Modified Source-Routing for Large-Scale Pseudo-Linear ZigBee Networks

Uttam Kumar Roy
Dept. of IT, Jadavpur University
Block-LB/8, Sector-III, Salt Lake City
Kolkata, India
royuttam@gmail.com

***Abstract*—ZigBee devices are resource-constrained. So, they cannot run traditional heavyweight routing algorithms. Although, ZigBee proposed a table-free, simple routing algorithm for tree networks, it does not support long linear networks typically set up in roadside, mines, agricultural field, mountains etc**

In this paper, we have proposed a 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. The algorithm leverages the properties that a pseudo-linear network has limited branching. Experimental results show that this flexible mechanism exhibits excellent packet delivery performance.

***Keywords*—ZigBee; Pseudo-Linear Networks; Addressing; Source-Routing.**

I. INTRODUCTION

There are many situations where network topologies are often linear or almost linear. Examples include networks formed in mines, roadside [Fig. 1], agricultural field, mountains etc.

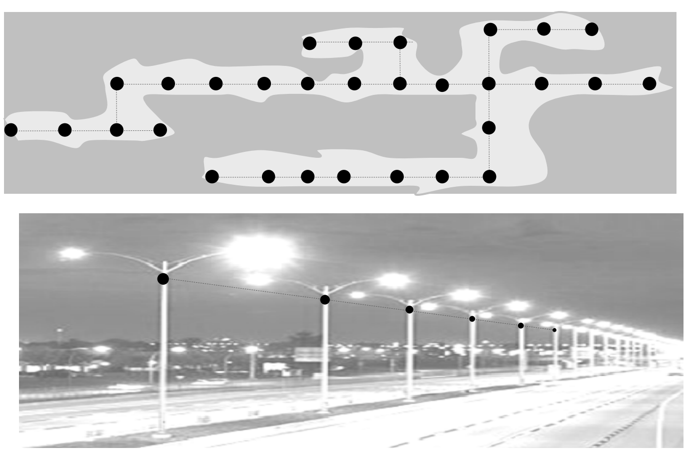


Fig. 1. Psudo-linear networks in mine(above) and roadside (below)

To have low-cost solutions, researchers are trying to use the novel IEEE 802.15.4/ZigBee [15] technology. It is a landmark in the attempt to bring ubiquitous networking into our lives. Since it promotes low cost devices; thereby resource constrained, ZigBee [15] provides a simple and table-free routing algorithm together with an addressing scheme. The beauty of this algorithm is that it uses only some mathematical equations to take routing decisions. However, it does not support long and asymmetric networks.

In[1, 2, 3, 4, 11, 12, 13, 14], we already have proposed several potential solutions to overcome the problems. In this paper, we have proposed a table-free, light-weight routing algorithm based on source routing that supports arbitrary long networks. The proposed algorithm is especially suitable for pseudo-linear networks where nodes typically have one or a few children nodes. There are many instances where networks are this kind. Some examples are already shown in Fig. 1. The route information is kept in sink only and is encapsulated in the packet. The nodes on its way use this information to take routing decisions. The strength of the method is that intermediate nodes do not need and routing table. Consequently devices having limited memory and computing power can run also such algorithm. Experimental results show that this flexible mechanism exhibits excellent packet delivery performance. Since the scheme does not also involve complex calculation and logic, the text of this paper also looks very simple.

II. RELATED WORK

Our investigation [11] revealed that the routing protocols in ZigBee networks is essentially a combination of tree routing and AODV with several optimizations done in consideration of stationary wireless topologies. In [11], we also developed an NS2 simulator for ZigBee PAN to analyze and optimize both tree and mesh routing in a ZigBee network.

In [1], we provided 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 based on mobile IP, was also proposed.

In [2], we extended the Tree routing proposed by ZigBee for the networks to be harsh and asymmetric.

In [3], 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.

In [12, 13], we proposed a flexible, variable-length addressing scheme together with a new lightweight, table-free routing algorithm. The addressing scheme leverages the properties of prefix code and allows devices to have arbitrary number of children and does not also limit network depth.

In [14], we extended the ZigBee addressing scheme. The proposed scheme has all the properties of original addressing scheme and hence the same table-free routing algorithm can be applied. In addition, it supports asymmetric tree networks which original scheme cannot cope up with.

There is relatively scant literature on 802.15.4/ZigBee although its applications have been discussed in [9, 5]. Authors provide one of the first studies of the MAC sub layer while the recent paper [8] 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 [1, 2, 3, 4, 11, 12, 13, 14] and provided several potential solutions using different methodologies. Let us quickly review those problems so that we can appreciate the need of a new addressing 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, before forming the network, 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 value 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 set they remain unchanged. Given those parameters, essentially a virtual tree network skeleton gets created. Devices may sit in an empty location and use its address. So, it may happen that a device can not join at a desired position even if addresses are available.

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:

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

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

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

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

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

IV. PROPOSED ALGORITHM

Since, IEEE 802.15.4 is very low-cost technology; researchers are now trying 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. These networks have often some common characteristics:

- Typically have a sink node (usually a full-fledge computer) and other nodes send sensed data (temperature, humidity, pictures etc.) to sink.
- Networks are often linear or pseudo-linear [Fig. 1] in nature.
- Nodes are static and join to and leave from 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 devices are intended to be low-cost and low-power consuming. Consequently, they are supposed to have low memory, low computing power. Keeping these requirements and the above network characteristics in mind, it is possible to device a simplified 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 procedure 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 needed. The proposed scheme leverages the properties that a pseudo-linear network has limited branches. This means nodes in these kind of networks have a very few (0, 1, 2, 3 or 4) children. This property is cleverly used so that a packet can be routed to a destination based on small routing information encapsulated in the packet itself. Before, discussing the routing algorithm, let us understand how network is formed.

A. *Network Formation*

We propose to use tree networks. Although, tree networks have some inherent problems (there is only one path between every pair of nodes), they have some advantages too. For example, no routing algorithm is necessary if non-root nodes of the tree want to send data 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:

Every router in the tree labels each of its outgoing links (if any) by a (locally) unique binary number. The order of labeling is not important. One such tree is shown in Fig 2.

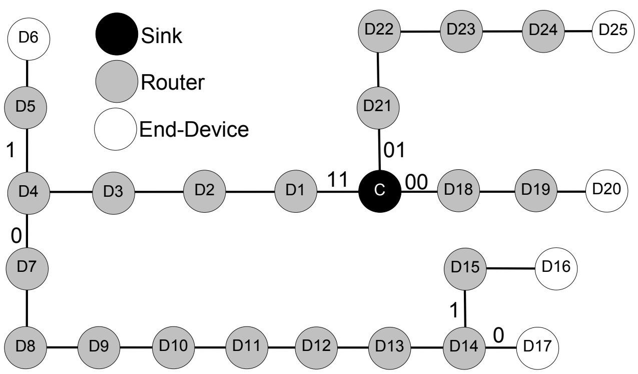


Fig. 2. Psudo-linear network formation

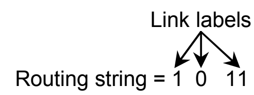
For example, node C has 3 children. So, it uses 2 bits to label its outgoing links. Similarly, node D4 has only two children. So, it uses only 1 bit to label its outgoing links. Nodes having only child (note most of the nodes are of this kind) need not label the link. In general if a router R has C_R number of children, minimum number of bits $N(C_R)$ required to label each outgoing link of R is:

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

Since for a network line roadside-network, most nodes will be non-crossover nodes and C_R for such node is 1 and no bit is required for link labeling. There are a very few crossover nodes where C_R is limited and often ≤ 4 (a crossover of 5 roads) and $N(C_R) \leq 2$. These indicate that the proposed method has very small overhead.

B. *Routing*

To keep network protocol simple, proposed method uses MAC addresses as network addresses. However, any other addressing scheme may also be used. 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 D16 will have following routing string:



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 string for all sensor nodes which look as follows:

D1, D2, D3, D4	D5, D6	D7, D8, D9, D10, D11, D12, D13, D14
11	111	011

D15, D16	D17	D18, D19, D20	D21, D22, D23, D24, D25
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 by carefully assigning the device addresses. For example, the previous table can be restructured as:

D1-D4	D5-D6	D7-D14	D15-D16	D17	D18-D20	D21-D25
11	111	011	1011	0011	00	01

This table contains only 13 network address. 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 X receives a packet from a child, X blindly forwards it to its parent. If X receives a packet destined to D from its parent, *X* checks, if D is X and if so, X accepts it. Else D is a descendant of X. The next hop device ID can be obtained as follows:

If X has one child, forward the packet to the child.
Else {
Extract routing string from the packet.
Get and remove least significant $N(C_X)$ bits from routing string

Put the modified routing string in the packet
Forward packet along the link labeled with those $N(C_X)$ bits
}

C. Example Scenario

To understand how routing takes places, consider the network in Fig 2. Suppose sink wants to send a packet to another end device D16.

The sink looks up the table and finds that the routing string for device D16 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 D1.

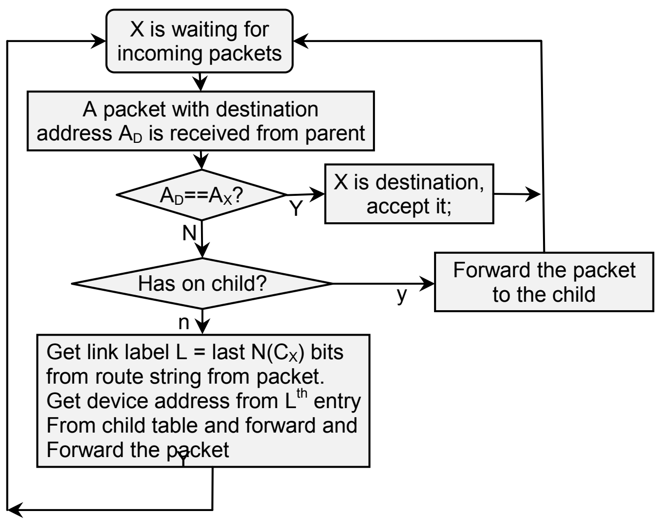


Fig. 3. Routing algorithm

Since D1 exactly one child, it forwards the packet to D2. Devices D2, D3 do the same thing and the packet reaches to D4. 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 through the link labeled 0. The packet comes to device D7. Device D7 to D13 do as D1 did and packet eventually reaches to D14. Since, it has 2 children, it extracts least significant N(2) [=1] bits from 1 which happens to be 1. It puts the remaining null string in the packet and forwards it through the link labeled with 1. The packet comes to device D15 which as one child. The device forwards the packet to it child and the packet finally reaches to the destination D16.

D. Restructuring

Note that number of bits required to label links may change if new roads are laid down. Let us consider Fig 4.

The number of bits $N(C_{R1})$ required to label each link of *R1* is 1 as number of children of router *R1* (i.e. C_{R1}) is two. Now, if a new road is laid [i.e. *X* joins to *R1*], $N(C_{R1})$ will become 2. Consequently, each of the outgoing links (including the new link) of *R1* must be re-labeled by 2 bits. This implies that routing information for all descendants of *R1* must be updated to the sink. We call this procedure as *restructuring*, which incurs an overhead. However, roads are laid down very in-frequently; thus overhead due to it is considerably low.

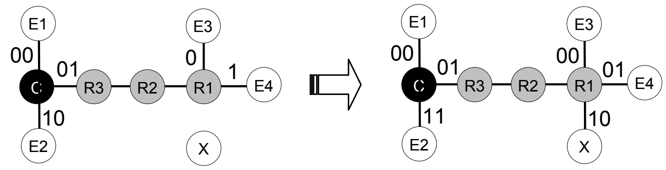


Fig. 4. "Restructuring" Links (left) before (right) After

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.

R1,R2,R3	E1	E2	E3	E4
01	00	10	**01**0	**01**1

=>

R1,R2,R3	E1	E2	E3	E4	X
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. 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 any more. 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 occurrs 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 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 ideal for low-power, low-cost applications. These modules are easy-to-use, share a common footprint 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 scenario is repeated over 10 times and the average packet delivery rate was obtained as 100%.

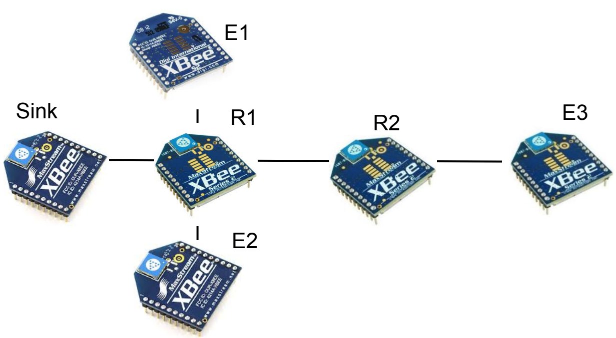


Fig. 6. Experimental testbed using Maxtream XBee OEM kit

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

The 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 every after one second 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.

The simulation was done 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.

VII. CONCLUSION

In this paper, we proposed 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. Analytical and simulation results show that this flexible mechanism exhibits excellent packet delivery performance.

REFERENCES

[1] 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[ukr_borrowing]

[2] 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. [ukr_single]

[3] 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[MCPAN]

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

[5] 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.[Low_Rate]

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

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

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

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

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

[11] 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 [UKR_AMOC]

[12] Uttam Kumar Roy, "Light-Weight Addressing and Routing Schemes for Resource-Constrained WPAN Devices," Proc. Of 8th International Conference on Sensing Technology (ICST 2014), September 2-4, 2014, Liverpool, UK., ISSN-1178-5608, p 46-51 [ICST]

[13] Uttam Kumar Roy, "Extending ZigBee Tree Routing Protocol for Resource-Constined Devices," Asia Pacific Conference Wireless and Mobile (APWiMob2014), August 28-30, 2014, Bali, Indonesia, p-48-53[APWiMob]

[14] Uttam Kumar Roy, Parama Bhaumik, "Enhanced ZigBee Tree Addressing for Flexible Network Topologies", 2nd International Conference Applications and Innovations in Mobile Computing (AIMoC 2015). Feb. 12-14, 2015, pp. 6-11[AiMOC]

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