RESEARCH ARTICLE OPEN ACCESS

Shortest Tree Routing With Security In Wireless Sensor Networks

P.T Arshad [1], Dr. C. Chandrasekar [2]
[1](M.Phil Research Scholar, Sree Narayana Guru College, K.G.Chavadi, Coimbatore, India.)
[2](Asst. Professor, Department of Computer Science and Applications, Sree Narayana Guru College, K.G.Chavadi, Coimbatore, India.)

ABSTRACT
We propose STR to resolve the main reasons of overall network performance degradation of ZTR, which are the detour path problem and the traffic concentration problem. Second, we prove that the 1-hop neighbor nodes used by STR improve the routing path efficiency and alleviate the traffic load concentrated on tree links in ZTR. Third, we analyze the performance of ZTR, STR, and AODV by differentiating the network conditions such as network density, ZigBee network constraints, traffic types, and the network traffic. For modification security purpose we are also encrypting the data packets during transmission. So that the intermediate nodes are not able to view the data during transmission. For Encryption process, we are using RC4 Algorithm. Short cut tree routing is used for minimizing the routing path from source to destination.

Keywords- RC4, Short Cut Routing , Wireless network ,Zigbee tree routing.

I. INTRODUCTION

The ZigBee network layer, which is the core of the standard, provides dynamic network formation, addressing, routing, and network management functions. ZigBee supports up to 64,000 devices in a network with the multihop tree and mesh topologies as well as star topology. Every node is assigned a unique 16-bit short address dynamically using either distributed addressing or stochastic addressing scheme. The routing protocols of ZigBee are diverse so that a system or users can choose the optimal routing strategy according to the applications. The reactive routing protocol in ZigBee is derived from AODVjr (AODV junior), which is one of the representative routing protocols in mobile ad hoc networks. Similar with other MANET routing protocols, ZigBee reactive routing protocol provides the optimal routing path for the arbitrary source and destination pair through the on-demand route discovery.

It requires the route discovery process for each communication pair, so the route discovery overhead and the memory consumption proportionally increases with the number of traffic sessions. Moreover, route discovery packets are flooded to the overall network, which interfere with transmission of other packets even in the spatially uncorrelated area with the route discovery. On the other hand, ZigBee tree routing (ZTR) prevents the route discovery overhead in both memory and bandwidth using the distributed block addressing scheme. In ZTR, since each node is assigned a hierarchical address, a source or an intermediate node only decides whether to forward a packet to the parent or one of the children by comparing its address with the destination address. The most benefit of ZTR is that any source node can transmit a packet to an arbitrary destination in a network without any route discovery overheads. Due to this efficiency, ZTR is considered as a promising protocol for resource constrained devices in diverse applications such as smart grid project and Internet of Things (IoT).

However, in ZTR, packets are forwarded along the tree topology to the destination even if the destination is located nearby. Thus, ZTR cannot provide the optimal routing path, while it does not require any route discovery overhead. Our objective is to provide the near optimal routing path like the reactive routing protocol as well as to maintain the advantages of ZTR such as no route discovery overhead and little memory consumption for the routing table. We propose the shortcut tree routing (STR) that significantly enhances the path efficiency of ZTR by only adding the 1-hop neighbor information. Whereas ZTR only uses tree links connecting the parent and child nodes, STR exploits the neighbor nodes by focusing that there exist the neighbor nodes shortcutting the tree routing path in the mesh topology.

In other words, in STR, a source or an intermediate node selects the next hop node having the smallest remaining tree hops to the destination regardless of whether it is a parent, one of children, or neighboring node. The routing path selection in STR is decided by individual node in a distributed manner, and STR is fully compatible with the ZigBee standard that applies the different routing strategies according to each node’s status. Also, it requires neither any additional cost nor change of the ZigBee standard including the creation and maintenance mechanism of 1-hop neighbor information.

II RELATED WORKS

Among many useful functions in ZigBee network layer, the tree routing algorithm supports simple but reliable routing for any destination address. In ZigBee, network addresses are assigned using a distributed addressing scheme that is designed to provide every potential parent with a finite sub block of network addresses. Due to such addressing scheme, the network constructs a tree topology; each device can manage the address space of its descendant. If the destination address is in the address space that a node is managing. Otherwise, it forwards the packet to its parent node. The parent or child node which receives the packet selects the next hop node according to the destination address in the same manner.

2.1 Zigbee Network

Based on IEEE 802.15.4 PHY/MAC, the ZigBee network layer provides functionality such as dynamic network formation, addressing, routing, and discovering 1 hop neighbors. The size of the network address is 16 bits, so ZigBee is capable to accept about 65535 devices in a network, and the network address is assigned in a hierarchical tree structure. ZigBee provides not only star topology, but also mesh topology. Since any device can communicate with other devices except the PAN Coordinator, the network has high scalability and flexibility. Besides, the self-formation and self healing features makes ZigBee more attractive. The deployed ZigBee devices automatically construct the network, and then changes such as joining/leaving of devices are automatically reflected in the network configuration. The routing protocols that ZigBee provides are tree routing and table-driven routing. Tree routing is based on the block address allocation mechanism, called Cskip, so each device has an address spaces to distribute to their children. When a device has no capability of routing table and route discovery table, it simply follows the hierarchical tree by comparing the destination address. The most significant benefit of tree routing is its simplicity and limited use of resources. Therefore, any device with low resources can participate in any ZigBee compliant network. On the other hand, table driven routing is basically similar to the Ad hoc On-demand Distance Vector (AODV) routing protocol for general multihop ad hoc network. Whereas tree routing is very simple and inefficient, the table-driven routing provides optimal routes to the destination.

III. OUR WORK

3.1 Tree Routing Algorithm

Every potential parent is provided with a finite sub-block of the address space, which is used to assign network addresses to its children. Given nwkmaxChildren (Cm), nwkcMaxDepth (Lm), and nwkmaxRouters (Rm), we can compute the function Cskip (d) as the size of the address sub-block distributed by each parent at depth d as follows:

$$Cskip(d) = \frac{1 + Cm - Rm - Cm \cdot Rm^{Lm-d-1}}{1 - Rm}$$

For example, the kth router and nth end device shall be assigned the network address by their parent at depth d as in the following equation.

$$A_k = A_{parent} + Cskip(d) \cdot (k-1) + 1 \ (1 \le k \le Rm)$$

$$A_n = A_{parent} + Cskip(d) \cdot Rm + n \ (1 \le n \le Cm - Rm)$$

A k^{th} router that has positive Cskip (d) can distribute address spaces to its child nodes. Since every device in the network is a descendant of the ZigBee coordinator and no device in the network is the descendant of any ZigBee end device, any device with address A at depth d has the destination device with address D if the following equation is satisfied.

$$A<D<A+ Cskip (d\_1)$$

In tree routing, if the destination is a descendant, the device sends the data to one of its children; otherwise, it sends to its parent.

3.2 Neighbor table

Each device in ZigBee maintains a neighbor table which has all the neighbor information in the 1-hop transmission range. If users limit the size of the neighbor table, the selected numbers of neighbor entries are stored in the table. The contents for a neighbor entry are the network’s PAN identifier; node’s extended address, network address, device type and relationship. Optionally, additional information such as beacon order, depth or permit joining can be included. Entries in the table are created when the node joins to an existing network. When a joining node requests a nlmenetwork- discovery, it receives response beacons from already joined nodes. The newly joined node stores neighbors’ information from the information contained in beacon packets.

Conversely, the neighbor entry is removed when the neighbor node leaves the network. Nodes can know this fact by receiving NLME-LEAVE. Indication messages. Since the information on the neighbor table is updated every time a device receives any frame from the some neighbor node, the information of the neighbor table can be said to be up-to-date all the time.

IV. PROBLEM FORMULATION

The tree routing protocol uses only parent and child relationship for routing, ignoring neighbor nodes. As a result, packets may be routed through several hops towards the destination even if this is within sender's 1-hop transmission range. Fig. 1 shows an example of the described problem. In Fig. 1, the packet from the source node goes up to the root node following the parent node, and goes back to the destination. In this way, 4 hops are required to reach the destination. However, if the source node sends the packet directly to the destination, it needs only 1 hop routing cost. In many cases, the routing overhead of tree routing algorithm can not be avoided if only parent-child relationships are considered in the routing. In order to overcome such problem, each node should consider its neighbor nodes as next hop nodes.

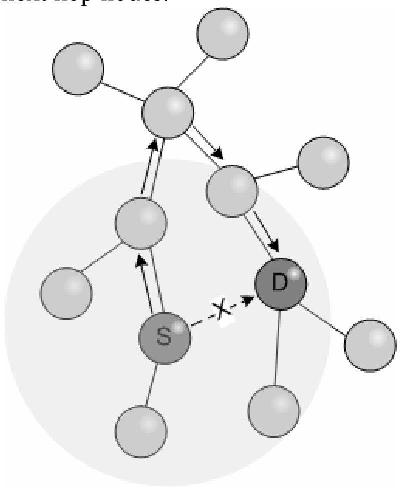

Fig. 1. Problem of Tree Routing

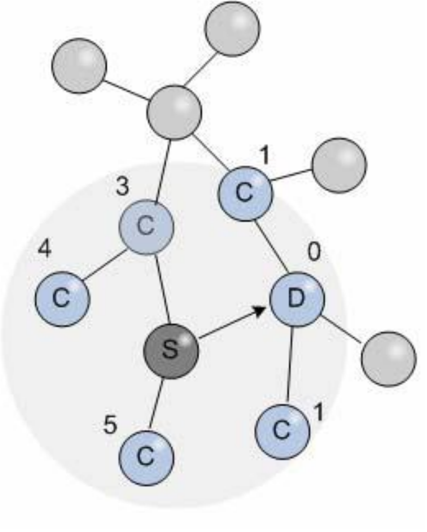

Fig. 2. Candidates for Next Hop

4.1 Shortcut Tree Routing Algorithm

We propose the shortcut tree routing algorithm that improves existing ZigBee tree routing by using the neighbor table. In other words, the proposed algorithm basically follows ZigBee tree routing algorithm, but chooses neighbor nodes as next hop nodes if the routing cost to the destination can be reduced. The neighbor table that we use in the proposed algorithm is defined in the ZigBee specification, so we don't need to make an effort to search neighbor list. In order to choose the next hop node that can reduce the routing cost, the remaining hop count from the next hop node to the destination is computed for all the neighbor nodes including parent and children nodes. As Fig. 2 shows, the remaining hops to the destination for each neighbor can be computed assuming that the route from the neighbor to the destination goes along the tree. In the above Fig.2, the route cost can be minimized if the sender transmits the data directly to the destination. Find_NextHopAddr() function described on table 1 is the algorithm for an intermediate or source node to select the next hop node which has the minimum remaining hop count for the given destination. Because the proposed algorithm follows fundamentally the ZigBee tree routing, the parent or child node is selected as the next hop node in lines 2-3. In addition, the remaining routing cost when we follow ZigBee tree routing is stored into minNHRouteCost. In line 4-13, intermediate or source nodes check the remaining routing cost myRouteCost when selecting a neighbor node as the next hop node. The remaining routing cost is calculated based on the remaining hop count to the destination assuming that the packet goes along the ZigBee tree routing. In order to calculate the remaining hop count, the hierarchical address structure is used.

Algorithm Implementation Steps

- A symmetric key encryption algo. Invented by Ron Rivest.
- Normally uses 64 bit and 128 bit key sizes.
- Most popular implementation is in WEP for 802.11 wireless networks and in SSL.
- Cryptographically very strong yet very easy to implement.
- Consists of 2 parts: Key Scheduling Algorithm (KSA) & Pseudo-Random Generation Algorithm
- Using a secret key generate the RC4 keystream using the KSA and PRGA.
- Read the file and xor each byte of the file with the corresponding keystream byte.
- Write this encrypted output to a file.
- Transmit file over an insecure channel.

4.2 AODV Routing Protocol

The Ad-hoc On-demand Distance Vector (AODV) routing protocol is a routing protocol used for dynamic wireless networks where nodes can enter and leave the network at will. To find a route to a particular destination node, the source node broadcasts a RREQ to its immediate neighbors. If one of these neighbors has a route to the destination, then it replies back with a RREP. Otherwise the neighbors in turn rebroadcast the request. This continues until the RREQ hits the final destination or a node with a route to the destination. At that point a chain of RREP messages is sent back and the original source node finally has a route to the destination.

We proved that AODV protocol never produces routing loops by proving that a combination of sequence numbers and hop counts is monotonic along a route. This means that there can't be any loop in the routing table. The proof was done completely automatically and our algorithm was able to generate all the predicates needed. The Ad hoc On Demand Distance Vector (AODV) routing algorithm is a routing protocol designed for ad hoc mobile networks. AODV is capable of both unicast and multicast routing. It is an on demand algorithm, meaning that it builds routes between nodes only as desired by source nodes. It maintains these routes as long as they are needed by the sources. Additionally, AODV forms trees which connect multicast group members. The trees are composed of the group members and the nodes needed to connect the members. AODV uses sequence numbers to ensure the freshness of routes. It is loop-free, self-starting, and scales to large numbers of mobile nodes.

AODV builds routes using a route request / route reply query cycle. When a source node desires a route to a destination for which it does not already have a route, it broadcasts a route request (RREQ) packet across the network. Nodes receiving this packet update their information for the source node and set up backwards pointers to the source node in the route tables. In addition to the source node's IP address, current sequence number, and broadcast ID, the RREQ also contains the most recent sequence number for the destination of which the source node is aware. A node receiving the RREQ may send a route reply (RREP) if it is either the destination or if it has a route to the destination with corresponding sequence number greater than or equal to that contained in the RREQ. If this is the case, it unicasts a RREP back to the source. Otherwise, it rebroadcasts the RREQ. Nodes keep track of the RREQ's source IP address and broadcast ID. If they receive a RREQ which they have already processed, they discard the RREQ and do not forward it. As the RREP propagates back to the source, nodes set up forward pointers to the destination. Once the source node receives the RREP, it may begin to forward data packets to the destination. If the source later receives a RREP containing a greater sequence number or contains the same sequence number with a smaller hopcount, it may update its routing information for that destination and begin using the better route. As long as the route remains active, it will continue to be maintained. A route is considered active as long as there are data packets periodically travelling from the source to the destination along that path. Once the source stops sending data packets, the links will time out and eventually be deleted from the intermediate node routing tables. If a link break occurs while the route is active, the node upstream of the break propagates a route error (RERR) message to the source node to inform it of the now unreachable destination(s). After receiving the RERR, if the source node still desires the route, it can reinitiate route discovery.

V. CONCLUSION

This paper introduces the problem of ZigBee tree routing and proposes a shortcut tree routing protocol that overcomes the overhead occurred when following the tree topology. In the proposed algorithm, the neighbor table that is originally defined in the ZigBee standard is used to find the optimal next hop node that has the smallest remaining hop count to the destination. The shortcut tree routing algorithm is efficient in terms of both routing performance and time complexity: it reduces significantly the required routing costs and it can be solved within polynomial time even when the number of neighbors is not limited. We have identified the detour path problem and traffic concentration problem of the ZTR. These are the fundamental problems of the general tree routing protocols, which cause the overall network performance degradation. To overcome these problems, we propose STR that uses the neighbor table, originally defined in the ZigBee standard. In STR, each node can find the optimal next hop node based on the remaining tree hops to the destination. The mathematical analyses prove that the 1-hop neighbor information in STR reduces the traffic load concentrated on the tree links as well as provides an efficient routing path. The network simulations show that STR provides the comparable routing performance to AODV as well as scalability respect to the network density and the network traffic volume by suppressing the additional route discovery process.

REFERENCES

[1] *ZigBee Alliance*, Network Layer Specification 1.0, Dec. 2004.
[2] *ZigBee Alliance*, www.ZigBee.org

[3] “*Wireless Medium Access Control and Physical Layer Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs),*” IEEE Std 802.15.4-2003, IEEE Computer Society, 01 October 2003.

[4] G. Ding, Z. Sahinoglu, P. Orlik, J. Zhang, and B. Bhargava, "*Efficient and Reliable Broadcast in ZigBee Networks*,” IEEE Transaction on Mobile Computing, IEEE SECON’05.

[5] D. Kim, Y. Doh, S. Yoo, K. Chang, W. Park, C. Seo, “*Low Rate WPAN Technologies and Standards*,” KISS Information and Communications Journal, Dec. 2004.

[6] J. Kim, H. Lee, D. Hwang, B. Kim, “*Development Trend of Standards for Low-Rate, Low-cost, and Low-Power Wireless PAN*, “ Vol. 18, No. 2, pp. 37, ETRI trends, 2003.

[7] E. Callaway, P. Gorday, L. Hester, J. A. Gutierrez, and M. Naeve, “*Home networking with IEEE 802.15.4: A developing standard for lowrate wireless personal area networks,*” IEEE Communications Magazine, page 70-77, August 2002.

[8] D. Han and J. Lim, “*Smart Home Energy Management System Using IEEE 802.15.4 and ZigBee,*” IEEE Trans. Consumer Electronics, vol. 56, no. 3, pp. 1403-1410, Oct. 2010.

[9] S. Chen et al., *“A Reliable Transmission Protocol for ZigBee-Based Wireless Patient Monitoring,*” IEEE Trans. Information Technology in Biomedicine, vol. 16, no. 1, pp. 6-16, Nov. 2012.