should be proposed for sensor networks that are made of sensors with fixed transmission power.

Here are some hierarchical protocols listed according to their classes:

- Groups
  - SOP [22]: Nodes are classified into Groups, then a hierarchical tree is formed to create a routing table at every node. Two metrics are used in routing for choosing the path. The first is the minimum energy consumption for transmitting 1 bit. The second is the path capacity in terms of bits. Transmission always done in the path with maximum capacity.
- Clusters
  - LEACH [14]: Two hop communication is done, in the first one a node in a cluster sends its data directly to the cluster head of its

cluster, the second is that the cluster head forwards the data to the base station directly.

- Chains
  - PEGASIS [23]: Nodes uses codes in communication to reduce interference. A chain is formed, and each node sends data to its neighbor till the packet is received by the chain leader which sends the packet to the base station.
- Zones
  - ZHLS [24]: each zone consists of a center node, and nodes that are in range of the center range. The center node range is a radius in terms of hops.
  - ZHRP [1]: is a protocol consisting of zones. Unlike ZLHS, there is no center node.

The challenge addressed in this paper presents an approach of virtual structuring of networks without using topology control technique. Our contribution to topology construction addresses two main issues in WSNs: distributed approaches, and energy efficiency. Moreover, our approach is independent of the embedded sensor technology (being able to vary the transmission power or not); the only parameter considered is the current node's transmission range. The algorithm is executed simultaneously with the neighborhood discovery protocol for random sensor node deployments. Out Approach In this section we will detail the new approach for sensor grouping in multiple zones.

3 Zone Hierarchical Routing Protocol (ZHRP)

Zone Hierarchical Routing Protocol [1] is a protocol for constructing a large scale WSN where the nodes are randomly deployed. ZHRP is energy efficient, low cost, and makes efficient routing between nodes. ZHRP aims to group nodes in a large WSN into disjoint zones without the need for information, like energy level and geographic location, from neighboring nodes, which makes the WSN decrease the number of packets sent from one node to another and decrease the energy needed by the nodes that indeed increase the lifetime of the WSN. In ZHRP, no node has a management role; nodes are deployed randomly.

3.1 ZHRP Stages

ZHRP divides the WSN into zones and saves useful information for routing in each node (like nodeId, nextHopId, and number of hops). Each zone consists of an inviting node, Normal nodes, and Border nodes. The Normal nodes are nodes that can sense data and send and receive data packets. The Border nodes are like normal nodes but they are at the border of the zone and they are neighbors of other zones, and they save information about neighbor zones' Border Nodes. The Inviting node is a normal node, but it has one and only one additional task, which is to broadcast a packet to invite nodes to join its zone. Each node has a Node ID, which is unique over the WSN.

Before using it in routing, ZHRP is divided into three stages. Zones Construction stage [25], Intra- Zone Routing Table Construction stage [26], and Inter- Zone Routing Table Construction stage [26].

Nodes are classified into zones. In each zone, an Intra-Zone Routing Table will be constructed at all nodes. Then at the Border Nodes, an Inter-Zone Routing Table is constructed. When a node wants to send a packet, it uses the Intra-Zone Routing Table to send the packet to one of its zone Border Nodes. The Border Node then uses the Inter-Zone Routing Table to send the packet to the destination zone. At the destination zone, a Border Node will receive the packet, then it uses its Intra-Zone Routing Table to send the packet to the destination node.

3.1.1 Zones Construction Stage

The first stage in ZHRP is the Zones Construction stage [25]. In this stage, nodes are categorized into zones with low cost, without the need of a cluster head. There are three parameters that the Zones Construction depends on: R, zN, and N.

- R: is the zone radius, which is the maximum number of nodes between the inviting node and the invited nodes.
- zN: is the required number of zones.
- N: is the number of nodes in the wireless sensor network.

During this stage, each node specifies some attributes: ZoneId, NodeType, and BorderTable. ZoneId is the Id of the zone that the node belongs to. NodeType is the type of the node (Normal or Border node). Initially nodes are

Normal nodes but they can change to Border nodes if they are located at the border of the zone. BorderTable shown in Table 2 is the table in which the Border node saves information about the neighbor zones' Border nodes. Each record in the Border Table contains the attributes NodeId and ZoneId. NodeId is Id of the node and ZoneId is the Id of the zone which the node belongs to.

NodeId	Id of the node
ZoneId	Id of the zone that node belongs to

Table 2: Border Table

During zones construction stage, nodes exchange packets to construct the zones. One packet structure is used during the zones construction stage. The fields of this packet are shown in Table 3.

The packet Subject can be: INVITATION, DISAGREEMENT, BORDER and NEW_NODE. The algorithm of Zones construction phase is shown in Figure 2. Initially the wireless sensor network consists of Normal nodes and Inviting nodes. The number of Inviting nodes is equal to the number of zones. Initially each zone consists of one inviting node.

SrcId	Node id of the sender node
DestId	Node id of the destination node
ZoneId	Zone id of the sender node
Subject	Subject of the packet
NodeType	Type of the sender node
TTL	Packet's Time to Live

Table 3: Zone Construction Packet Fields

The inviting nodes start the construction phase by broadcasting (only once) a construction packet of subject INVITATION to its neighbor nodes that initially don't belong to any zone. This construction packet has TTL equal to R. When a node receives a packet, it first checks the Subject of the packet. If the Subject is "INVITATION", the node then checks if it is already joined to a zone or not. If it is already in a zone, then if the received packet ZoneId is the same as the node ZoneId, it does nothing. But if they are different, the node changes its type to Border node and broadcasts a construction packet of Subject DISAGREEMENT and of TTL 1, and adds a record to the BorderTable that specifies the packet sender's node id and its zone. If the node doesn't already belong to a zone, it joins the zone from the packet and checks the received packet TTL. If the TTL is 0, the node changes its type to Border and broadcasts a packet of Subject BORDER and TTL 1. Else if the TTL is greater than 0 then the node broadcasts a packet of Subject INVITATION and TTL equal to the received packet's TTL minus 1. If the received packet belongs to a different zone, and its subject is BORDER or DISAGREEMENT, then the node changes its type to BORDER and broadcasts a construction packet of subject BORDER and TTL 1. If the received packet subject is NEW_NODE, then the node sends a construction packet of subject INVITATION and TTL 1 to the sender node. At the end of this phase, NodeType, ZoneId and BorderTable will be specified for each node in the wireless sensor network.

```
INPUT : R
    PART A :
    IF (n is an inviting node, it sends the INVITATION packet P only once)
       broadcast INVITATION packet P(n.nodeId, ANY, n.ZoneId, INVITATION, NORMAL, R)
    ENDIF

    PART B : when node n receives a packet P
    IF (P.Subject = INVITATION)
       IF (n.ZoneId undefined)
          join the zone (n.ZoneId := P.ZoneId)
          IF (P.TTL = 0)
              n.NodeType := BORDER
              broadcast a BORDER packet P'( n.NodeId, ANY, n.ZoneId, BORDER, n.NodeType, 1 )
              RETURN
          ELSE
              broadcast an INVITATION packet P'( n.NodeId, ANY, n.ZoneId, INVITATION, n.NodeType, P.TTL - 1 )
              RETURN
          ENDIF
       ELSE
          IF (n.ZoneId = P.ZoneId)   RETURN
          ELSE
              n.NodeType := BORDER
              broadcast a DISAGREEMENT packet P'( n.NodeId, ANY, n.ZoneId, DISAGREEMENT, n.NodeType, 1 )
              n.BorderTable.add (P.ZoneId, P.SrcId)
              RETURN
          ENDIF
       ENDIF
     ENDIF
     IF (P.Subject = BORDER OR P.Subject = DISAGREEMENT)
        IF (n.ZoneId = P.ZoneId)  RETURN
        ELSE
           n.NodeType := BORDER
           broadcast a BORDER packet P'( n.NodeId, ANY, n.ZoneId, BORDER, n.NodeType, 1 )
           n.BorderTable.add (P.ZoneId, P.SrcId)
           RETURN
        ENDIF
     ENDIF
     IF (P.Subject = NEW_NODE)
        send an INVITATION packet P'(n.NodeId, P.SrcId, n.ZoneId, INVITATION, n.NodeType, 1 ) RETURN
     ENDIF
OUTPUT : NodeType, ZoneId, BorderTable at node n
```

Figure 2: Zone Construction Packet Received Algorithm

3.1.2 Intra-Zone Routing Table Construction Stage

The second stage in ZHRP is the Intra-Zone Routing Table Construction [26]. In this stage, nodes in the same zone will know the minimal path to send packets to each other.

destNodeId	Destination node Id
nextHopId	Next hop Id
M (metric)	Number of nodes
nodeType	Node Type of the destination node (Border or Normal)
neighZonesId	List of neighboring zones' ids if the destination node is of type Border node

Table 4: Intra-Zone Routing Table Entry Fields

The Intra-Zone table is constructed based on the Distance-Vector Algorithm (Bellman-Ford [27]). Each entry in the Intra-Zone table contains attributes as shown in Table 4. During ZHRP Intra-Zone Routing Table Construction stage, nodes exchange packets to complete the phase. The exchanged packets contain the following structure (Table 5).

srcId	Node Id of the sender
zoneId	Zone id of the destination node
destinationId	Destination node Id
nextHopId	next node Id
M (metric)	Metric computed in number of nodes
nodeType	Node type of the sender node
borderTable	If the sender node is a Border node, then it sends the Border Table

Table 5: Intra-Zone Routing Table Packet Fields

The algorithm of Intra-Zone Routing Table Construction is composed of two steps; Figure 3 shows the first step. Each node broadcasts an Intra-Table Construction packet specifying its zoneId, nodeType and borderTable (if it is a Border node). Any node that receives an Intra-Zone Construction Packet checks if the zoneId of the sender is the same as its zoneId. If it isn't, it ignores the packet; else if the received packet's sender zoneId is the same as the receiver zoneId (packet is received from the same zone) and there is no entry in the Intra-Zone Table for that sender node, then the node adds an entry to the Intra-Zone table with

- destinationId as the sender nodeId.
- metric M equals to 1.
- nextHopId same as the sender nodeId.
- nodeType as the nodeType in the packet.
- If the nodeType is Border, it adds the BorderTable from the packet.

Hence every node knows its neighbors.

```
Each node n broadcasts an Intra-Zone Routing Construction Packet(n.srcId, n.zoneId, NULL, NULL,
NULL, n.nodeType, neighZonesId)

If Node n received Packet p
If n.zoneId == p.zoneId
    addToIntraTable(p.srcId, p.srcId, 1, p.nodeType, p. neighZonesId)
End If
```

Figure 3: Intra-Zone Routing Table Construction Step1

```
Each node n broadcast its Intra-Zone Routing Table
Node n receives Packet p
If n.zoneId == p.zoneId  Then
    If IntraTable contains p.destId then
        Update Intra record
    Else
        Add new Record(p.destId, p.srcId, p.M + 1, p.nodeType, p. borderTable)
    End If
End If
```

Figure 4. Intra-Zone Routing Table Construction Step 2

Figure 4 describes the second step of Intra-Zone Routing Table construction. After knowing their neighbors, each node broadcasts its Intra-Zone Routing Table. When a node receives a packet, if the packet is from another zone, it ignores it; else the node checks if its Intra-Table contains an entry for the packet's destinationId. If so, the node checks if the packet's metric M is less than the entry's metric in the Intra-Table, and updates the entry with the new values from the packet (nextHopId, M, nodeType, BorderTable). Then, the packet rebroadcasts the modified entry.

If there is no entry for the received packet's destinationId, the node will add a new entry to the Intra-Table setting the fields (destinationId, nextHopId, nodeType, M, BorderTable) from the received packet. Then it will broadcast the new added entry. After this phase, each node will have an Intra-Table with shortest paths for packets to be sent to their destination. The

Manuscript received February 5, 2025
Manuscript revised February 20, 2025
https://doi.org/10.22937/IJCSNS.2025.25.2.9

routing algorithm will be discussed later in ZHRP Data Routing section.

3.1.3 Inter-Zone Routing Table Construction Stage

The third stage in ZHRP is the Inter-Zone Routing Table Construction [25]. In this stage, all Border Nodes will have an Inter-Zone Routing Table which contains information about other zones and the paths that should be taken to reach them.

destZoneId	Destination zone Id
nextZoneId	Next Zone Id
zoneM	Zone Metric, which is the longest path (number of nodes) between two nodes in a zone. It is computed during the Intra-Table construction

Table 6: Inter-Zone Routing Table Entry Fields

This stage is also based on Distance-Vector algorithm (Bellman-Ford [27]), which is applied between zones to form the Inter-Zone Routing Table. Each entry in the Inter-Zone Routing Table contains the fields presented in Table 6. To avoid redundant computation, a Border node is chosen in every zone to accomplish the task of building the Inter-Routing Table for the corresponding zone. This node is then called the BORDER_CHIEF node. The BORDER_CHIEF node is the node with the highest identification between Border Node identifications (border node with the highest nodeId). It is computed from the Intra-Zone Routing Table. The BORDER_CHIEF node then computes the Inter-Zone Routing Table and s ends it to all other Border nodes.

Table 7: Inter-Zone Routing Table Packet

srcId	Source node Id
nextHopId	Next hop node Id
srcZoneId	The zoneId of the Source Node
subject	Packet Subject
interZoneRoutingTable	The Inter-Zone Routing Table produced by the BORDER-CHIEF
finalDestId	The final destination nodeId

Table 7 shows the packet fields that are used during the Inter-Zone Routing Table construction.

The packet Subject can be:

COMPL_TABLE	Complete the table
UPDATE_TABLE	Update the table

Table 8: Inter-Zone Routing Table Packet Subject

The Inter-Zone Routing Table construction is composed of two stages:

The first stage: constructing the initial Intra-Zone Routing Table that contains entries for Zones that the current zone can reach directly (neighbor zones).

The second stage: the stage where zones exchange their Inter-Zone Routing Table with each other to complete them.

In the first stage, each Chief-Node constructs the Initial Inter-Zone Routing Table from their Intra-Zone Table. For each entry in the Intra-Zone table, if the nodeType is Border-Node; then for each zone in neighZonesId list, add an entry to the Inter-Zone Routing Table setting the destZoneId and the nextZoneId to zone and zoneMetric as computed during Intra-Zone Table construction. Figure 5 shows how the initial Inter-Zone Routing Table is constructed at the Chief Border Node.

```
For Each (intraEntry in Intra-Zone Table)
        If (intraEntry.nodeType == Border-Node)
                For Each (ZoneId in intraEntry. neighZonesId)
                        AddInterRecordIfNotExists (ZoneId, ZoneId, zoneMetric)
                End For
        End If
End For
```

Figure 5. Initial Inter-Zone Routing Table Construction at Chief Node

In the second stage, the Chief-Border node sends its Inter-Zone Routing table to all Border-Nodes in its zone and the neighbor nodes from the Chief-Border Border Table as shown in Figure 6.

```
INPUT : InterZoneRoutingTable at CHIEF-BORDER nodes
When node n receives a packet P
    PART A :
    IF (n.NodeType = NORMAL) THEN
     Find entryRT in IntraZoneRoutingTable | entryRT.DestNodeId = P.FinalDestId
     Send a packet P'(n.NodeId, entryRT.NextHopId, P.FinalDestId, n.ZoneId, P.Subject, P.ZoneT)
    ELSE
    PART B :
       IF (n.NodeType = CHIEF-BORDER) THEN
        UPDATE n.InterZoneRoutingTable
        IF (InterZoneRoutingTable is not complet) THEN
         TempZones := Ø
         For each entryRT in IntraZoneRoutingTable | entryRT.NodeType = BORDER DO
           For each zone in entryRT.ZoneIds | zone ∉ TempZones DO
             Send a packet P'(n.NodeId, entryRT.NextHopId, entryRT.DestNodeId, n.ZoneId, UPDATE_TABLE, n.InterZoneRoutingTable)
             Save zone in TempZones
           ENDDO
         ENDDO
         For each entryBT in BorderTable DO
           Choose randomly node from entryBT.BorderNodesIds
           Send a packet P' (n.NodeId, node, NULL, n.ZoneId, UPDATE_TABLE, n.InterZoneRoutingTable)
         ENDDO
        ELSE
           For each entryRT in IntraZoneRoutingTable | entryRT.NodeType = BORDER DO
             Send a packet P'(n.NodeId, entryRT.NextHopId, entryRT.DestNodeId, n.ZoneId, COMPLET_TABLE, n.InterZoneRoutingTable)
           ENDDO
        ENDIF
       ELSE

    PART C :
          IF (n.ZoneId ≠ P.ZoneId) THEN
           Find entryRT in IntraZoneRoutingTable | entryRT.NodeType = CHIEF-BORDER
           Send a packet P'(n.NodeId, entryRT.NextHopId, entryRT.DestNodeId, n.ZoneId, P.Subject, P.InterZoneRoutingTable)
          ELSE
             IF (n.NodeId = P.FinalDestId) THEN
              IF (P.Subject = UPDATE_TABLE) THEN
               For each entryBT in BorderTable DO
                  Choose randomly node from entryBT.BorderNodesIds
                  Send a packet P' (n.NodeId, node, NULL, n.ZoneId, P.Subject, P.ZoneT)
               ENDDO
              ELSE
                Save P.ZoneT in n.InterZoneRoutingTable
              ENDIF
             ELSE
                Find entryRT in IntraZoneRoutingTable | entryRT.DestNodeId = P.FinalDestId
                Send a packet P'(n.NodeId, entryRT.NextHopId, P.DestNodeId, n.ZoneId, P.Subject, P.ZoneT)
             ENDIF
          ENDIF
       ENDIF
    ENDIF
OUTPUT : InterZoneRoutingTable at BORDER nodes
```

Figure 6. Inter-Zone Routing Table Construction

Manuscript received February 5, 2025
Manuscript revised February 20, 2025
https://doi.org/10.22937/IJCSNS.2025.25.2.9

Figure 7 shows how nodes act when receiving an Inter-Zone Routing Table. When a Normal node receives an Inter-Zone Routing Table Construction Packet, it just forwards it to its destination. If a Border node receives an Inter-Zone Routing Table Construction Packet from another zone, it changes the nextZoneId for all the packet's Inter Table to the packet's source zoneId and then forwards the packet to the Chief-Border node. If the received packet is from the same zone and the packet's destinationNodeId is the current Border node, then:

If the packet's subject is UPDATE_TABLE, forward the packet to the neighbor Border nodes in the Border Table.

If the subject is COMPLETE_TABLE, save the packet's Inter-Zone Routing table in the current Border node. If the packet's destinationNodeId is not the current Border node, forward it to the packet's destinationNodeId (get the nextHopId from the Intra-Table).

If the Chief-Border receives the packet, it updates its Inter-Zone Routing Table. Then if the Inter-Zone Routing Table is complete (number of zones = number of entries), it sends the Inter-Table to all Border nodes in the same zone setting the packet's Subject to COMPLETE_TABLE.

If the Inter-Zone Routing Table is not complete, the Chief-Border sends its Inter-Table to all Border nodes in the same zone and to the neighbor nodes of neighbor zones setting the packet's Subject to UPDATE_TABLE.

3.2 ZHRP Data Routing

Once the Intra-Zone and Inter-Zone Routing Tables are constructed, the data routing can be accomplished easily. The packet structure used in Data Routing contains the following entries:

SrcId	Global id of the sender node
LocalDestId	Destination node Id in the zone
NextHopId	Node Id of the nextHop
FinalDestId	Final destination global node Id
DestZoneId	Destination Zone Id
Data	The data to be send

Table 9: Data Routing Packet Fields

3.2.1 Sending Data Packet

As described in [28], when a node n1 wants to send data to node n2, it follows the algorithm in Figure 8: If the srcZoneId equals to the destZoneId (in the same zone), then the Intra-Table is used to find the corresponding information to build the packet. Else, if the srcZoneId is not equal to the destZoneId, then if n1 is a Normal node then search the Intra-Table to find a Border-Node such that this Border-Node is a neighbor to the Destination Zone. If such a Border-Node is found, then send a packet to one of its neighbor node. If no Border-Nodes are found, then send the packet to any Border-Node.

If the sender node n1 is a Border-Node, then if there exists a node in the borderTable such that destinationZoneId equals the neighbor border node ZoneId, then send the packet to it.

Else, if there is no such Border Node in the Border Table, then search the Inter-Table for an entry (interRecord) such that interRecord.destZoneId equal to the packets destinationZoneId, then find a borderTable record (borderTableRecord) such that the interRecord.NextZoneId equals to borderTableRecord.zoneId and send the packet to the borderTableRecord.nodeId.

If no such borderTableRecord exists, then search the Intra-Table for a Border Node that is a neighbor to the destZoneId and send the packet to that border node.

```
When source node n wants to send DATA to destination node n'
 PART A :
 IF (n.ZoneId = n'.ZoneId) THEN
  Find entryRT in IntraZoneRoutingTable | entryRT.DestNodeId = n'.NodeId
  Send a packet P (n.NodeId, n'.NodeId, entryRT.NextHopId, n'.NodeId, n'.ZoneId, DATA)
 ELSE
 PART B :
   IF (n.NodeType = NORMAL) THEN
    TempNodes := ∅
    For each entryRT in IntraZoneRoutingTable | entryRT.NodeType = BORDER DO
      For each zone in entryRT.ZoneIds DO
        IF (zone = n'.ZoneId) THEN
         Save entryRT.DestNodeId in TempNodes
        ENDIF
      ENDDO
    ENDDO
    IF (TempNodes ≠ ∅) THEN
     Choose randomly destNode from TempNodes
     Find entryRT in IntraZoneRoutingTable | entryRT.DestNodeId = destNode
     Send a packet P (n.NodeId, destNode, entryRT.NextHopId, n'.NodeId, n'.ZoneId, DATA)
    ELSE
       Find randomly entryRT in IntraZoneRoutingTable | entryRT.NodeType = BORDER
       Send P (n.NodeId, entryRT.DestNodeId, entryRT.NextHopId, n'.NodeId, n'.ZoneId, DATA)
    ENDIF
   ELSE

 PART C :
      IF (∃ entryBT in BorderTable | n'.ZoneId = entryBT.neighZoneId) THEN
       Choose randomly node from entryBT.borderNodesIds
       Send P (n.NodeId, node, node, n'.NodeId, n'.ZoneId, DATA)
      ELSE
         Find entryZone in InterZoneRoutingTable | entryZone.DestZoneId = n'.ZoneId
         IF (∃ entryBT in BorderTable | entryZone.NextZoneId = entryBT.neighZoneId) THEN
          Choose randomly node from entryBT.borderNodesIds
          Send P (n.NodeId, node, node, n'.NodeId, n'.ZoneId, DATA)
         ELSE
            TempNodes := ∅
            For each entryRT in IntraZoneRoutingTable | entryRT.NodeType = BORDER DO
              For each zone in entryRT.ZoneIds DO
                IF (zone = entryZone.NextZoneId) THEN
                 Save entryRT.DestNodeId in TempNodes
                ENDIF
              ENDDO
            ENDDO
            Choose randomly destNode from TempNodes
            Find entryRT in IntraZoneRoutingTable | entryRT.DestNodeId = destNode
            Send a packet P (n.NodeId, destNode, entryRT.NextHopId, n'.NodeId, n'.ZoneId, DATA)
         ENDIF
      ENDIF
   ENDIF
 ENDIF
```

Figure 7. Data Packet Sending Algorithm

```
When a node n receives a data packet P
 PART A :
 IF (n.NodeId = P.FinalDestId) THEN
  Process P.data
 ELSE

 PART B :
    IF (n.ZoneId = P.DestZoneId) THEN
     Find entryRT in IntraZoneRoutingTable | entryRT.DestNodeId = P.FinalDestId
     Send a packet P' (n.NodeId, P.FinalDestId, entryRT.NextHopId, P.FinalDestId, P.DestZoneId, P.data)
    ELSE

 PART C :
       PART C.1 :
       IF (n.NodeType = NORMAL) THEN
        Find entryRT in IntraZoneRoutingTable | entryRT.DestNodeId = P.LocallDestId
        Send a packet P' (n.NodeId, P.localDestId, entryRT.NextHopId, P.FinalDestId, P.DestZoneId, P.data)
       ELSE
          PART C.2 :
          IF (∃ entryBT in BorderTable | entryBT.neighZoneId = P.DestZoneId) THEN
           Choose randomly node from entryBT.borderNodesIds
           Send P'(n.NodeId, node, node, P.FinalDestId, P.DestZoneId, P.data)
          ELSE
             PART C.3 :
             TempNodes := ∅
             For each entryRT in IntraZoneRoutingTable | entryRT.NodeType = BORDER DO
               For each zone in entryRT.ZoneIds DO
                 IF (zone = P.DestZoneId) THEN
                  Save entryRT.DestNodeId in TempNodes
                 ENDIF
               ENDDO
             ENDDO
             IF (TempNodes ≠ ∅) THEN
              Choose randomly destNode from TempNodes
              Find entryRT in IntraZoneRoutingTable | entryRT.DestNodeId = destNode
              Send P'(n.NodeId, destnode, entryRT.NextHopId, P.FinalDestId, P.DestZoneId, P.data)
             ELSE
                PART C.4 :
                Find entryZone in InterZoneRoutingTable | entryZone.DestZoneId = P.DestZoneId
                IF (∃ entryBT in BorderTable |entryBT.neighZoneId = entryZone.NextZoneId) THEN
                 Choose randomly node from entryBT.borderNodesIds
                 Send P'(n.NodeId, node, node, P.FinalDestId, P.DestZoneId, P.data)
                ELSE
                   PART C.5 :
                   TempNodes := ∅
                   For each entryRT in IntraZoneRoutingTable | entryRT.NodeType = BORDER DO
                     For each zone in entryRT.ZoneIds DO
                       IF (zone = entryZone.NextZoneId) THEN
                        Save entryRT.DestNodeId in TempNodes
                       ENDIF
                     ENDDO
                   ENDDO
                   Choose randomly destNode from TempNodes
                   Find entryRT in IntraZoneRoutingTable | entryRT.DestNodeId = destNode
                   Send a packet P' (n.NodeId, destnode, entryRT.NextHopId, P.FinalDestId, P.DestZoneId, P.data)
                ENDIF
             ENDIF
          ENDIF
       ENDIF
    ENDIF
 ENDIF
```

Figure 8. Data Packet Receiving Algorithm

3.2.2 Receiving Data Packet

As described in [28], when a node n receives a packet p it acts as the following (Figure 9):

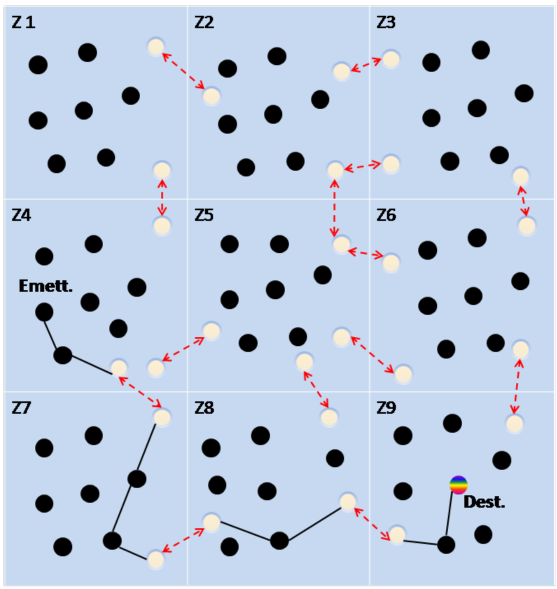

Figure 9. Packet Emitted from Zone 4 to Zone 9

If the n.nodeId equals to p.finalDestId, then the packet arrived to its destination, hence process the data. If n.zoneId equals to p.destZoneId, then p has arrived to the destination zone but not to the destination node. Then find an entry in the Intra-Table that has destNodeId equals to p.finaldDstNodeId and forward the packet to it. If the p is not in the destination zone, then if n.nodeType is a Normal-Node, forward the packet to p.localDestId. If n is a Border-Node and is a neighbor to p.destZoneId then send the packet to a neighbor node in the neighbor zone. If it is no neighbor, then find a Border Node in the same zone that is a neighbor with p.destZoneId from the Intra-Table. If such node exists, then forward the packet to it. Else, the destination zone is not a neighbor to this zone, so use the Inter-Table to know to which zone the packet should be forwarded. Figure 10 shows how a data packet is sent from Emett in Z4 to Dest in Z9.

In Z4, Emett wants to send a packet to Dest in Z9. First, it uses its Intra-Zone Routing Table to send the packet to a Border Node in Z4. At the Border Node, the node uses the Inter-Zone Routing Table to send the packet to the next zone Z7. At Z7, use the Intra-Zone Routing Table to reach a Border Node. Then use the Inter-Zone Routing Table to reach zone Z8. Z8 follows the same procedure, then the packet reaches a Border Node in zone Z9. Finally, this node will use its Intra-Zone Routing Table to forward the packet to Dest.

4 Simulations and results

In this section, we will show the implementation of all stages of the protocol ZHRP,. The implementation has been done using a simulation framework called Omnet++ [29]. Omnet++ is not a simulator, it is a discrete event network simulation framework. In other words, it provides infrastructure and tools to build network simulations. A simulator based on Omnet++ is used in building the simulation, which is called Castalia WSN simulator [30].

4.1 ZHRP Error Ratio

In this simulation, the value of R (zone radius) was 5, 15, and 25 nodes; the number of nodes was 200, 300, and 400. Figure 11 shows how the error ratio changes when R and number of zones and number of packets changes.

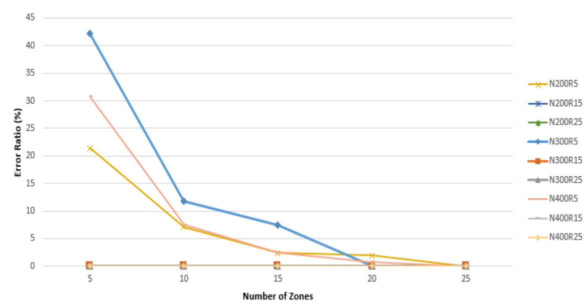

Figure 10. Error Ratio

In Figure 11, when R has the value of 10 or higher, the error ratio is always zero whatever the number of zones and the number of nodes is. On the other hand, when R has the value of 5, some nodes are not associated to any zone, but as the number of zones increases, error ratio decreases till it becomes 0 at 25 zones. That is because the number of INVITING nodes increase as number of zones increase, and the number of nodes in each zone increase as R increase, so more nodes will join the zone.

4.2 ZHRP Installation

The ZHRP installation is implemented in three stages, which are: Zones Construction, Intra-Zone Routing Table construction, and Inter-Zone Routing Table construction. In each stage, the number of Sent and Received packets is monitored when to the number of zones and the zone radius R changes. The radius can take the values 5, 15, and 25 while the number of zones changes between 5, 10, 15, 20, and 25. The simulation takes place with 200, 300, and 400 nodes.

4.2.1 ZHRP Zones Construction

Figure 12 and Figure 13 show that the increase in R correlates with an increase in the number of sent and received packets. However, when R takes a value of 15 or 25, we get an equal amount of sent and received packets. That's because zones were already neighbors, but the nodes within the radius are less than 10. Increasing R will let more nodes joins the zone, hence more packets exchange will occur. When the number of zones increases, the number of packets sent/received will increase, because increasing the number of zones will increase the number of inviting nodes, so more INVITATION messages will be transmitted.

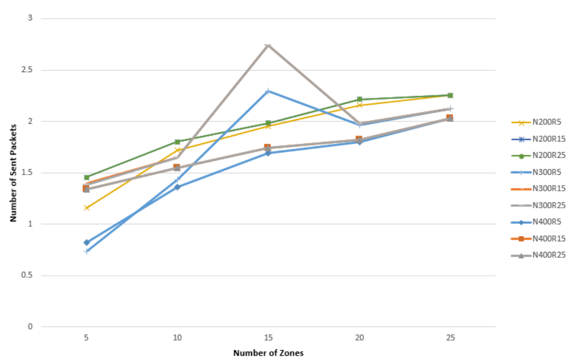

Figure 11: Zones Construction Sent Packets

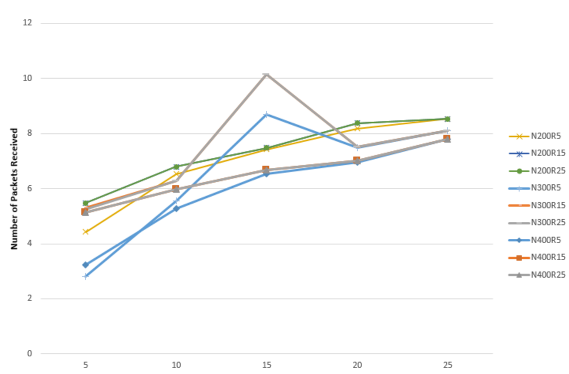

Figure 12: Zones Construction Received Packets

4.2.2 ZHRP Intra-Zone Routing Table Construction

As shown in Figure 14 and Figure 15, in Intra-Zone Routing Table construction, when the number of zones increases, the number of sent and received packets decreases. This is because the number of nodes in each zone will decrease when the number of zones increase, so the communication between nodes in the same zone will decrease. When R has the value of 10 or more, the values became the same for same number of zones and number of nodes. This is because the zone is fully constructed (have Border nodes) before R reaches 10. The number of sent packets decrease from 200 to 75 when the number of zones increases from 5 to 25, while the number of received packets decrease from 550 to 250.

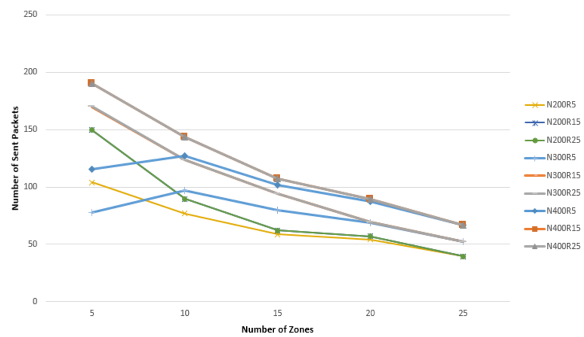

Figure 13: Intra-Zone Routing Table Sent Packets

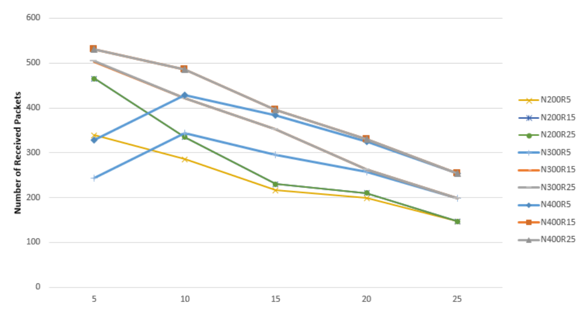

Figure 14: Intra-Zone Routing Table Received Packets

4.2.3 ZHRP Inter-Zone Routing Table Construction

Figure 16 and Figure 17 show that when the number of zones increase, the number of packets sent and received increase. When the number of nodes = 400, the number of sent and received packets will increase till they reach a maximum value of 900 for sending and 700 for receiving. When the number of nodes = 300, the number of sent packets will reach 890 and that of the received ones will reach 700. When the number of nodes = 200, the number of sent packets reach 650 and received packets reach 450. The peeks in the graphs show the maximum number of packets sent and received that the WSN can reach as long as the number of zones increases. When the number of zones is less than the value at the peek, if a zone wants to reach another zone, it must pass through many zones to reach the destination zone. But when the number of zones is greater than the value at the peek, the zone will have many neighbors. Hence, in order for a zone to reach another zone, it must pass through less zones, because its will have more neighbors, and the probability for the destination zone to be a neighbor will increase. For 400 nodes, the peek is at number of zones = 20. For 300 nodes, the peek is at 15. For 200 nodes, the peek is not reached before number of zones equal to 25.

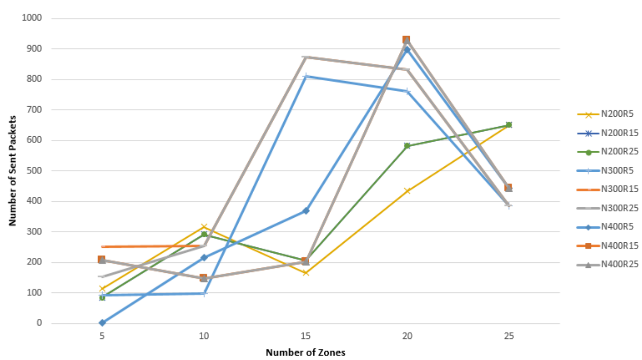

Figure 15: Inter-Zone Routing Table Sent Packets

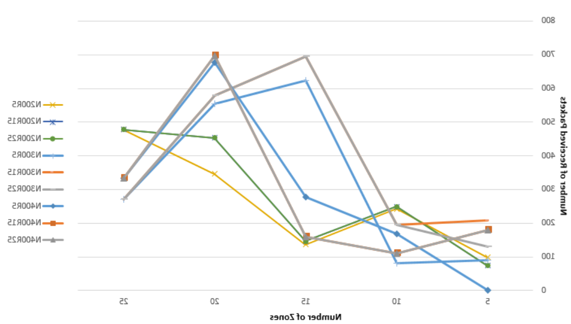

Figure 16: Inter-Zone Routing Table Received Packets

4.3 Memory capacity

One other important evaluation metric for the WSN routing protocol that we propose is the size of the data structure used for routing. Using table-driven algorithms may need important memory space in the context of largely-deployed networks. We already reduce this complexity by the two-level routing tables that we construct. Next, we are interested in the space complexity, in terms of number of bytes occupied by the involved data structures. As far as we know, no other routing mechanism proposed, in literature, a wireless sensor network that considers this metric. The formula for computing the size (in bytes) of the routing data structures is given in Table I, for N deployed nodes, when nZ zones are constructed, each zone having in average nB border nodes.

4.4 Lower bound for the number of zones

The previous metric does not only estimate the size of the needed data structure in order to assure pro-active routing based on routing tables, it also gives a lower bound of the number of zones. This computation is based on a memory limit imposed for sensors in respect with the total memory capacity of a sensor. Depending on the technology used, this total capacity may vary. Therefore, we make the following assumption: nodes technically dispose of MEM_RAM RAM memory capacity. Obviously, only a fraction of the total available memory can be used for protocol data structures. We did it by the MAX_MEM_PRCTG percentage (%). Considering the theoretical memory capacity needed by the protocol for a normal node, we have

$$10 + 12 * \frac{N}{nZ} \leq MEM\_RAM * MAX\_MEM\_PRCTG$$

This gives a lower bound for the number of zones,

$$nZ \geq \frac{12 * N}{MEM\_RAM * MAX\_MEM\_PRCTG - 10}$$

We neglected in our estimation the memory capacity for the border nodes. Meanwhile, memory constraints for border nodes can be included by varying judiciously the MAX_MEM_PRCTG constant.

4.5 Routing in ZHRP

In this simulation, a variable number of events will take place over the WSN. Nodes that receives those events will send their data to a destination node in another zone. The total number of events will vary between 300, 400, 500, and 600. The number of nodes is 400. The number of zones will change between 10, 20, and 30.

4.5.1 Sending Packets

Figure 18and Figure 19 show that the number of Sent packets when number of zones respectively equals to 10, 20.

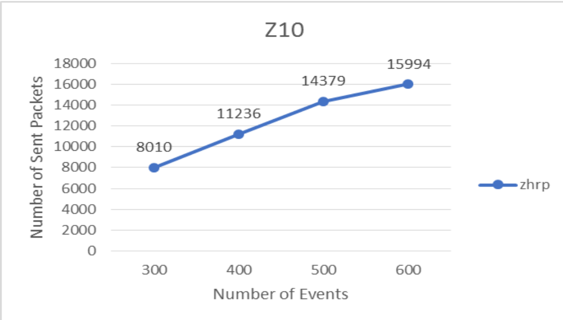

Figure 17: Z10 Sent Packets

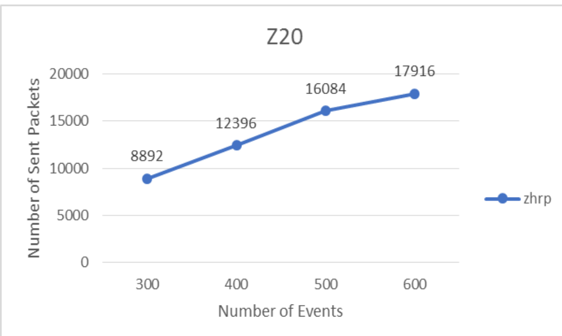

Figure 18: Z20 Sent Packets

4.5.2 Receiving Packets

Figure 20 and Figure 21 show that the number of received packets when number of zones respectively equals to 10, 20.

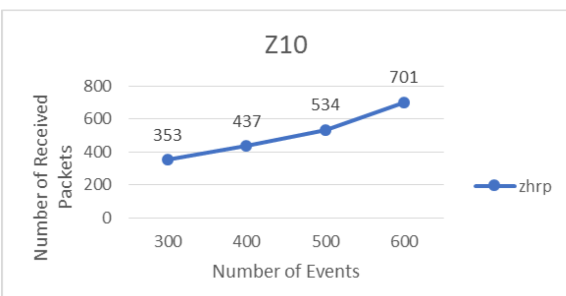

Figure 19: Z10 Received Packets

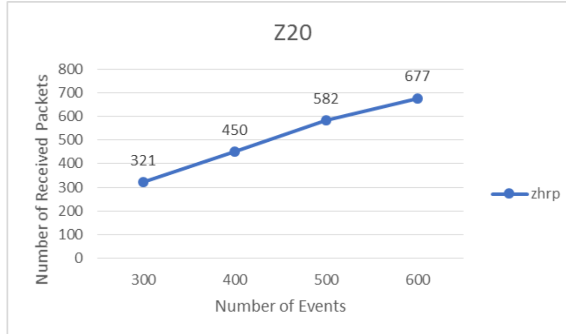

Figure 20: Z20 Received Packets

5 Battery Consumption

CPU Consumption	8 mAh
Receiving Consumption	10 mAh
Transmitting Consumption	27 mAh
Initial Energy	2900 mAh
Voltage	3 V
Data Transfer Rate	38400 bits/s
Communication Range	500ft

Table 10. Characteristics of MICA2 Sensor

Energy Consumption has been calculated based on the characteristics of the MICA2[31] sensor node. The characteristics of MICA2 is shown in Table 10. In this example, we have calculated battery consumption during ZHRP. We changed the number of nodes between 200, 300, and 400, and the number of zones between 5, 15, and 25, and for R we used the values 5 and 25. We also calculated the battery consumption during the routing scenario using ZHRP.

5.1 Battery Consumption in Zones Construction

Battery consumption for the Zones Construction stage are shown in Figure 22 for the sent packets, and in Figure 23 for the received packets.

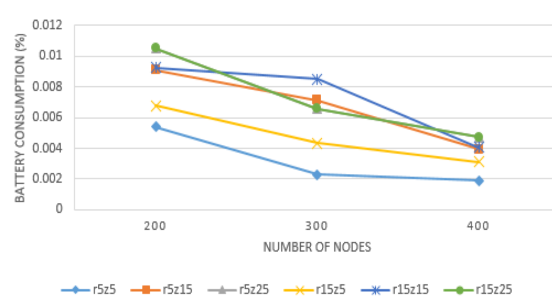

Figure 21: Zone Construction Sent Packets

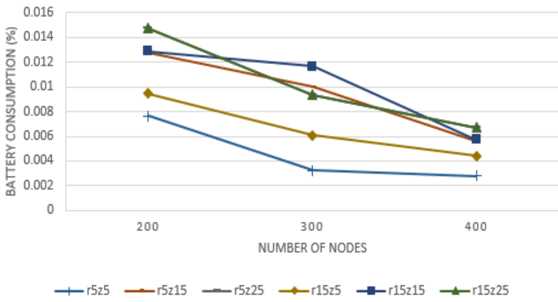

Figure 22: Zone Construction Received Packets

5.2 Battery Consumption in Intra-Zone Routing Table Construction

Battery consumption for the Intra-Zone Routing Table Construction stage are shown in Figure 24 for the sent packets, and in Figure 25 for the received packets.

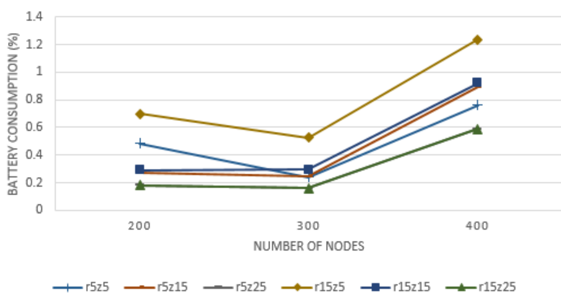

Figure 23: Intra-Zone Routing Table Sent Packets

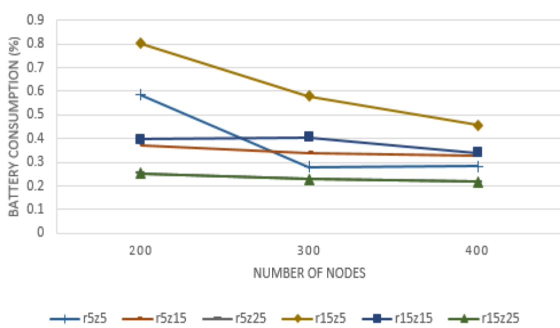

Figure 24: Intra-Zone Routing Table Received Packets

5.3 Battery Consumption in Inter-Zone Routing Table Construction

Battery consumption for the Inter-Zone Routing Table Construction stage are shown in Figure 26 for the sent packets, and in Figure 27 for the received packets.

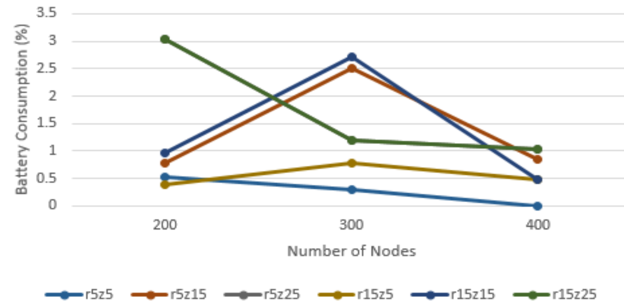

Figure 25: Inter-Zone Routing Table Sent Packets

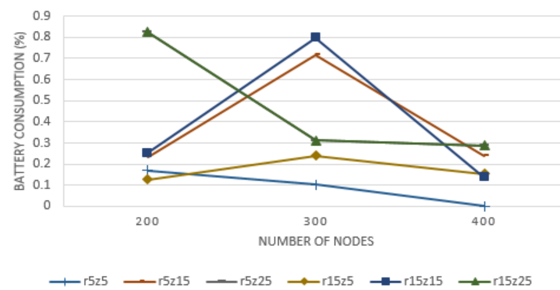

Figure 26: Inter-Zone Routing Table Received Packets

5.4 Battery Consumption in ZHRP Routing

Battery consumption in ZHRP routing are shown in Figure 28 for the sent packets, and in Figure 29 for the received packets.

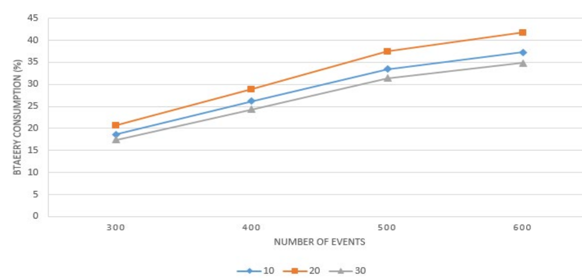

Figure 27.ZHRP Routing Sent Packets

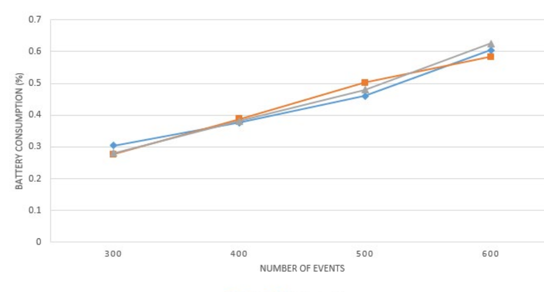

Figure 28: ZHRP Routing Received Packets

6 Conclusion

WSN is a collection of sensor nodes that are deployed randomly over a large area. Those sensor nodes have limited resources such as processing unit, battery, memory, and limited transmission range. The sensors need to cooperate to route and send data to the base station. However, this routing should be energy efficient in order to increase the WSN lifetime.

In this work, WSN was introduced with its characteristics and its applications. Then, we have described some routing protocols, which are very important for energy consumption efficiency. ZHRP, a hierarchical routing protocol, was deeply introduced and its stages were well described. ZHRP was also implemented and tested.

7 References

[1] K. Beydoun and V. Felea, "Energy-efficient WSN infrastructure," in Collaborative Technologies and Systems, 2008. CTS 2008. International Symposium on, 2008, pp. 58–65.

[2] Karl H., and Willig A. "A short survey of wireless sensor networks". Telecommunication Networks Group. Technical University Berlin, Berlin, October 2003.

[3] Akyildiz, I.F., Su, W., Sankarasubramaniam Y., and Cayirci E. "Wireless sensor networks: a survey". Computer Networks.Vol. 38, 2002, pp. 393-422.

[4] Tubaishat M.and Madria S. "Sensor Networks: An Overview". IEEE Potentials Magazine, Vol. 22(2), 2007, pp. 20-23.

[5] Perkins, C. "Ad hoc On-Demand Distance Vector (AODV) Routing". Network Working Group. http://rfc.sunsite.dk/rfc/rfc3561.html, July 2003.

[6] Broch, D., Johnson, D., and Maltz, J. "DSR The Dynamic Source Routing Protocol for Multihop Wireless Ad Hoc Networks". C.E. Perkins. Addison-Wesley. Vol. 5, 2001, pp. 139-172

[7] Bhagwat, P., and Perkins, C. Highly Dynamic Destination- "Sequenced Distance-Vector Routing (DSDV) for Mobile Computers". ACM SIGCOMM'94 Conference on Communications Architectures, Protocols and Applications. London, United Kingdom, 1994, pp. 234-244.

[8] Tilak S., Abu-Ghazaleh N.B., and Heinzelman W. "A Taxonomy of Wireless Micro-Sensor Network Models". Mobile Computing and Communications Review, Vol. 6, 2002. pp. 28-36.

[9] Kamal A.E., and Al-Karaki J.N. "Routing techniques in wireless sensor networks: a survey". IEEE Wireless Communications. Vol. 11, December 2004, pp 6-28.

[10] Estrin, D., Intanagonwiwat, C., and Govindan R. "Directed diffusion: A scalable and robust communication paradigm for sensor networks". Proceedings of ACM MobiCom' 00. Boston 2000. pp. 56-67

[11] Yu Y., Estrin D., and Govindan R. "Geographical and Energy-Aware Routing: A Recursive Data Dissemination Protocol for Wireless Sensor Networks". Technical Report, UCLA Computer Science Department, May 2001.

[12] Chen, B., Jamieson, K., Balakrishnan, H., and Morris R. SPAN: "An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks". Wireless Networks, Vol. 8, September 2002, pp. 481-494.

[13] Xu Y., Heidemann J., and Estrin D. "Geography-informed Energy Conservation for Ad-hoc Routing". Proceedings of the Seventh Annual ACM/IEEE International Conference on Mobile Computing and Networking. pp. 70-84, 2001.

[14] Heinzelman W., Chandrakasan A., and Balakrishnan H. "Energy-Efficient Communication Protocol for Wireless Microsensor Networks". Proceedings of the 33rd Hawaii International Conference on System Sciences (HICSS '00), 2000.

[15] Rus D., Li Q., and Aslam J. "Hierarchical Power-aware Routing in Sensor Networks". Proceedings of the DIMACS Workshop on Pervasive Networking, May 2001.

[16] Hebden P., and Adrian R. "Pearce Distributed Asynchronous Clustering for Self-Organisation of

Wireless Sensor Networks". Proceedings of the Fourth International Conference on Intelligent Sensing and Information Processing (ICISIP-06). Bangalore, India 2006, pp. 37-42.

[17] Niu Y., Y, Z., and Hua,F. "Sub Cluster Aided Data Collection in Multihop Wireless Sensor Networks". IEEE, Wireless Communications and Networking Conference WCNC. Kowloon, China, 2007. pp. 3967-3971.

[18] Fahmy S, and Younis O. "Distributed clustering in ad-hoc sensor networks: A hybrid, energy-efficient approach". Proceedings of the IEEE (INFOCOM) Conference on Computer Communications. Hong Kong 2004.

[19] Boutaba, R., and Aoun, B. Clustering in WSN with Latency and Energy Consumption Constraints. Journal of Network and Systems Management, Vol. 14, September 2006.

[20] H. Lin, Y. Chu. A clustering technique for large multihop mobile wireless networks. Vehicular Technology Conference Proceedings,Tokyo, Japan. Vol. 2, pp. 1545-1549. (2000)

[21] O. Younis, S. Fahmy. Heed: A hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks. IEEE Transactions on Mobile Computing. Vol.3, Issue 4, pp. 366–379. (2004)

[22] L. Subramanian and R. Katz, "An architecture for building self-configurable systems," in Proceedings of IEEE/ACM Workshop on Mobile Ad, Hoc Networking and Computing, Boston, MA, 2000.

[23] Lindsey S, Raghavendra C. PEGASIS: "power-efficient gathering in sensor information systems". In: IEEE aerospace conference proceedings, vol. 3; 2002. p. 1125–30.

[24] Hamma, T. Katoh, T. Bista, B.B. Takata, T. "An Efficient ZHLS Routing Protocol for Mobile Ad Hoc Networks". 17th International Conference on Database and Expert Systems Applications. 2006, pp. 66-70.

[25] K. Beydoun, V. Felea, and H. Guyennet, "Wireless sensor network infrastructure: construction and evaluation," in Wireless and Mobile Communications, 2009. ICWMC'09. Fifth International Conference on, 2009, pp. 279–284.

[26] K. Beydoun and V. Felea, "Wireless sensor networks routing over zones," in Software, Telecommunications and Computer Networks (SoftCOM), 2010 International Conference on, 2010, pp. 402–406.

[27] Walden, D. The Bellman-Ford Algorithm and" Distributed Bellman-Ford". [En ligne] 2009.

[28] Kamal Beydoun, "Conception d'un Protocole de Routage Hirarchique pour les Reseaux de Capteurs", PHD Report, 2009, Franche-Comte.

[29] Omnet++ User Manual, 2014. https://omnetpp.org/

[30] A. Boulis, "Castalia", Simulator for Wireless Sensor Networks and Body Area Networks User Manual Online, 2009. https://castalia.forge.nicta.com.au

[31] Crossbow. MICA2 Data sheet. [Online] 2009. http://www.xbow.com/products/Product_pdf_files/Wireless_pdf/MICA2_Datasheet.pdf.