International Journal of Smart Sensor and Adhoc Network

Volume 2 | Issue 4 Article 8

July 2012

EFFICIENT E-MRZT ALGORITHM BASED TREE CONSTRUCTION TECHNIQUE FOR ZIGBEE MOBILE WIRELESS NETWORKS

HARIRAM R M
Dept of CSE, CMR Institute of Technology, Bangalore, hariram934@gmail.com

MANJUNATHA S
Dept of CSE, CMR Institute of Technology, Bangalore, manju02@gmail.com

JITENDRANATH MUNGARA
Dept of CSE, CMR Institute of Technology, Bangalore, jmungara@yahoo.com

EFFICIENT E-MRZT ALGORITHM BASED TREE CONSTRUCTION TECHNIQUE FOR ZIGBEE MOBILE WIRELESS NETWORKS

HARIRAM R M[1], MANJUNATHA S[2] & JITENDRANATH MUNGARA[3]

[1,2&3]M-Tech (CNE), Dept of CSE, CMR Institute of Technology, Bangalore
Email:hariram934@gmail.com, manju02@gmail.com, jmungara@yahoo.com

Abstract- In Wireless Personal Area Networks (WPANs), the Zigbee protocol/IEEE 802.15.4 standard is a protocol specification for low range, less cost and low power systems. The Zigbee network is usually constructed using cluster trees for the purpose of performing data delivery among nodes and for power saving. Here the data delivery failures occur due to node mobility. In order to handle node movements so as to increase data delivery we use mobility-robust tree construction technique with clustering mechanism. In this paper we propose an E-MRZT (Extended MRZT) algorithm. We collect information about regularity of mobility patterns and by using this history, perform data delivery. We employ a clustering mechanism in our network to obtain a set of cluster trees; each with a specific mobility pattern. The entire setup is developed and simulated by using NS2 network simulator. The result is that we obtain mobility-robust tree with a considerable increase in data delivery.

Keywords: Zigbee wireless networks, Mobility robust trees, E-MRZT algorithm.

1. INTRODUCTION

Zigbee wireless networks are a class of networks which occupy the low power, low cost region in the wireless networks hierarchy.

The Zigbee standard initialized by the Zigbee Alliance [3], specifies the network and application layers for sensing data delivery.

There are various applications of Zigbee wireless networks in the real world. Applications include such as wireless light switches, electrical meters with in-home-displays, and other consumer and industrial equipment that require short-range wireless transfer of data at relatively low rates.

These applications have increased in the past decades as a result of the widespread growth in wireless communication and sensing [2].Zigbee is targeted at radio-frequency (RF) range applications that require a low range, long battery life, and secure networking. Zigbee has a defined rate of 250 kbps, has a frequency band range of 2.4GHz and supports up to 16 channels.

It is best suited for periodic, intermittent or single signal data transmission from a sensor or input device. It is also secure as it uses AES 128bit encryption technique [3].

Traffic management systems have also been implemented base on this technology.

The Zigbee network layer natively supports star, mesh, and tree topology networks. The following figure shows the Zigbee protocol stack.

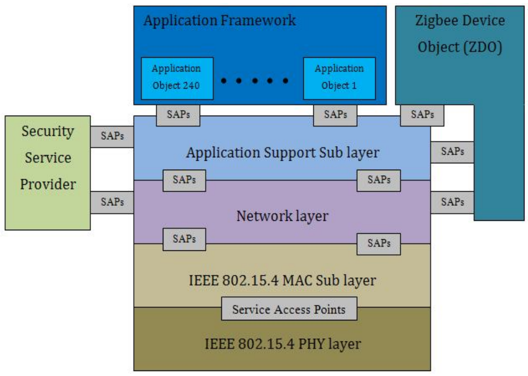


Figure 1: Architecture of Zigbee Protocol Stack

We see in the above figure that Zigbee builds upon both the physical layer and medium access control layers as specified in the IEEE 802.15.4 standard. The standard specification gets complete by adding four main components namely the network layer, application layer, Zigbee device objects (ZDO's) and manufacturer-defined application objects which could be customized and integrated as per our needs and functionality [2].

Apart from adding two high-level network layers to the underlying structure, the most significant improvement is the introduction of ZDO's. These ZDO’s are responsible for tasks like keeping of device roles, management of requests to join a network, device discovery and security[2].

There are several types of wireless networks including wireless local area networks (WLANs), mobile ad hoc networks (MANETs), Bluetooth, WiFi etc., and all these different networks have their respective ways for efficient data delivery and handling the mobile nodes in their network. They range from performing handoff (base station transfer)for cellular networks, mobile IP for WLAN and some specialized algorithms to handle routing in mobile ad hoc networks[11,12].

These different strategies and protocols are specific to their type of wireless network. This causes a problem for Zigbee based networks. Because the Zigbee networks are unique in their own way, by having low power, low cost etc, there has been several research experiments conducted regarding the mobility patterns and issues present in Zigbee networks. Our researchers have finally come up with a unique characteristic method called “mobility-robustness” [1].

In this paper, we discuss the mobility-robust tree construction in Zigbee networks by employing clustering techniques, so as to further increase the effectiveness of this technique. According to this technique, the focus is to collect information about the mobility or movement patterns of the mobile nodes across the network, and to bring the nodes with similar pattern of movement into one group, so that the mobility tree constructed for it will be specific for that group.

The aim is to introduce clustering in the Zigbee network so that instead of having one mobility-robust tree, we can have one mobility-robust tree for each identified mobility pattern. These mobile nodes, while they are moving across the network can send and receive data from routers as long as they follow their highly probable data path. The initially available Zigbee tree network is refined using graph optimization so as to form a more useful and compact network for faster and reliable robust data delivery. We use NS2 simulator for setting up the environment and simulating the “mobility-robust” tree structure.

2. SYSTEM MODEL

In a Zigbee network there are three types of devices: the Zigbee Coordinator(ZC), Zigbee Router (ZR) and Zigbee End Device (ZED). All these devices follow a certain hierarchy model in the network. Basically the Zigbee network is formed by one Zigbee coordinator and multiple Zigbee routers and Zigbee end devices. The following is a brief description [3] about their roles and functions in the network. They are:

i. Zigbee Coordinator (ZC)

A Zigbee Coordinator performs the task of initializing, maintaining and controlling the other nodes in the network. There is only one ZC node for every Zigbee network. It also acts as a router once a network is formed. It is not necessarily a dedicated device, can also perform application tasks.

ii. Zigbee Router (ZR)

It is used for storing and forwarding (routing) data between the ZED (end device/mobile node) and ZC (coordinator).It manages the local address allocation/de-allocation. It participates in multi-hop routing of messages.

iii. Zigbee End Device (ZED)

They are the mobile nodes which discover and associate with ZC or ZR. They can be optimized for very low power operation.

Though we can form Zigbee based star or mesh networks, we still opt for cluster topology because it provides for power saving and supports a very light-weighted protocol without the need to maintain a routing table.

The following figure shows a Zigbee Network.

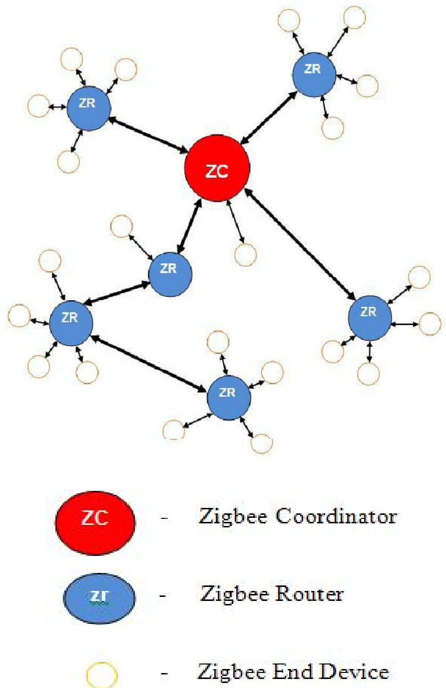


Figure 2: Zigbee Network Structure

Address assignment is different from that done in a conventional Zigbee network. Here every mobile end device in our network is assigned a random address, so as to uniquely identify every mobile node in our network.

3. RELATED WORK

In this section we present the related work done on Zigbee wireless networks. First of all, there is a constraint put on the maximum number of child routers of a router/the coordinator (R_m) and the depth of the network (L_m).Whenever a mobile end device has a packet to send, it just sends it. On receiving a packet, the router forwards it to its parent in the network.

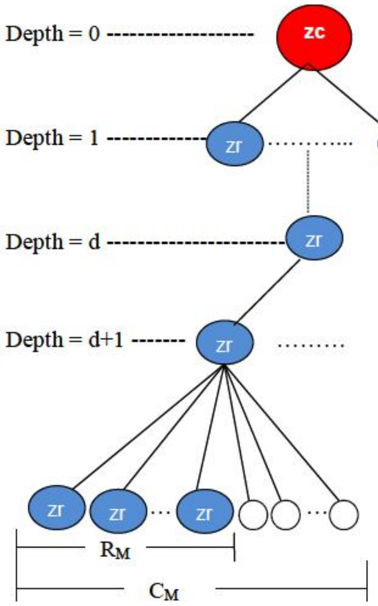


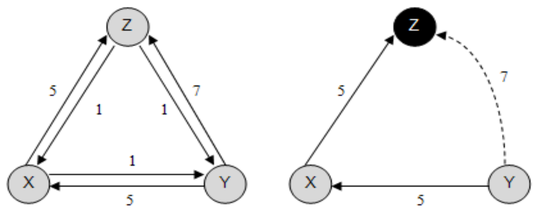
Figure 3: Tree structure of Zigbee

When a downlink packet from the coordinator is destined to a mobile end device, the coordinator delivers the packet by using the previously recorded location (i.e., the last router receiving the packet from the mobile end device) as the destination[1]. On receiving the downlink packet, the router then simply sends it to the mobile node, and expects to receive an acknowledgement from the mobile end device. If the destined mobile node has already moved out of its previous location, then the data delivery fails, and the system searches for the mobile node by sending a broadcast message throughout the network to inquire its current location.

Mobility-robust tree construction can be formulated as a graph problem [1], in which a vertex represents an immobile node, i.e., the coordinator or a router, and a directed edge represents a possible transmission link from one immobile node to another. That is, a Zigbee network is represented as $G = (V, E)$, where V is a set of immobile nodes (vertices set) and E is a set of transmission links(edges set) in the network. As the movement historical data is collected among immobile nodes, each edge $e = (u,v) \in E$ is associated with a weight, $W(e)$, which represents the number of transitions of all mobile nodes moving from the transmission range of immobile node u to that of v in the collected data. We have for any directed edge in E, say $e = (u,v) \in E$ there exists a corresponding directed edge $\bar{e} = (v,u) \in E$ in the reverse direction, which is denoted by $\bar{e}$.

The weights of the edges in E are non-negative. The method followed is to construct a Zigbee cluster tree T from the bi-directed weighted graph G. We should note that in the edge $e = (v,u)$, the node v is the parent of node u. Movements from u to v follows the upstream path, while down-link data forwarding is from v to u. The objective was to maximize the total counts of movements toward the upstream of data forwarding paths, so that we can minimize missed data deliveries caused due to the mobility of the end devices[1].

In order to achieve this objective, we define a new term mobility-robustness of the constructed tree T, as the sum of the edge weights of the chosen directed edges, $W(e), \forall\ \bar{e} \in T$, and the weights of those unselected edges, given as $W(e), e \notin T$, which connect all of the descendants-to-parent pairs of vertices in the same branch in T.



a. A connectivity graph with mobility-robustness 17
b. A spanning tree

Figure 4: An Example of Mobility-robustness scenario

4. EXISTING APPROACH

The existing approach [1] involves constructing the maximum-mobility robust tree constructed in the following manner:

Instance: An instance includes a bi-directed graph $G = (V,E)$ with edge weight $W(e) \geq 0$, $\forall e \in E$, and two positive constraint integers R_m and L_m.

Objective: The objective was to find a rooted spanning tree T in G such that the mobility-robustness of T is maximized among all possible trees in G. Also, the out-degree of every vertex in T does not exceed R_m, and the depth of T does not exceed L_m.

We now discuss the assumptions which were used in the existing approach.

A. Assumptions:

1. We assume that the coordinator maintains the location of the mobile end device when it sends an uplink data packet to the coordinator.
2. For adapting to quick topology changes there is no association mechanism between mobile end devices and routers.

Next we give a brief outline of the purpose of two algorithms, which were used sequentially to construct the Zigbee cluster tree structure.

B. Algorithms Used:

There are two algorithms used in the construction of mobility-robust Zigbee trees [1]. They are: the ZTG phase algorithm and the FIX phase algorithm.

1. **ZTG phase**: The ZTG phase searches and connects the vertices that add the most mobility-

robustness to the tree. The output of this phase is a forest of trees which is sent to the next phase/algorithm.

2. **FIX phase**: The FIX phase merges the trees constructed in the ZTG phase. These trees are directed trees, which mean that we can only connect one tree to another by their roots.

The result after applying these two algorithms is a single Maximum-mobility robust tree with reliable data flow.

The limitation of this approach is that it due to the presence of single mobility robust tree for performing routing operations to all mobile nodes in the network; it is not very efficient in quicker data delivery. And also because of changes in the network due to node movements, there is a need for clustering technique for performing efficient routing.

5. PROPOSED APPROACH

In our proposed approach, we make the following assumptions:

A. Assumptions

1. The network topology remains static during execution of the clustering algorithm.
2. Each mobile node joins a particular cluster group based on its mobility pattern.
3. All cluster groups have only one common cluster head/root(coordinator node) which communicates with all nodes.

B. Algorithms

The proposed approach consists of two parts:

1. Mobility based clustering
2. Multicast tree building

We develop a new algorithm called ***E-MRZT (Extended Mobility Robust Zigbee Tree Construction)*** algorithm. It could be considered as an enhancement of the existing MRZT algorithm, by incorporating clustering technique followed by multicast tree building involving ZTG phase algorithm for individual sub-graphs. Here instead of finding the maximum mobility robust tree as in existing approach, we find the mobility robust trees only.

Mobility Based Clustering:

In the clustering method each node's movement or mobility pattern over a period of time is taken, i.e.; the sequence of routers it visits is taken.

This information is then used as the criterion for performing clustering. Clustering in movement pattern is done to bring the nodes having similar pattern into one cluster, say, let P_1, P_2 be the mobility patterns for the mobile nodes N_1 and N_2. It can be thought of as a graph partitioning problem with added constraints.

ALGORITHM 1: MobilityBasedClustering

Input: A Bi-directional graph G=(N,E) where each mobile node $N_1, N_2, N_3, \ldots N_n$ has a corresponding mobility sequence of $moveSeq_1, moveSeq_2, \ldots moveSeq_n$

Output: Cluster Trees

```
Begin
set configurationParameter: moveSeq
for all nodes Ni, where 1 ≤ i ≤ n do
Cluster(i).node ← i
Custer(i).moveSeq ← Get moveSeq( Ni)
end for

for all nodes Ni, where 1 ≤ i ≤ n do
for all nodes Nj, where i ≤ j ≤ n do
if  i = j break;
if  (Cluster(j).moveSeq  ≅ Cluster(i).moveSeq)
append Cluster(j).node to Cluster(i).node;
Cluster(j).node ← null;
end if
end for
. end for
End
```

In the above clustering algorithm, initially we create clusters $C_1, C_2 \ldots C_n$ for all mobile nodes $N_1, N_2 \ldots N_n$. This is indicated in lines 2-4 in the above algorithm, where we add every node to every cluster and add its corresponding mobility pattern sequence "movSeq" to it. We then perform our clustering as indicated by lines 6-11, in which we compare pattern sequences for every cluster with every cluster, and if a match is found, we append the cluster nodes in the present matching cluster to the matched cluster.

The result is that we obtain a set of clusters, each having definite pattern information for its set of nodes, say like $C_1 = \{N_1, N_2, N_3\}$, $C_2 = \{N_5, N_7, N_8\}$, ...where each cluster will have one or more nodes in it, such that all nodes in that particular cluster have a common mobility pattern. Then the Multicast tree building process starts with this cluster information.

Multicast Tree Building:

In this phase we construct a multicast tree for selectively routing data along the highest probable path for the mobile nodes. It follows the procedure shown below. We also discuss algorithms that are used for implementing this procedure.

We present two algorithms: one for updating the mobility sequence/pattern and another for building the multicast tree.

Procedure: MulticastTreeBuilding
For each Cluster C found after Clustering
1. Consider a graph with these nodes alone present and the coordinator.
2. Assign weights to the links based on the movement pattern.
3. Build Multicast tree using ZTG phase.

End for
End Procedure

We now consider the algorithm for updating the mobility pattern for each node. This algorithm is implemented at each mobile node. Here "movSeq" is a vector quantity containing the list of routers a mobile node visits.

ALGORITHM 2: UpdateMobilityPattern

Input: moveSeq, position of node
Output: updated moveSeq
Begin
1. for all nodes N_i, where $1 \leq i \leq n$
2. if $moveSeq_i$!= null
3. $moveSeq_i \leftarrow moveSeq_i$ + current position of node
4. else
5. $moveSeq_i \leftarrow$ initial position of node
6. end if; end for

End

In the above algorithm, we compute moveSeq i.e., mobility sequence/pattern for every mobile node. After every fixed time interval the mobility pattern is updated for all mobile nodes. Then the next task is to build a multicast tree. The multicast tree construction algorithm involves a series of tasks like getting the mobility pattern (algorithm 2), creating sub-graphs based on edge weights and applying ZTG algorithm for each sub-graph created. Then algorithm BuildTree is invoked at the Coordinator in the Zigbee cluster tree network.

ALGORITHM 3: BuildTree

Input: Cluster trees, movSeq for all nodes in clusters
Output: Zigbee Multicast Mobility Trees
Begin
1. for all Clusters C_i, where $1 \leq i \leq n$, do
2. $E1 \subseteq E$ in which Cluster(i) edges are present.
3. subgraph = (Cluster(i).nodes, E1);
4. ZTG(subgraph);
5. end for

Initially for all clusters we get the updated mobility pattern information. After getting this information, we then create edge subsets $E1 \subseteq E$, and then we produce sub-graphs, each consisting of the corresponding cluster's nodes and its specific edges that are used to connect all nodes within that cluster. These individual sub-graphs are processed though the ZTG phase algorithm to yield the set of multicast mobility trees.

The worst case time complexity for the MobilityBasedClustering algorithm is $O(n^2)$, and worst case time complexity of BuildTree algorithm is $O(n^2)$ and for UpdatingMobilityPattern algorithm it is $O(n)$.

The above algorithms are used to produce the network tree structure in Zigbee, containing one or more multicast mobility trees with each tree group capable of independently and individually routing data packets to and from the mobile nodes and corresponding routers.

This ensures timely delivery of messages to the mobile nodes, irrespective of whichever cluster they may belong to. We also notice that in the algorithm for building the multicast tree we employ the ZTG phase algorithm that was present in our existing system, to get a set of sub-trees wherein each of the trees produced, have a mobility robustness value associated with them. We now have a brief overview at the system architecture for the proposed approach, as shown below:

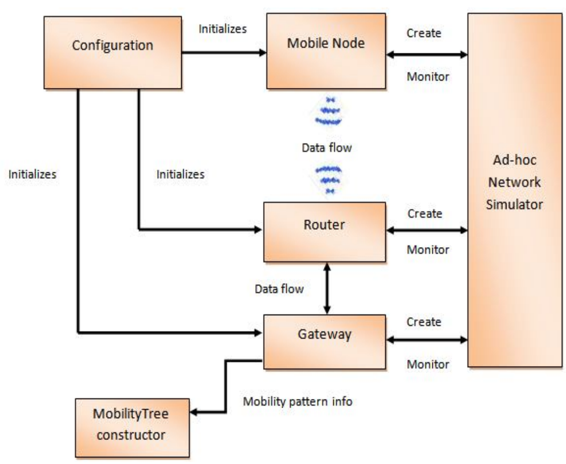


Figure 5: System Architecture.

These modules work as follows:
User can configure the number of nodes in the simulation and their mobility movement. Mobile Nodes are created by Configuration module. Mobile Node uses the Ad-hoc Network Simulator to communicate with other Mobile Nodes. Mobile Nodes move from one router to another. Gateway Node keeps track of mobility pattern and constructs the mobility tree. It uses the mobility tree to deliver

the messages to the nodes i.e., messages are delivered along the mobility tree. Emphasis is laid on the clustering technique used in the network.

The routing protocol used for this purpose is AODV (Ad hoc On-Demand Distance Vector) routing protocol which is dynamic in nature and performs incremental data transfer during regular time intervals, and has very low delay. So, node-to-node interactions follow a dynamic robust approach. These qualities make AODV an optimal protocol for our tree construction.

Prior to forming a cluster-tree topology, the three system configuration parameters must be determined which are: the maximum number of children of a router/coordinator (C_m), the maximum number of child routers of a router (R_m), and the depth of the network tree to be constructed (L_m).Once after creating a network, we then try to cluster the network. In the previous approach, the mobility tree was constructed based on the overall movement of the mobile nodes from router to router, and there was only one Mobility–robust tree created for the entire network.

Instead, as we have seen in our approach we can cluster the nodes based on the movement pattern and can construct a mobility tree for each cluster identified. This means that for every identified node mobility pattern, there is a corresponding mobility-robust cluster tree connected to the Zigbee Coordinator node. This technique results in higher throughput, energy efficient, reliable and robust tree structure for data delivery in the Zigbee networks.

6. SIMULATIONS AND RESULT

NS2 simulator is used to develop the environmental setup with a set of parameters as shown below:

Parameter	Settings
Channel type	Wireless Channel
Network standard	IEEE 802.15.4
Radio propagation model	Two-ray ground
Antenna	Omni Antenna
Media access control	CSMA/CA
Frequency	2.4GHz ISM Band
Data rate	250Kbps

Table 1: Parameter settings in NS2

The input for the network simulator is stated above. The simulation tool used is NS 2.34. It provides support for tasks like simulation of TCP, routing and multicast protocols. The simulator is written in C++ and script language is OTcl. The user writes the script in OTcl to define a network (nodes and links), the traffic in the network (source, destination and type of traffic) and which protocols it will use. The results of the simulations are an output trace file that can be used to data processing and to visualize simulations with a program called Network Animator (NAM) [9, 10].

The networking operation is illustrated through suitable diagrams and graphs. They are as follows:

In the first screenshot Figure 5 we see a Zigbee cluster tree having all the three node types namely: Coordinator, Routers and Mobile Nodes. Coordinator node (at the top) is followed by a matrix of routers nodes and mobile nodes moving along the routers. We can see the mobile nodes in the router set matrix moving in different directions. This is the configuration phase where we analyze the different node movements, and the distinct data patterns are recorded. This data is then used for performing clustering and subsequently data delivery.

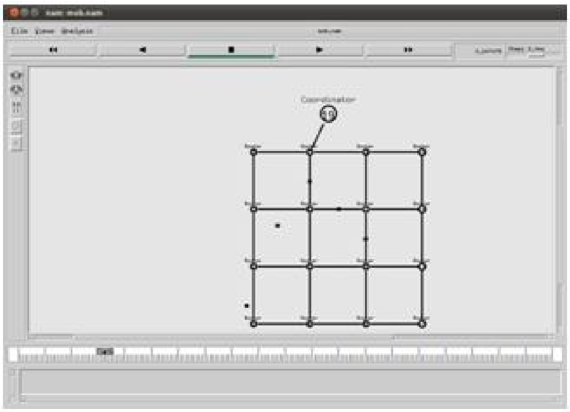


Figure 6: Screenshot of ns2 simulation showing mobile node movements

In the second screenshot Figure 6, we see an actual tree construction. The constructed tree (shown in red) represents the maximum-mobility robust multicast tree. Data transfer is also shown. Here only one tree is shown because the data patterns collectively follow the same route in this example.

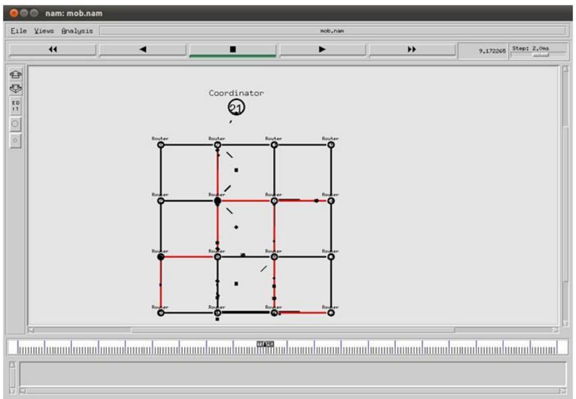


Figure 7: Screenshot of ns2 simulation showing multipath path (mobility-robust) tree construction

Based on the above NS2 simulation settings, we can compare the performance metrics of our proposed algorithm E-MRZT (Extendable MRZT) with that of the existing base MRZT algorithm and the previous

ZTG algorithm that was developed for Zigbee trees before the MRZT algorithm.

The relationship between the “packet delivery ratio” vs “mobility speed” and “packet delivery ratio” vs “number of mobile devices” is shown in the following graphs.

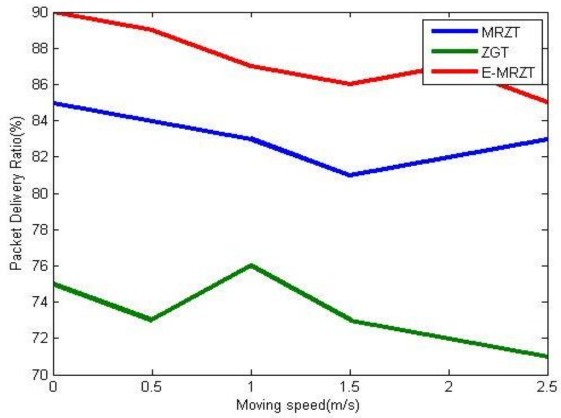


Figure 8: Packet delivery ratio vs Moving speed

This graph indicates that, when it comes to packet delivery our E-MRZT algorithm gives the highest outcome in terms of data delivery with respect to the moving speed of mobile nodes, when compared to the existing algorithms.

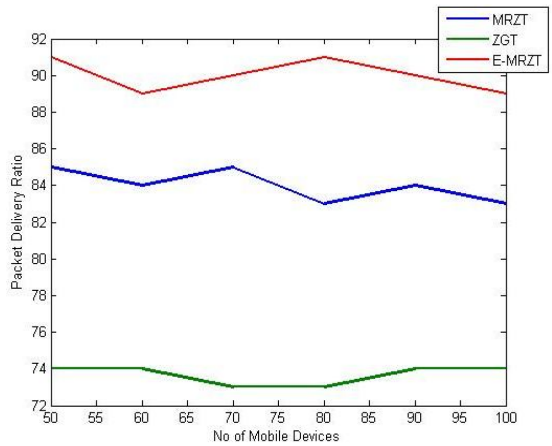


Figure 9: Packet delivery ratio vs No of mobile nodes

In this graph, we observe that E-MRZT algorithm has a higher data packet delivery ratio with respect to the number of mobile nodes, in comparison to existing algorithm. Thus, we have successfully proved that our proposed algorithm is much more efficient than the previous two algorithms.

7. CONCLUSION

The clustering mechanism has been proposed in this paper to obtain a much higher efficiency in the construction of Zigbee trees. By using this method, we can create a tree topology that can consist of several clusters, with each cluster capable of individually routing the data packets to their respective mobile nodes through the routers in a distributed manner. We have proposed a new Zigbee tree construction algorithm called E-MRZT or “Extended Mobility Robust Tree Construction “, with the aim of increasing the data delivery ratio to mobile nodes and be more resilient to node movements in Zigbee wireless networks. From the simulation experiments it is found that this method is found to be more efficient in its performance when compared to the previous approaches.

REFERENCES

[1] Wei-Ho Chung and Pi-Cheng Hsiu, Yuan-Ya Shih, Ai-Chun Pang, Yu-Kai Huang, and Kuan-Chang Hung, “Mobility-Robust Tree Construction in Zigbee Wireless Networks”, IEEE conference paper 2011.

[2] Wikipedia (http://www.wikipedia.com)

[3] ZigBee Alliance (http://www.zigbee.org) 2006.

[4] I.F. Akyildiz, J. Xie, and S. Mohanty. “A survey of mobility management in next-generation all-ip-based wireless systems”. In IEEE Wireless Communications, volume 11, pages 16–28, 2004.

[5] Y.K. Huang et al. “An integrated deployment tool for Zigbee-based wireless sensor networks”, In IEEE/IFIP EUC, volume 1, pages 309–315, 2008.

[6] I.F. Akyildiz, W. Su, Y. Sankara subramaniam, and E.Cayirci.“A survey on sensor networks.” In *IEEE Communications Magazine*, volume 40, pages102-114, 2002.

[7] Yu-Kai Huang, Ai-Chun Pang,Pi-Cheng Hsiu,Weihua Zhuang, and Pangfeng Liu’s”Distributed Throughput Optimization for Zigbee Cluster Tree Networks” IEEE transactions on parallel and distributed systems,2011

[8] B. An and S. Papavassiliou.“A mobility-based clustering approach to support mobility management and multicast routing in mobile ad-hoc wireless networks”. In *International Journal of Network Management*, volume 11, 2001.

[9] Teerawat Issariyakul and *Ekram* Hossain “Introduction to Network Simulator NS2” 2009. [11] Teerawat Issariyakul and *Ekram* Hossain “Introduction to Network Simulator NS2” 2009.

[10] De los Andes, Merida, venezuela and ESSI “NS simulator for beginners” Lecture notes: 2003-2004.

[11] C.E. Perkins and D.B. Johnson. Mobility support in IPv6. In *ACM MobiCom*, pages 27–37, 1996.

[12] P.C. Hsiu and T.W. Kuo. A maximum-residual multicast protocol for large-scale mobile ad hoc networks.In *IEEE Transactions on Mobile Computing*, volume 8, pages 1441–1453, 2009.

◈◈◈