Enhancing ZigBee Tree Routing to Support Large Pseudo-Linear Network Topologies

Uttam Kumar Roy
Dept. of Information Technology,
Jadavpur University, Kolkata, India
u_roy@it.jusl.ac.in

***Abstract*—IEEE 802.15.4 accordant devices are resource-restrained; thereby cannot execute conventional awkward routing algorithms. Even though, ZigBee provided a elementary routing algorithm for tree networks, it cannot manage long linear networks typically set up in roadside, mines, agricultural field, mountains etc.**

In this paper, we have nominated a source routing based light-weight routing algorithm that supports arbitrary long networks. The proposed algorithm especially suits for pseudo-linear networks. The algorithm takes the advantage that a pseudo-linear network has limited branching. Experimental results prove that this simple scheme exhibits excellent packet delivery performance.

Keywords—ZigBee; Psudo-Linear Networks; Addressing; Routing.

I. INTRODUCTION

In the area of computer networks, there exist many areas where network topologies are often linear or almost linear. Fig 1 depicts a few such as networks formed in mines, roadside, agricultural field, mountains etc.

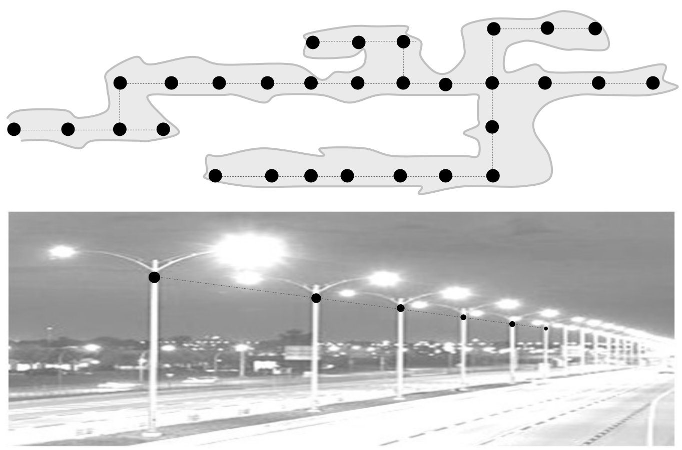

Fig. 1. Examples of psudo-linear networks: mine(above) and roadside (below)

Scientists are trying to apply the novel IEEE 802.15.4/ZigBee [15] technology to achieve low-cost solutions. It is a novel technology to bring ubiquitous networking into our lives. As it enables low-cost instruments; thereby resource-limited, ZigBee [15] provides a rudimentary and table-free routing algorithm as well as a scheme for addressing. However, it lacks to support very large and long and asymmetric networks.

We suggested [1, 2, 3, 4, 11, 12, 13, 14] many potential solutions to conquer the problems. This paper proposes a table-free, light-weight routing algorithm based on source routing that supports long networks. It especially fits in pseudo-linear networks where nodes typically have one or a few children nodes. The routing information is stored in sink and is affixed in the packet. This information is used by intermediate nodes that do not need and routing table. Consequently devices having limited resources can also run this algorithm. Experimental results demonstrate the correctness of the scheme.

II. RELATED WORK

We investigated [11] ZigBee routing protocol and found that it is basically a union of Tree-routing and AODV having few optimizations assuming static topologies. We also set up an simulator (NS2) for ZigBee WPAN to experiment and optimize tree as well as mesh routing. We recommended [1] a mobile IP based scheme that can easily be applied to eliminate the problem of address exhaustion by lending addresses from nodes having unused addresses. In [2], we enhanced the ZigBee Tree for unbalanced the network topologies. We recommended routing scheme for application space to be dynamic and overcome the problem of link disruption by multi-channeling however adding no extra overhead.

In [12, 13], we suggested a adaptable, variable-sized addressing scheme together with a new lightweight, table-free routing algorithm. They leverage properties of prefix code and allow instruments accepting any no. of children and does not also limit network depth. In [14], we enhanced ZigBee's C_{skip} which can handle asymmetric topologies which actual scheme cannot handle. Some applications of ZigBee have been discussed in [9, 5] and [8] was a rigorous comprehensive performance analysis of ZigBee/802.15.4.

III. ZIGBEE ADDRESS ASSIGNMENT LIMITATIONS

The way the ZigBee assigns addresses to devices has some shortcomings. In [1, 2, 3, 4, 11, 12, 13, 14] We already have marked those pitfalls and gave potential solutions. Here, we shall first recapitulate those drawbacks so that we can quickly understand the necessity of a better addressing mechanism.

According to specification, some important network parameters C_m, R_m and L_m are determined by the coordinator. However, it was not resoled how to determine them. Since, before setting up the network, we usually don't have much information of how many devices will come, when and where; it is difficult to predict favorably good values of C_m, R_m and L_m.

Inappropriate value of C_m and R_m may lead to wastage of device addresses as all router devices use identical C_m and R_m. Moreover, they are not even varied once they are chosen. In fact, a virtual skeleton of a tree network is formed. Instruments seat in an empty position and get its address. Hence, a device may not be able to join even though there are available addresses.

The number left-over addresses might be huge for larger L_m. Suppose, C_m=4, R_m=2 and L_m=14. Then a router device R (at depth 1) can provide 16381 [$=C_{skip}(1)$] addresses to each of its two router children. Since, R may accept two end device children and two router children, the total no. of addresses it can provide is 2+2*16381(=32764). If R does not have any children (i.e. there are no devices within the transmission boundary of R), huge number of addresses (32764) that R could have provided would remain un-utilized. This means there is a direct wastage of roughly 50% of the total 2^{16} (=65536, assuming address length is 16-bit) potential addresses.

The scenario can be even worse when we form road-side networks, mines etc. Consider C_m=4 and R_m=3. Then it can be proved [Eqn. 1] that, the L_m can have a maximum value of 9. This implies no instrument can join at a depth more than 9.

The prime drawback is that the addressing logic basically restricts network depth. Note that the addresses that can be provided by coordinator (at depth 0) to each of its R_m routers is $C_{skip}(0)$ is [Eqn 1]. So, $R_mC_{skip}(0)$ is the sum of all addresses provided to all of its routers. So, the total addresses is $R_mC_{skip}(0)$ plus the number of end devices $E_m(=C_m-R_m)$ of the coordinator plus one for its own address. For 16-bit network address and assuming $R_m>1$, the following inequality must hold:

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

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

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

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

Here is an example. If $C_m = 8$, $R_m = 4$, (1) shows that longest value of L_m is only 7. This implies that certain very long network are impossible to create.

IV. SUGGESSTED ALGORITHM

It may be notes that 802.15.4 is very less costly technology. So, scientists are attempting to use it virtually everywhere such as in mines, in smart cities for monitoring environmental metrics and traffic controlling, in agriculture, in remote places for volcano monitoring what's not. If observed critically, we can find some common characteristics for these kinds of networks:

- Networks are often linear or pseudo-linear [Fig. 1] in nature.
- Typically have a sink node (usually a full-fledge computer) and other nodes send sensed data (temperature, humidity, pictures etc.) to sink.
- Nodes are static and join and leave the network infrequently once a network is formed.
- Sink, occasionally sends data to other nodes (e.g. traffic control network)
- A non-sink node hardly sends data to another non-sink node.

Note that, ZigBee instruments are intended to be less costly and power-friendly. Consequently, they are supposed to have low memory, low computing power. Keeping these requirements and the above network characteristics in mind, we should device a simple routing algorithm that can run on those devices. For example, we can use a tree network where no extra routing algorithm is needed to send data from sensor node to sink. Each sensor node (having exactly one parent) forwards data to its parent and this step continues until the data reach to the sink.

However, if sink wants to send data to a sensor node (for traffic control network, say) a routing algorithm is required. The proposed algorithm makes use of the properties that a pseudo-linear network has limited branches; i.e. nodes have a very few (0, 1, 2 or 3) children. This property is intelligently used so that a packet can be routed to a destination based on small routing information included in the packet itself.

A. Network Formation

We propose to use tree networks. Although, tree networks have some inherent problems (only one path between node-pair), they have some advantages too. For example, no routing algorithm is necessary if non-root nodes want to send packet to the root. This typically happens in the networks shown in Fig. 1. However, root of these kinds of networks occasionally sends data to other nodes. Non-root nodes hardly send data to other non-root node. So, we form the tree networks as follows:

Each intermediate node marks all of its children with a locally unique binary string. The marking may be done in any order. One such tree is shown in Fig 2.

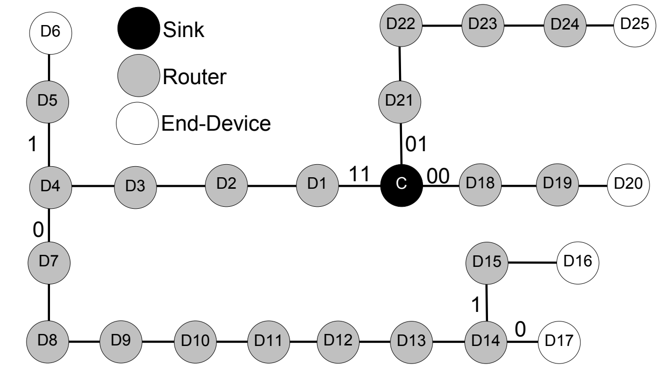

Fig. 2. Psudo-linear network formation

The node C has 3 children. So, it uses 2 bits to mark its outgoing links. Similarly, node D4 has two children. So, it uses 1 bit to label its outgoing links. Nodes having only 1 child (most nodes are of this kind) need not label the link. In general if a node D has C_D children, no. of bits $N(C_D)$ needed to mark each child of R is:

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

Since for a network like roadside-network, most nodes will be non-crossover nodes and C_D for such node is 1 and no bit is required for link labeling. There are a very few crossover nodes where C_D is limited and often ≤ 4 and $N(C_D) \leq 2$.

B. Routing

Once a pseudo-linear tree network is setup, sensor nodes send data to the sink. No routing algorithm is needed for this. Every node forwards packets to its parent and the packet eventually reaches to the sink. When a packet traverses from a sensor node to sink, intermediate nodes *append* link label to the packet. For example, the packet received by sink C from D15 will have following routing string:

Link labels

Routing string = 1 0 11

where first 1 is appended by node D14, next 0 is appended by D4 and last two bits 11 are appended by C itself. Sink keeps this information in a table. This process continues and eventually sink will have routing strings for all nodes which look as follows:

D_1, D_2, D_3, D_4	D_5, D_6	$D_7, D_8, D_9, D_{10}, D_{11}, D_{12}, D_{13}, D_{14}$
11	111	011

D_{15}, D_{16}	D_{17}	D_{18}, D_{19}, D_{20}	$D_{21}, D_{22}, D_{23}, D_{24}, D_{25}$
1011	0011	00	01

Since, sink is typically a full-fledge computer, it has enough memory to store this information. This table contains only total 20 bits of routing information and 25x (size of device address in bits) bits device addresses which can further be reduced further. For example, the previous table can be restructured as:

D_1-D_4	D_5-D_6	D_7-D_{14}	D_{15}-D_{16}	D_{17}	D_{18}-D_{20}	D_{21}-D_{25}
11	111	011	1011	0011	00	01

This table contains only 13 network addresses. Let us have an estimation of amount memory required in practice.

Consider a city with 10x10 km area with roads laid as a grid separated by 100 meters apart. The transmission range of a ZigBee device is 20 meters (say). So, there will be approximately 50x100x100=500000 devices. If the network is formed carefully and the sink is placed in the centre of the city, average length of label string will have very few bits. So, sink needs a few MB of memory which even a very old computer possesses.

Algorithm:

If D receives a packet from a child, D blindly forwards it to its parent. If D receives a packet targeted to T from its parent, *D* checks, if T is D and if so, D accepts it. Else T is a descendant of D. We can obtain the next hop device ID as follows:

```
If D has single child, send the packet to the child.
Else {
        Get routing string from the packet.
        Remove least significant N(C_D) bits from routing string
        Put the modified routing string in the packet
        Send packet along the link labeled with those N(C_D) bits
}
```

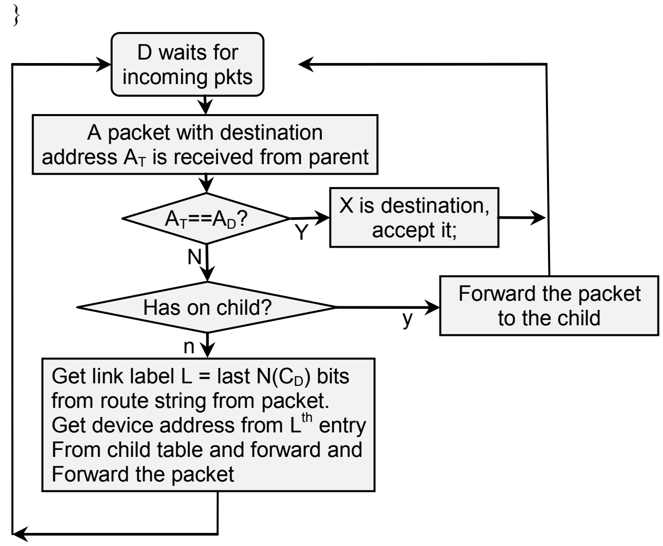

Fig. 3. Routing algorithm

C. Sample Scenario

We shall consider the network in Fig 2 to demonstrate how routing takes place. Consider the coordinator sends a packet to the end device D_{16}. The sink looks up the table and finds that the routing string for device D_{16} is 1011. Since, it has 3 children, it extracts least significant N(3) [=2] bits from 1011 which happens to be 11. It puts the remaining string 10 in the packet and forwards it through the link labeled with 11. The packet comes to device D_1.

Since D_1 exactly one child, it forwards the packet to D_2. Devices D_2, D_3 do the same thing and the packet reaches to D_4. Since, it has 2 children, it extracts least significant N(2) [=1] bits from 10 which happens to be 0 and puts the remaining string 1 in the packet and forwards it along the link marked 0. The packet comes to device D_7. Device D_7 to D_{13} do as D_1 did and pkt eventually reaches to D_{14}. Since, it has 2 children, it extracts least significant N(2) [=1] bits from 1 which happens to be 1. It places the remaining null string in the packet and forwards it along the link marked with 1. The pkt comes to device D_{15} which as one child. The device forwards the packet to it child and the pkt finally comes to the target D_{16}.

D. Restructuring

The no. of bits needed to mark links might vary if new roads are laid down. We shall take Fig 4 for demonstration.

As the no. of children of *R1* (= C_{R1}) is two, the no. of bits $N(C_{R1})$ needed to mark each child of *R1* is 1. Now, if a new road is laid [i.e. *X* wants to join to *R1*], $N(C_{R1})$ will be 2. As a result, all children (including X) of *R1* must be re-marked by 2 bits. It results that routing information for all descendants of *R1* must be updated to the sink. This procedure is called *restructuring*, which results an extra overhead. Since, roads are laid down very in-frequently, overhead incurred by this is too small.

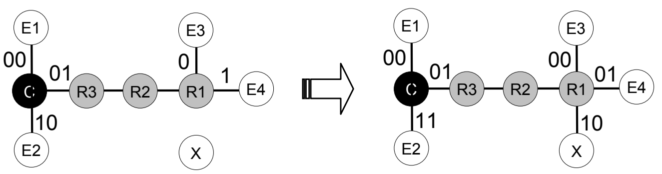

Fig. 4. "Restructuring" Links (left) before (right) After

The restructuring may be done by sending a single update message to sink. The sink extracts routing string RS from this packet and updates the table by substituting all routing string of the form RSX by RS0X. The entry for X is added when sink receives any data from X.

R_1,R_2,R_3	E_1	E_2	E_3	E_4	=>	R_1,R_2,R_3	E_1	E_2	E_3	E_4	X
01	00	10	**010**	**01**1		01	00	01	**010**0	**010**1	0110

Fig. 5. Route update by sink node for X (left) Before (right) After

An example is shown in Fig. 5

V. Overhead Calculation

Let us now estimate the overhead incurred by the "restructuring" phenomena. It may be noted that ZigBee devices supposed to be stationary. Specification says devices join from time to time. Once a network is formed, devices hardly move or leave. Example includes temperature sensor nodes attached on light posts, nodes attached to a ceiling fan, tube light etc. In these cases "restructuring" only occurs when the network is formed. Thereafter, it does not occur further. So, although it seems that "restructuring" results much overhead; fortunately, it occurs very unlikely. Hence average no. of nodes affected per restructuring is very very small. Consider the following examples to understand it better:

A clever observation results that the restructuring is only needed when a device either joins to or leave from the network. More explicitly, it happens if the no. of children of a router changes either from $2^n+1 \rightarrow 2^n$ or $2^n \rightarrow 2^n+1$, for n = 1, 2, 3,… No restructuring will occur, for future $2^{(n+1)}-2^n$ times (later case), and future $2^n-2^{(n-1)}$ times(former case). In general, only n-1 no. of times restructuring will happen for a router having 2^n no. of children. Here is an example. Consider a router has 2^3(=8) children. Restructuring happens 2 cases (i) when no. of children varies from 2 to 3 and (ii) when no. of children varies from 4 to 5). The exact relation between no. of children of router and no. of restructuring occurs is shown in Table . This demonstrates that, a very small fraction [= $(n-1)/2^n$] of times restructuring happens. So, for even a small value of n, this fraction is too small. Hence, extract overhead by restricting is almost negligible.

TABLE I. Relation between no. of children of a router and No. of restructuring

Number of child	Number of "restructuring"	Number of child(contd.)	Number of "restructuring"(contd.)
0 to 2	1	33 to 64	6
3 to 4	2	65 to 128	7
5 to 8	3	129 to 256	8
9 to 16	4	257 to 512	9
17 to 32	5	513 to 1024	10

VI. Experimental Results

We used Digi International's IEEE 802.15.4 complaint XBee and XBee-PRO OEM RF modules. It is a frequently used embedded solution suitable for applications that requires low - power and low-cost. They are easy to use, has standard footprints and require minimal development time and risk.

Due to cost constraints, we have used only 6 devices (one sink, two routers and 3 end-devices) to form a psudo-linear network as shown in the Fig. 5. All the devices except sink sent a randomly generated temperature value (from 20º C to 25º C) to the sink. Sink also sent data to the other devices. The experiment was carried out 1 hour. The set up is carried out over 10 times. The 100% average packet delivery rate was obtained.

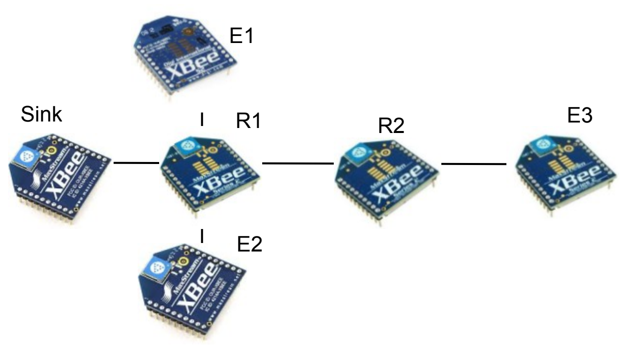

Fig. 6. Experimental testbed using Maxtream XBee OEM kit

Since, we couldn't arrange large number of devices physically; we also carried out a simulation using NS3 simulator.

The area of graphical interface was considered as 1360x640 pixel area. We know that ZigBee specification allows the coordinator to start a network with itself. It then sends beacons periodically. We then add various routers and end devices periodically in one second interval on the simulation area. These devices receive beacons from surrounding router devices. They send a JOIN_REQUEST packet to the intended parent and join to the network. Router devices, after joining to the network, also send beacons periodically.

We carried out simulation on a large pseudo-linear network having 2000 nodes. The scenario is repeated over 100 times and the packet delivery rate was also observed as 100%. These two experiments prove the correctness of the algorithm.

We then used the original ZigBee Tree routing for the network shown in Fig 6. Since node R1 has three children, we set C_m, R_m and L_m (important network parameters) as 4, 3 and 3 respectively. The same previous scenario is repeated again. i.e. all the devices except sink sent a randomly generated temperature value (from 20º C to 25º C) to the sink. Sink also sent data to the other devices. The experiment was carried out 1 hour. The experiment was carried 10 times. The average packet delivery rate was noted. The result is shown in Fig. 7 and Fig. 8:

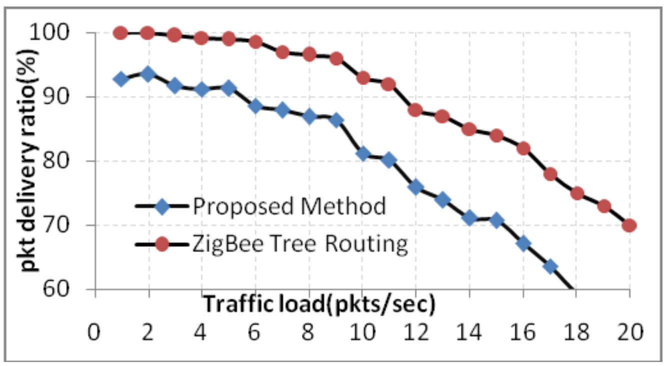

Fig. 7. Perfromance comparison w.r.t. packet delivery ratio

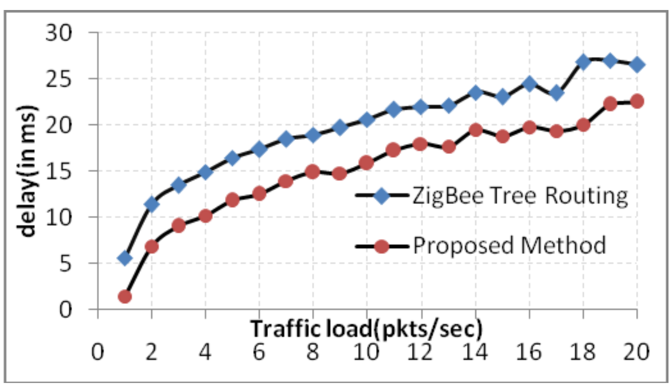

Fig. 8. Perfromance comparison w.r.t. end to end packet delivery time

VII. Summary

This paper proposes a table-free, light-weight routing algorithm based on source routing that supports arbitrary long networks. The proposed algorithm is especially suitable for networks that look almost linear. Experimental and simulation results demonstrate that this novel algorithm exhibits excellent packet delivery performance.

References

[1] Roy U. K., Giri D., "Address borrowing in wireless personal area network", Proceedings. of Intl Anvanced Computing Conf., IACC 2009, March 6-7, India, pp 1074-1079.

[2] Roy U. K., Giri D., "Single level address reorganization in wireless personal area network", Proc. of the Intl Conf. on Computers & Devices for Comm, CODEC 2009, Dec. 14-16, 2009, Calcutta Univ., India. pp 1-4, ISBN- 978-1-4244-5073-2

[3] Roy U. K., Giri D., "Multi-channel personal area net (MCPAN) formation and routing", Proceedings Intl. Conf. on Industrial Engineering Science and Applications, IESA 2014, April 2-4, Durgapur, India, 2014,pp. 73-79

[4] Roy U. K., Giri D., "WPAN routing using huffman technique", Journl of Engg. Science, vol-2, issue-2, Feb., 2013 pp 49-61.

[5] P. Gorday, L. Hester E. Callaway, et. al. "Home networking with IEEE 802.15.4: Developing Standard for Low-Rate wireless personal area networks", IEEE Comm. Magazine, August 2002.

[6] "Wireless Local Area Networks, 1999": IEEE standard 802.15.11

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

[8] IEEE 802.15.4, WMAC (Wireless MAC) and PHY (PHysical Layer) Spec. for low-rate WPANs.

[9] M. J. Lee, J. Zheng, "A comprehensive performance study of IEEE 802.15.4", 2004

[10] J. Garcia-Luna-Aceves, S. Murthy, "An efficient routing protocol for WN (Wireless Networks)", Mobile Nets and Apps, 1996.

[11] U. K. Roy, S. Ray, D. K. Sanyal, "Analysis and optimization of Routing Protocol in IEEE802.15.4", Proc. of the AMOC (Asian Intl. Mobile Comp. Conf.) 2006, JU, Calcutta, India, pp. 172-181

[12] Roy U. K, "Light Weight Addr and Routing Scheme for Resource Constrained WPAN Device," Proceedings. Of 8th ICST 2014 (Intl Conf. on Sensing Tech.), Sept. 2-4, 2014, ISSN-1178-5608, Liverpool, UK. pp 46-51,

[13] Roy U. K., "Extending ZigBee Tree Routing Protocol for Resource-Constrained Devices," APWiMob (Asia Pacific Conf. on Wireless and Mobile), 2014, Bali, Indonesia, Aug. 28-30, pp-48-53

[14] Roy U. K., P. Bhaumik, "Enhanced ZigBee Tree Addressing for Flexible Network Topologies", Proc. of the AIMoC, 2015 (Intl Conf. on Appl. and Innov in Mobile Comp.). February. 12-14, pp. 6-11

[15] ZigBee Doc. 02130r7, Dft 0.90: Net. Spec., July, '04.