Extending ZigBee Tree Routing Protocol for Resource-Constrained Devices

Uttam Kumar Roy
Dept. of Information Technology,
Jadavpur University, Salt Lake Campus,
Blok-LB/8, Sector-III, Kolkata-700098, India
u_roy@it.jusl.ac.in

***Abstract*—Although, ZigBee [21] provided a table-free Tree routing algorithm together with an addressing scheme for resource constrained IEEE 802.15.4 [15] devices, it is applicable for symmetric limited-sized tree networks only [6, 7, 8, 9].**

This paper has presented an efficient routing algorithm and a flexible, variable-length addressing scheme based on prefix code. The scheme eliminates routing tables and does not limit network size and also allows devices to have arbitrary number of children. It leverages simple mathematical and/or logical equations to take routing decisions and can be applied for almost all types of tree networks. Analytical and simulation results show that this flexible mechanism incurs very low overhead.

Keywords—Tree Networks, Address, Routing, Prefix-code, ZigBee.

I. Introduction

In the area of ubiquitous computing, IEEE 802.15.4 [15] is a remarkable standard. As this standard encourages low cost devices; hence resource-limited, a simple, lightweight, table-free tree routing algorithm was provided by ZigBee [21]. The advantage of this scheme is that it uses only mathematical equations to take routing decisions.

Due to its low-cost feature and its unquestionable success in **P**ersonal **O**perating **S**pace (POS), researchers are now trying to use it beyond POS, where networks are large and often asymmetric. However, the ZigBee Tree routing lacks the support for asymmetric, long and harsh networks.

Some of those issues have already been addressed by us in [6, 7, 8, 9] with potential solutions. In [6], we proposed a scheme to borrow addresses whenever addresses are exhausted in a specific place using the concept of mobile IP. Others are summarized in related work section.

In this paper, we have proposed variable-length addressing scheme based on prefix-code together with a routing algorithm. Note that the proposed scheme does not eliminate all the drawbacks of ZigBee Tree routing. Instead, it minimizes the existing problems. The proposed algorithm leverages the properties of prefix code and does not use any routing table. It takes routing decisions using only mathematical and/or logical equations. Analytical and simulation results show that the proposed scheme incurs very little overhead. The routing algorithm can be used for both symmetric as well as asymmetric and large networks. Moreover, it does not impose any restriction on the network diameter.

II. Related Work

Since 1970s, wireless networks have rapidly gained popularity. However, an investigation into low-cost, low-rate, low-power **P**ersonal **A**rea **N**etwork (PAN) is relatively new.

We investigated [20] that the routing protocols in ZigBee networks is essentially a combination of tree routing (for tree topology) and **A**d hoc **O**n-demand **D**istance **V**ector (AODV) routing (for mesh topology) with several optimizations done in consideration of stationary wireless topologies. In [20], we also developed a module for ZigBee PAN and inserted it in NS2 simulator to analyze and optimize both tree and mesh routing in a ZigBee network. We performed several experiments to study its various features, including: (i) the amount of routing packets generated, (ii) the packet-delivery ratio, (iii) the number of hops taken by application packets from source to destination. The packet-delivery ratio is found to be excellent, although the number of hops taken is not least always. We make several interesting optimizations in the implementation to control the huge burst of control packets produced for route discovery.

In [6], we demonstrated a unified address borrowing scheme which can be easily applied to grow the network beyond 16 hops and overcome the address exhaustion problem by borrowing address. A routing algorithm was also proposed based on mobile IP.

In [7], we extended the Tree routing proposed by ZigBee for the networks to be harsh and asymmetric. The method uses a modified expression for $C_{skid}(d)$ to assign addresses.

In [8], we provided a unified multi-channel routing scheme which can be easily applied to tree network so that the network can be used in dynamic application space and overcome the link disruption problem by multi channeling but without adding any extra overhead of having a routing table.

In [22], The authors also provides a Neighbor Table Based Shortcut Tree Routing in ZigBee Wireless Networks. However, since its extends the ZigBee Tree routing, the problem of network depth remains there. Moreover, it works on only symmetric tree networks.

Performance evaluation of routing protocols is a challenging issue and requires a well-planned, reliable and if possible standardized test bed. The VINT project is a landmark in the arena of network simulation research. The concept of split-level programming, its merits and applicability are discussed in [12].

In [5], authors explored the complex behavior of a large number of low-power sensor nodes. Energy aware operation of wireless devices is the dominant theme in [16], [5].

A fascinating topic of research for long is routing in wireless networks. These routing protocols deal with the challenges of wireless networks, namely low-bandwidth, high error-rates and often energy and memory constraints. They are either table-driven (e.g. DSDV [2], WRP [18]) or source-initiated, that is demand-driven (e.g. DSR [4], AODV [1], [3]). A comprehensive survey of these protocols has been done in [11].

We found that there is relatively scant literature on 802.15.4/ZigBee although its applications have been discussed in [17, 10]. Authors provide one of the first studies of the MAC sub layer while the recent paper [15] is a comprehensive performance evaluation of 802.15.4.

III. LIMITATIONS OF ZIGBEE ADDRESS ASSIGNMENT

ZigBee distributed address assignment scheme has several limitations. We already have addressed those issues in [6, 7, 8, 9] and provided several potential solutions. Let us quickly review those problems so that we can appreciate the need of a new scheme.

ZigBee specified that the "*coordinator determines*" important network parameters C_m, R_m and L_m. However, it remains silent about how to determine them. Since, in advance, we have very little or no idea about when, where and how many devices will come, it is almost impossible to find favorably good values of these parameters

Note that improper values of C_m and R_m may result wastage of network addresses. This happens since all routers use same C_m and R_m and once chosen they remain fixed. Given those parameters, essentially a virtual tree network skeleton gets created. Devices may sit in an empty location and use its address. So, even if addresses are available, it is possible that a device is unable to join to a desired position.

The size of unused address sub-block may be very large for larger value of L_m. Consider C_m=4, R_m=2 and L_m=14. The number of addresses, a router R at depth 1 can distribute to each of its 2 children routers is $C_{skip}(1)$ [=16381]. Because *R* can have maximum 2 (two) end devices and 2 (two) routers children, total number of addresses it can distribute is 2+2*16381(=32764). If *R* has no children (this may happen when no device is within the transmission range of R), large number (32764) of addresses that *R* could have distributed would remain unused. This implies straight wastage of approximately 50% of the total 2^{16} (=65536, for 16-bit address) possible addresses.

This situation becomes even worse if we attempt to form networks in mines, road-side etc. For C_m=4 and R_m=3, it can be shown [see Eqn. 1] that, the maximum value of L_m is 9. This means no device can exist at depth beyond 9.

Another major problem is that the addressing scheme essentially limits network depth. $C_{skip}(0)$ is the address sub-block being distributed by coordinator (at depth 0) to each of its R_m routers [Eqn 1]. So total number of addresses distributed to all of its routers is $R_mC_{skip}(0)$. Then, total possible addresses is the sum of $R_mC_{skip}(0)$ and number of end devices E_m(=C_m-R_m) of the coordinator and 1(one) for its own address. Because network address is a 16-bit address, assuming $R_m>1$, the following equality must be valid:

$$R_m C_{skip}(0) + (C_m - R_m) + 1 \le 2^{16}$$

$$\Rightarrow R_m \frac{1 + C_m - R_m - C_m R_m^{L_m - 1}}{1 - R_m} + (C_m - R_m) + 1 \le 2^{16}$$

$$\Rightarrow \frac{1 + C_m - R_m - C_m R_m^{L_m}}{1 - R_m} \le 2^{16}$$

$$\Rightarrow L_m \le \frac{\log_{10} \frac{(2^{16} - 1)(R_m - 1)}{C_m} + 1}{\log_{10} R_m} \tag{1}$$

For example if $C_m = 8$ and $R_m = 4$, maximum possible depth $L_m = 7$ only. This proves the impossibility of creating very long and harsh networks.

IV. PROPOSED SCHEME

The primary requirement of ZigBee devices is that they should to be very less costly and low-power consuming devices. As a result they become resource-constrained. So, a simple light-weight routing algorithm is needed that can run on these resource-constrained devices efficiently. The proposed routing algorithm eliminates routing tables altogether. So, it requires very small memory to run. Since, it requires less memory, it is power-efficient as well. It also eliminates the overhead of placing routing information in the packet which is done by the source-initiated routing such as DSR. The proposed routing algorithm uses prefix-code-based variable length addressing scheme which is described in the following sections.

A. Assigning Network Addresses

The proposed scheme assigns network addresses to devices intelligently so that a route to a destination device can be determined from only the destination address. Consider the tree shown in Fig 1. for illustration. The addresses are calculated as follows:

Every router in the tree labels each of its outgoing links (if any) by a (locally) unique binary number.

If a router *R* has C_R number of children, minimum number of bits $N(C_R)$ required to label each outgoing link of *R* is:

$$N(C_R)=\begin{cases} C_R & \text{if } C_R=0 \text{ or } C_R=1 \\ \lceil \lg(C_R) \rceil & \text{if } C_R>1 \end{cases} \quad (1)$$

The unique network address of each node in the tree is then calculated as follows:

The address of root (coordinator) is always *1*. The address A_D of any other device *D* is obtained by concatenating its parent's address and its id (label of the link). For example, the address of the router *R8* in Fig 1. $A_{R8} = \underbrace{1011}_{parent\,address}\ \underbrace{01}_{id}$.

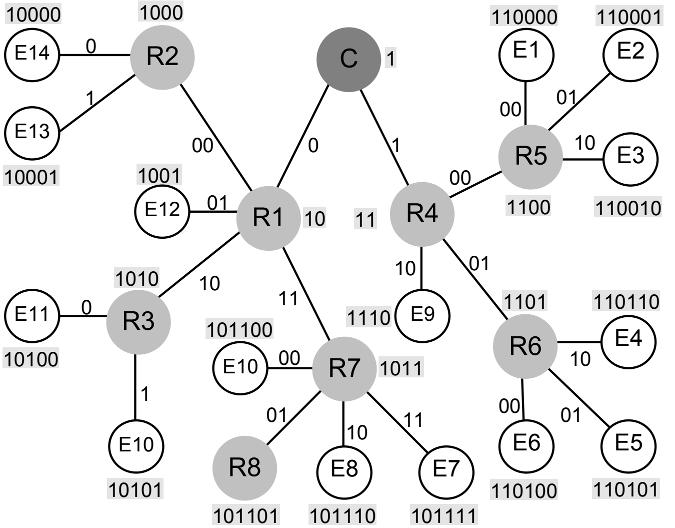


Fig. 1. Network address assignment

Here, 1011 is the address of *R7* and 01 is the label of the link from *R7* to *R8*. The addresses of other devices are calculated accordingly. This scheme has following important properties:

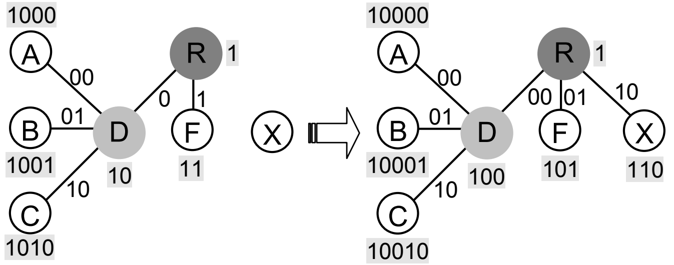
- The addresses, obviously, are always unique.
- Address of a leaf node can never be a prefix of another leaf node (*prefix property* for leaf nodes).
- Siblings have common prefix (parent's address).
- The address of every node has ancestor's address as its prefix.

Last property is very interesting and we use it in our routing algorithm to avoid routing table.

B. Restructuring

One of the problems of this scheme is that number of bits required to label links may change from time to time as devices join to or leave from the network. Let us consider Fig 2(i).

The number of bits $N(C_R)$ required to label each link of *R* is 1 as number of children of router *R* (i.e. C_R) is 2 [See eqn 1]. Now, if another device *X* joins to *R*, $N(C_R)$ will become 2. Consequently, each of the outgoing links (including the new link) of *R* must be re-labeled by 2 bits. This implies addresses of all existing descendants of *R* must be recalculated [Fig 2(ii)]. We call this procedure as *restructuring*, which incurs an overhead. However, we shall show (analytically as well as by simulation results) that overhead due to it is considerably low. Kindly note that this paper does not remove all problems of ZigBee Tree routing. It minimizes them.



(i) Before (ii) After

Fig. 2. Network Addresses "Restructuring".

C. The Routing Algorithm

In the following section, we shall describe how this prefix-based address can be used to find a route to a target device. The following notations are used:

A_i	:	Network **A**ddress of device i,
B_i	:	Number of **B**its in A_i,
C_i	:	Number of **C**hildren of router i,
ID_i	:	Local address of device i (Except root node)

Note that the prefix-code has an interesting property—"*The address of every node has ancestor's address as its prefix".*

Let us understand it clearly first. Consider the sequence of nodes, *V→W→X→Y→Z*. Then the address A_Z of node *Z* will be of the following form:

$$\overbrace{\underbrace{\underbrace{\underbrace{\underbrace{10011}_{A_V}\underbrace{1001}_{ID_W}}_{A_W}\underbrace{0101}_{ID_X}}_{A_X}\underbrace{0110}_{ID_Y}}_{A_Y}\underbrace{0110}_{ID_Z}}^{A_Z}$$

This implies that, if the address A_X of a node *X* is a prefix of address A_Y of another node *Y*, then *Y* must be a descendant of *X*. In other words, X must be ancestor of Y. So if *X* gets a packet destined to *Y*, the routing decision can be made using this information [Fig 4].

Suppose, a source node *S* having network address A_S wants to send a packet to a destination node *D* having network address A_D. Consider at any point of time, an arbitrary node *X* has received the packet, which has to be delivered to *D*.

$$\overbrace{\underbrace{10011}_{B_X}\underbrace{010}_{N(C_X)}110...101}^{B_D} \qquad \overbrace{\underbrace{10011}_{B_X}\ \underbrace{101}_{N(C_X)}}^{B_D}$$

(i) (ii)

Fig. 3. Deciding relation between parent X and descendant D (i) D is not immediate child of X (ii) D is immediate child of X

Algorithm: When X receives a packet destined to *D*, it checks, whether its own address (A_X), is a prefix of destination address (A_D). If not so, destination *D* is not a descendant of *X* and in this case *X* has nothing to do except forwarding the packet to its parent, which will in turn take care of rest of the routing decisions. Otherwise (i.e. *X*'s address is a prefix of destination address), destination MUST be a descendant (direct or indirect) of *X*. Two cases exist:

1) Number of bits in destination address (B_D) is exactly equal [Fig 3(i)] to the sum of number of bits in its own address (B_X) and number of bits required to represent its children [$N(C_X)$]. This means, destination address is just the concatenation of *X*'s address (A_X) and child ID [Fig 3(i)]. In this case, destination *D* is a direct child of *X*.
2) Otherwise [Fig 3(ii)], the destination is a descendant but not direct child.

Both the cases, next hop device ID can be obtained as follows:

-- Start from MSB of destination address A_D;
-- Ignore first B_X bits of A_D.
-- Next $N(C_X)$ bits from A_D constitutes next hop device ID.

Note that the algorithm described above uses only mathematical and logical calculations and thus eliminating the need of routing tables.

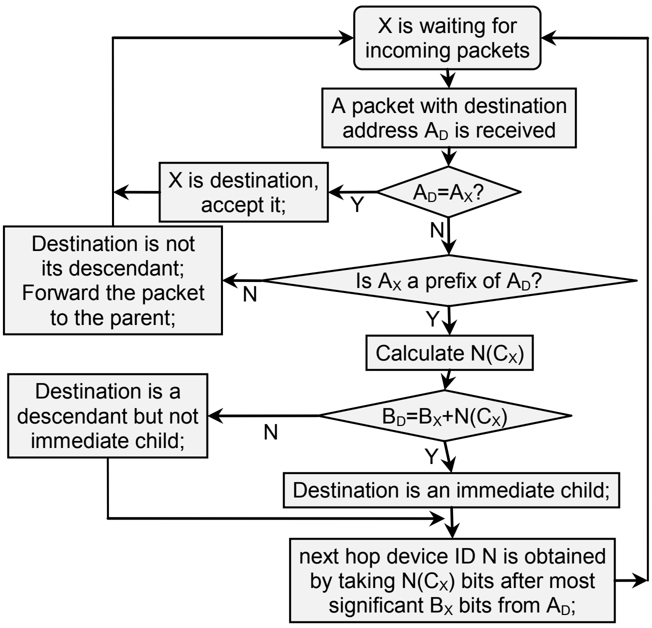


Fig. 4. Proposed routing algorithm

D. Example Scenario

To understand how routing takes places, consider the network in Fig 1. Suppose end device *E1* (A_{E1}=110000) wants to send a packet to another end device *E11* (A_{E11}=10100).

Since, 110000 is not a prefix of 10100 (neither can it be as the source is an end device and an end device can not have children), it simply forwards the packet to its parent *R5*. *R5* and *R4* perform similar steps as *E1* and the packet eventually reaches *C* having address 1. Now *C*'s address (1) is a prefix of 10100. Since, C_C=2, *C* extracts $N(C_C)$ (=1) bit after B_C (=1) bit from A_{E11}(=10100) and gets 0. *C* forwards the packet through the link labeled 0 and the packet reaches to *R1* having address 10. *R1* and then *R3* perform exactly the same way and the packet eventually reaches to its destination *E11*.

V. Calculation of Overhead

In this section, we shall calculate the amount the overhead due to "restructuring" processes.

Note that ZigBee networks are primarily intended to be static. Devices gradually come and join to form a network. Once the devices are joined, they hardly move or leave. For example devices a temperature sensor node on a light post, nodes attached to a ceiling fan, tube light etc. do not move. Such a case, even if "restructuring" occurs, it only occurs during the network formation. Once the network is formed, it does not happen anymore. Furthermore, it apparently seems that "restructuring" incurs significant overhead. However, fortunately, it occurs infrequently; thus average number of nodes affected per restructuring is significantly small. Let us understand it in the following:

Note that the restructuring is required only when the number of children changes from 2^n to 2^n+1 or 2^n+1 to 2^n (n = 1, 2, 3,…) whenever a device joins to or leaves from network respectively. In the former case, future $2^{(n+1)}-2^n$ times, and in the later case future $2^n-2^{(n-1)}$ times no restructuring will occur. On an average, if a router has 2^n number of children, (n-1) number of cases restructuring will occur. For example, if a router has 8 (2^3) children restructuring occurs two times (one when number of children changed from 2 to 3 and other when number of children changes from 4 to 5). Table 1 shows the relation between number of children of router and number of restructuring on that router.

So, a fraction of $(n-1)/2^n$ cases restructuring will occur. For a moderate value of n, this factor is very small. So, overhead due to this is negligible.

TABLE I. Effect of No. of Children of a Router on No. of Restructuring

No. of children	No. of restructuring	No. of children	No. of restructuring
0-2	1	33-64	6
3-4	2	65-128	7
5-8	3	129-256	8
9-16	4	257-512	9
17-32	5	513-1024	10

Let us now analyze the total number of "restructuring" needed considering all routers. Consider the following parameters:

D: Total number of devices at a particular time.
R: Number of routers
C (D-R): Number of end devices

Each router has on an average D/R number of children. The number of restructuring needed for each router is $\log_2(D/R)-1$. The total number of restructuring is then N = (log2(D/R)-1)R.

If we plot N with respect to R. it looks like as shown in the Fig 5.

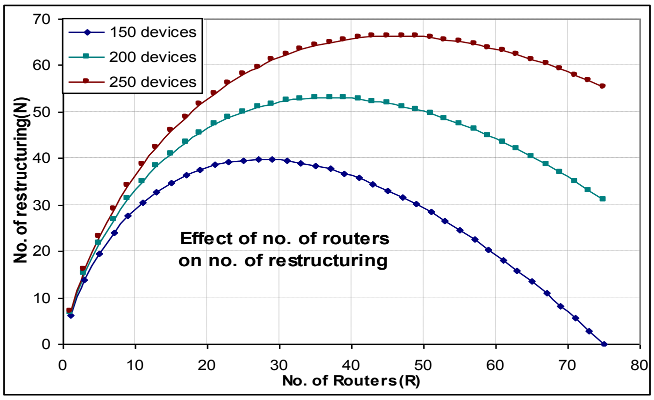


Fig. 5. Analytical result: Effect of number of routers on number of restructuring

This shows that for small number (<10) of routers, the overhead is around 3% to 10% (for 250 devices). For large number of devices, the overhead is also small. The graph also helps us to estimate the number of routers to have minimum overhead. In addition to this, remember only those nodes that are descendant of a router are affected by restructuring procedure. Moreover, for a static wireless network, joining to and leaving from network will occur during the network formation phase. So, once a network is set up, there is practically no overhead. So, average number of address update per restructuring procedure is small. As a consequence, overall overhead is expected to be low.

VI. SIMULATION RESULTS

In this section, we shall show how the proposed tree routing algorithm outperforms than the existing tree routing algorithm specified by ZigBee alliance.

A. Simulation setup

We developed a Java-based simulator to perform the experiment. The simulation area is taken as 1360x640 pixel area. According to the ZigBee specification, the coordinator forms a network consisting of itself and starts beaconing. A number of routers and end devices are then added in the simulation area. The devices get beacons from nearby routers and join the network by sending a JOIN_REQUEST packet. Routers also start beaconing after joining the network. The network was formed according to the ZigBee Specification.

B. Observations

The experiment was carried out for 150, 200 and 250 number of devices. For each case, the number of router was varied from 1 to 70. The scenario is repeated over 2000 times and average result was obtained.

Fig 6. shows the effect of number of routers on the number restructuring needed. The simulation result, as expected, is very similar to the result as obtained from equation 9. It can be stated from the graph that fraction of numbers, the restructuring occurs (for 250 devices) is small with maximum about 23% cases.

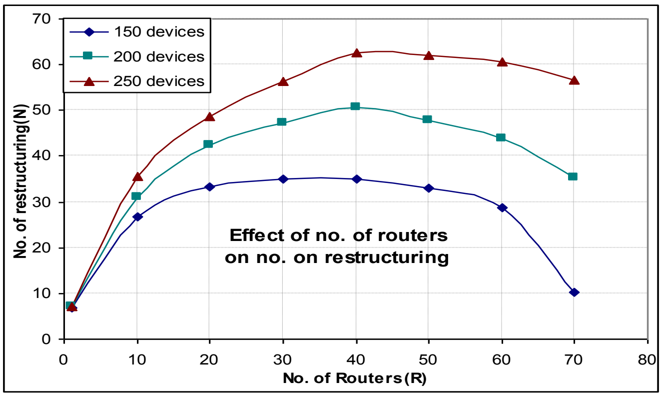


Fig. 6. Simulation result : Effect of number of routers on number of restructuring

From Fig 6., we can also find the relation between the numbers of routers with the average number of nodes affected per restructuring. It shows that even if restructuring occurs, number of nodes affected per restructuring is significantly less. It is observed that only 6-10 numbers of nodes are affected per restructuring most of the cases which is a quite acceptable figure.

The general observation is that for relative small number of routers (note that number of routers compared to number of end devices in a ZigBee network is really very small), overhead due to restructuring is almost negligible. Moreover, for static wireless networks (which was the primary motivation of ZigBee networks), once the network is established and stabilized, there is no further overhead. Furthermore, routing is very cost-effective in term of memory requirement as it is routing table free and does not incur any overhead on packet.

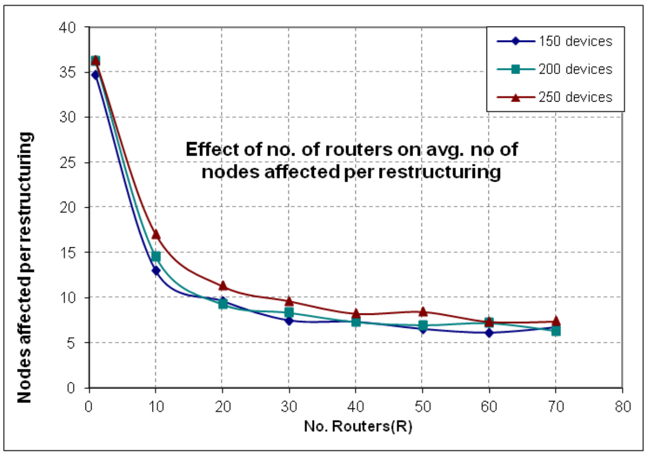


Fig. 7. Simulation result : Effect of number of routers on number of nodes involved in restructuring

Fig 7 shows that effect of number of routers on number of nodes involved in restructuring. It indicates that although restructuring occurs, a very few nodes suffers from this restricting process. So, overhead due to the restructuring process is negligible in practice.

VII. CONCLUSION

In this paper, we proposed a new routing algorithm together with an addressing scheme for IEEE802.15.4 tree networks. Each device is assigned a unique binary address cleverly so that routing decision can be made only from destination address. The proposed algorithm does not need any routing table to be maintained by each router. Algorithm can still deliver the packet to the proper destination. We have also shown that the overhead for this routing algorithm is minimal.

[1] C. E. Perkins, E. Belding-Royer, and S. R. Das, "Ad hoc On-Demand Distance Vector (AODV) Routing", http://www.ietf.org/rfc/rfc3561.txt, July 2003. RFC 3561.

[2] C. E. Perkins and P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers", Proceedings of ACM SIGCOMM, 1994.

[3] C. E. Perkins and Elizabeth Royer, "Ad-hoc On-Demand Distance Vector Routing", Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications, New Orleans, LA, February 1999.

[4] D. B. Johnson, D. B. Maltz, "Dynamic Source Routing in Ad-hoc Wireless Networks", Mobile Computing, T. Imielinski, H. Korth, Eds. Kluwer Academic Publishers, 1996, ch. 5, pp. 153-181.

[5] D. Ganeshan, B. Krishnamachari, "Complex Behavior at Scale: An Experimental Study of Low-Power Wireless Sensor Networks", UCLA/CSD-TR 02-0013, UCLA Computer Science, 2002.

[6] Debabrato Giri, Uttam Kumar Roy, "Address Borrowing In Wireless Personal Area Network", Proc. of IEEE International Anvanced Computing Conference, (IACC '09, March 6-7), Patiala, India, page no 1074-1079

[7] Debabrato Giri, Uttam Kumar Roy, "Single Level Address Reorganization In Wireless Personal Area Network", 4th International Conference on Computers & Devices for Communication (CODEC-09), December 14-16, 2009, Calcutta University, India.

[8] Debabrato Giri, Uttam Kumar Roy, "Multi Channel Personal Area Network(MCPAN) Formation and Routing", International Conference on Industrial Engineering Science and Applications (IESA 2014), April 2-4, 2014

[9] Debabrato Giri, Uttam Kumar Roy, "WPAN Routing Using Huffman Technique", Journal of Engineering Sciences, Vol (2), No (2), February, 2013 Pages 49-61.

[10] Ed Callaway, Paul Gorday, Lance Hester, Jose A. Gutierrez, Marco Naeve, Bob Heile and Venkat Bahl. "Home Networking with IEEE 802.15.4: A Developing Standard for Low-Rate Wireless Personal Area Networks",IEEE Communications Magazine August 2002.

[11] Elizabeth Royer and C-K Toh, "A Review of Current Routing Protocols for Ad-Hoc Mobile Wireless Networks", IEEE Personal Communications Magazine, April 1999.

[12] Gang Lu, Bhaskar Krishnamachari, Cauligi S. Raghavendra, "Performance Evaluation of the IEEE 802.15.4 MAC for Low-Rate Low-Power Wireless Networks", IEEE International Conference on Performance, Computing, and Communications, 2004.

[13] IEEE 802.15.11 Standard: "Wireless Local Area Networks, 1999".

[14] IEEE 802.15.1, Wireless Medium Access Control (MAC) and Physical layer (PHY) specifications for Wireless Personal Area Networks (WPANs)

[15] IEEE 802.15.4, Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)

[16] J. Heidemann, W. Ye and D. Estrin."An Energy-Efficient MAC Protocol for Wireless Sensor Networks", Proceedings of the 21st International Conference of the IEEE Computer and Communications Societies (INFOCOM 2002), New York, NY, June 2002.

[17] Jianliang Zheng and Myung J. Lee. "A Comprehensive Performance Study of IEEE 802.15.4", http://www-ee.ccny.edu/zheng/pub, 2004.

[18] C. Schurgers, S. Park and M. B. Srivastava, "Energy-Aware Wireless Microsensor Networks", IEEE Signal Processing Magazine, Volume: 19, Issue: 2, March 2002.

[19] Shree Murthy, J. J. Garcia-Luna-Aceves, "An Efficient Routing Protocol for Wireless Networks", Mobile Networks and Applications, 1996.

[20] Uttam Kumar Roy, Debarshi Kumar Sanyal, Sudeepta Ray, "Analysis and Optimization of Routing Protocols in IEEE802.15.4" (Asian International Mobile Computing Conference (AMOC 2006), Jadavpur University, Kolkata, India

[21] ZigBee Alliance (ZigBee Document 02130r7) Draft Version 0.90: Network Specification, July 2004.

[22] Taehong Kim; Seong Hoon Kim; Jinyoung Yang; Seong-eun Yoo; Daeyoung Kim, "Neighbor Table Based Shortcut Tree Routing in ZigBee Wireless Networks," Parallel and Distributed Systems, IEEE Transactions on , vol.25, no.3, pp.706,716, March 2014.