A Path-Connected-Cluster Wireless Sensor Network and Its Formation, Addressing, and Routing Protocols

Mr. Yogesh V. Patil, Lord Krishna College of Technology, Indore,
Mr. Deepak Singh Chouhan, Lord Krishna College of Technology, Indore,
Dr.Sanjay Thakur Lord Krishna College of Technology, Indore,

Abstract:-Although *wireless sensor networks (WSNs)* have been extensively researched, their deployment is still a main concern. We observe that many monitoring applications for WSNs have adopted a *path-connected-cluster (PCC)* topology, where regions to be monitored are deployed with clusters of sensor nodes. Since these clusters might be physically separated, paths of sensor nodes are used to connect them together. We call such networks *PCC-WSNs*. PCC-WSNs may be widely applied in real situations, such as bridge-connected islands, street-connected buildings, and pipe-connected ponds. In this paper, we show that the address assignment scheme defined by ZigBee will perform poorly in terms of address utilization. We then propose a systematical solution, which includes network formation, automatic address assignment, and light-weight routing. Simulation results verify the effectiveness of the proposed solution.

Index Terms — Automatic Address assignment, Wireless Sensor Network, ZigBee.

1 INTRODUCTION

WIRELESS sensor network (WSN) usually needs to configure itself automatically and support ad hoc routing. A lot of research works have been dedicated to WSNs, including power management, routing , data gathering , sensor deployment and coverage issues, and localization. On the application side, habitat monitoring is explored in, wildfire monitoring is addressed in, healthcare system is proposed in, and navigation is studied in.

To form a WSN, two most important issues are addressing and routing. Strict per-node addressing is expensive in a dense network, because not only the address space would be large, but also these addresses would need to be allocated and managed according to the topology change. Allocation of addresses in a dense network is a problem which is often underestimated. On the other hand, routing is to discover paths from source nodes to destination nodes based on their network addresses. Path discovery in a dense network could incur high communication overhands. Therefore, designing a light-weight addressing and routing protocol for WSNs is very important.

Recently, ZigBee has been proposed for addressing and routing on WSNs. It supports three kinds of network topologies, namely star, tree, and mesh networks. A *ZigBee coordinator* is responsible for initializing, maintaining, and controlling the network. Star networks can only cover small areas. For tree and mesh networks, communications can be conducted in a multi-hop fashion. The backbone of a tree/mesh network is formed by one ZigBee coordinator and multiple

ZigBee routers. An *end device* must associate with the coordinator or a router. In a tree network, routing can be done in a stateless manner; a node can simply route packets based on nodes' 16-bit short addresses, which are assigned based on the tree structure. In fact, a mesh network also has a tree inside to serve as its backbone; routing can go directly along the tree without route discovery or go along better paths if a node is willing to conduct route discovery first.

In the literature, most works have assumed that a ZigBee network grows in an arbitrary manner. Recently, the *long-thin topology* (Fig. 1(a)) has been proposed for applications where sensor deployment is subject to environmental constraints.

The use of long-thin network ranges from leakage detection of fuel pipes, tunnel monitoring, street lights monitoring, flood protection of rivers, debris flow monitoring, barrier coverage [19], and in-sewer gas monitoring. In this paper, we further extend the long-thin topology to a *path-connected-cluster (PCC)* topology, where regions requiring intensive sensing are deployed with clusters of sensor nodes and these clusters, which are physically separated, are connected by long paths for occasional communications.

We call such topologies PCC-WSNs. Fig. 1(b) shows an application for habitat monitoring in a wildlife park. Sensors for different habitat zones form different clusters. Data from these clusters is collected through paths connecting them. Such "sometimes fat, sometimes slim" topologies would worsen the *orphan problem*, which states that the ZigBee address assignment may not allow some nodes (called *orphans*) to join the network even if there are available addresses elsewhere.

Although ZigBee supports address-based routing through its distributed addressing scheme, it could incur a lot of orphans or result in waste of address space. The virtual coordinate addressing schemes in try to provide stateless routings directly from nodes' addresses. However, additional GPS devices or localization mechanisms should be involved. Moreover, these schemes still need a lot of address spaces.

These works all focus on compact assignment of addresses to sensor nodes, but they need additional routing protocols to deliver packets because they do not support address-based routing. The work allows network addresses to be reused to conserve address space, but it only supports many-to-one communication. The goal of our work is to propose an address-light and routing-light protocol for PCC-WSNs. Our approach is based on the principle of ZigBee address assignment, but leads to much more compact address usage than the original ZigBee's design, thus significantly alleviating the orphan problem in PCC-WSNs. Furthermore, based on our addressing, routing still incurs low communication overheads. This work contributes in formally defining the PCC-WSN topology. Given a PCC-WSN, we present a formation scheme to automatically separate paths from clusters in a distributed manner. Then we propose a ZigBee-like address assignment scheme for a PCC-WSN. In particular, we design different addressing strategies for slim parts (paths) and fat parts (clusters) of a PCC-WSN. This design allows us to conduct light-weight, address-based routing. Although this requires slight modification to ZigBee specification, we find this leads to quite efficient communications. The rest of this paper is organized as follows.

EXISTING SYSTEM & ITS DEFINITION

A. ZigBee Address Assignment and Tree Routing

In ZigBee, network addresses are assigned to devices in a distributed manner. To form a network, the coordinator determines the maximum number of children per router (*Cm*), the maximum number of child routers per router (*Rm*), and the maximum depth of the network (*Lm*). Note that children of a router include child routers and child end devices. So $Cm \geq Rm$ and up to $Cm - Rm$ children of a router must be end devices (an end device cannot have children). Addresses are assigned in a top-down manner. The coordinator takes 0 as its address and divides the remaining address space into $Rm + 1$ blocks. The first *Rm* blocks are to be assigned to its child routers and the last block has $Cm - Rm$ addresses, each to be assigned to a child end device. The similar approach is adopted by each router to partition its given address space.

From *Cm*, *Rm*, and *Lm*, each router at depth *d* can compute a *Cskip(d)* value, which is the size of one address block to be assigned to a child router:

$$Cskip(d) = \begin{cases} 1 + Cm \times (Lm - d - 1), & \text{if } Rm = 1. \\ \dfrac{1+Cm-Rm-CmRm^{Lm-d-1}}{1 - Rm}, & \text{otherwise.} \end{cases}$$

The value of *d* is 0 for the coordinator and is increased by one after each level. For example, given an address block, a router at depth *d* will take the first address for itself, reserve *Rm* blocks, each with *Cskip(d)* addresses, for its child routers, and reserve $Cm - Rm$ addresses for its child end devices. Fig. 2 shows an example of ZigBee address assignment. Clearly, in Fig. 2, the value of Rm is at least 3 for supporting 3 router children. Note that ZigBee network address is 16 bits. Even though we set Lm to 9, *B* and *C* still can not associate with the network. Even worse, such address assignment would work poorly in a PCC-WSN because of its "sometimes fat, sometimes slim" nature.

With the above address assignment, ZigBee supports very simple address-based routing. When a router receives a packet for *Adest* , it first checks if it is the destination or one of its children is the destination. If so, it accepts the packet or forwards this packet to its child whose address block contains *Adest* . Otherwise, it relays the packet to its parent.

Assume that the depth of this router is *d* and its address is *A*. This packet is forwarded to its child *Ar* which satisfies $Ar < Adest < Ar + Cskip(d) + 1$ such that *Ar* = *A* + 1 + _Adest − (A + 1)/Cskip(d)_ × *Cskip(d)*. If the *Adest* is not a descendant of *A*, this packet will be forwarded to its parent. Note that in a mesh network, nodes are also assigned addresses following these rules. This means that address-based routing can coexist with a mesh routing.

B. Definition of PCC-WSN

A WSN is modeled as an undirected graph $G = (V, E)$, where *V* contains all nodes and *E* contains all communication links between nodes. Each edge in *E* is bidirectional (we do not consider directed links). One special node $t \in V$ is designated as the *coordinator*. A PCC-WSN has a special topology in that *V* can be divided into two sets *C* and *P*, where *C* is a set of clusters and *P* is a set of paths. A cluster in *C* is a group of connected nodes with dense connectivity. A path in *P* is a linear topology with at least δ nodes and each node has a degree ≤ 2. An end node of a path either connects to a cluster node or is a terminator itself (which has degree = 1). Intuitively, we assume that clusters are sufficiently separated physically by at least δ nodes. Therefore, there may exist short paths (with length ≤ δ) in a cluster. The value of δ can be chosen by the network administrator according to physical constraints. Fig. 3 shows an abstraction of PCC-WSN. A possible instance of Fig. 3(a) is shown in Fig. 3(b). Here, δ = 5, so nodes in *H* form a path. (Note that node *w* in H can be a degree-1 terminator or degree-2 node connecting to another cluster.) Nodes *x*, *y*, and *z* in *C* are not regarded as a path, but as a part of cluster *C*.

NETWORK FORMATION, ADDRESSING, AND ROUTING PROTOCOLS

Given a PCC-WSN, we propose a low-cost, fully automated scheme to initialize it, assign addresses to nodes, and conduct ZigBee-like tree routing. First, a distributed network formation procedure will be launched by the coordinator *t* to divided nodes into two sets *C* and *P*. Then, a two-level address assignment scheme is conducted to assign a level-1 and a level-2 addresses to each node. A level-1 address is to uniquely identify a path or a cluster. A level-2 address is similar to ZigBee addressing but is confined within one cluster/path. For simplicity, we assume that all nodes are router-capable devices. Finally, we show how to conduct routing based on our two level addressing. Also, we address how our protocol can adapt to changeable topologies.

A. Network Formation

Given a PCC-WSN *G* = *(V*, *E)*, the network formation process has three goals: (i) to partition *G* into clusters and paths, (ii) to assign a *group ID (GID)* to each cluster/path, which should be known to each member in that cluster/path, and (iii) to identify an *entry node* for each cluster/path, which is the one nearest to *t* in terms of the number of clusters/paths if we travel from *t* to the entry node (as a special case, *t* will serve as the entry node of its cluster). At this stage, addressing is based on devices' MAC addresses. It is assumed each node *v* has a unique MAC address *MAC(v)*. The GID of a cluster/path will be the MAC address of its entry node. For example, in Fig. 3, each cluster/path has an entry node. Nodes *x*3 and *x*5 are the entry nodes of path *H* and cluster *C*, respectively.

In our protocol, a node has four states, as shown in Fig. 4. First, it is in the INIT (initial) state. In the CLF (classification) state, it tries to decide if it is within a cluster or a path. In the PB (probe) state, it helps to determine the range of its cluster/path and elect the entry node of its cluster/path. Finally, it enters the TERM (terminated) state. The detailed process is discussed below.

Initially, all nodes are in the INIT state. The coordinator *t* first enters the CLF state and starts the network formation process by broadcasting a START message around the network. Each node receiving a START for the first time should help rebroadcast it.

2) On a node *v* receiving the first START message, it enters the CLF state. Then *v* tries to determine its degree (i.e., number of neighbors) in *G*. This can be easily achieved by nodes exchanging periodically HELLO packets. According to its degree, *v* classifies its type by setting *type(v)* = "cluster" if its degree is ≥ 3 and setting *type(v)* = "t-path" otherwise. Here, "t-path" means a *tentative* path.

3) If *v* has *type(v)* = "t-path", it needs to confirm its type further (recall that a path must contain at least δ nodes). To do so, if *v* is adjacent to a node of type "cluster" (i.e., it is an end node of a tentative path), it sends a TRAVERSE packet to calculate the length of its path. (This can be done by sending a TRAVERSE packet containing *len* = 1 to its neighbor with type "t-path".

On *u* receiving the packet and *u* not having a neighbor with type "cluster", it sets *len* = *len* + 1 and forwards the packet to the other direction. Otherwise, *u* is the end of the path. It checks if (*len* +1) ≥ δ. If so, *u* notifies all nodes along the reverse path to change their types to "path"; otherwise, *u* notifies them to change their types to "cluster".)

4) A node *v*, once confirming its type as "cluster" or "path", will enter the PB state. The probe process involves two messages, C-PROBE and P-PROBE, for searching the ranges of clusters and paths, respectively. *v* keeps a variable *GID(v)* to track its group ID, a variable *Dist (v)* to track its distance to the coordinator, in terms of the number of clusters/paths from *t* to *v*'s cluster/path, and a variable *PAR(v)* to track its parent cluster/path if *v* is an entry node. Initially, *v* sets all variables to ∞, except the coordinator *t*, which sets *GID(t) = MAC(t)*, *Dist (t)* = 0, and *PAR(t) = MAC(t)*.

This process is started by *t* after it enters the PB state. It first broadcasts a C-PROBE(*GID(t)*, *Dist (t)*). Below, given two pairs
(GID, Dist) and
(GID_, Dist_),
We say that *(GID, Dist)* < *(GID_, Dist_)* if (*Dist* <*Dist_*) or (*Dist* = *Dist_* and *GID* < *GID_*). C-PROBE
and P-PROBE are propagated by the following rules.

a) On *v* of *type(v)* = "cluster" receiving a C-PROBE (*g*, *d*), it checks if *(g, d)* < *(GID(v)*, *Dist (v))*. If so, it updates its *GID(v)* = *g* and *Dist (v)* = *d* and broadcasts a C-PROBE(*GID(v)*, *Dist (v)*).

b) On v of $type(v)$ = “path” receiving a P-PROBE (g, d), it checks if $(g, d) < (GID(v), Dist (v))$.
If so, it updates its $GID(v) = g$ and $Dist (v) = d$ and broadcasts a P-PROBE($GID(v)$, $Dist (v)$).
c) On v of $type(v)$ = “cluster” receiving a P-PROBE (g, d), it checks if $(MAC(v), d + 1) < (GID(v), Dist (v))$. If so, v will bid to serve as the entry node of its cluster and update its $GID(v) = MAC(v)$,
$Dist (v) = d + 1$, and $PAR(v) = g$. Then v broadcasts a C-PROBE($GID(v)$, $Dist (v)$).
On v of $type(v)$ = “path” receiving a C-PROBE (g, d), it checks if $(MAC(v), d + 1) < (GID(v)$,
$Dist (v))$. If so, v will bid to serve as the entry node of its path and update its $GID(v) = MAC(v)$,
$Dist (v) = d + 1$, and $PAR(v) = g$. Then v broadcasts a P-PROBE($GID(v)$, $Dist (v)$).
5) After all nodes are stable, they enter the TERM state. We make some remarks below. The above process tries to form a minimum spanning tree by regarding each path/cluster as a supernode. Each node which is in a cluster and connects to a path could be a candidate entry node. Note that every node in a path has degree ≤ 2. By our definition, an end point in a path will connect to only one node in a cluster.
For example, in Fig. 3(a), cluster D could have four candidate entry nodes, and the one which has the shortest distance to the coordinator will become the entry node (in this example, it is $x7$). In Fig. 3(b), $x5$ is the only node connected to H. If there exist multiple candidate entry nodes having the same distance to the coordinator, we use nodes’ MAC addresses to break the tie. The selection of an entry node of a path is similar. Hence, there exists only one entry node in each supernode. Also, a node v knows that it is an entry node if $MAC(v) = GID(v)$. Also note that nodes may not enter the same state at the same time. For example, a node in the CLF state may receive a “premature”
C-PROBE/P-PROBE message, which it is unable to process yet. In this case, the receiving node will buffer such packets. Once it enters the legal state, it can retrieve them for processing. For simplicity, we do not discuss how we know that all nodes are “stable” in step 5. This can be achieved by *distributed termination detection* protocols [27], and we omit the details.
We further discuss the effect of δ. A smaller δ could result in many short paths connecting clusters. This may increase the number of supernodes in our level-1 addressing. Thus, larger $Cm(1)$ and $Lm(1)$ may be needed, resulting in larger address spaces. But the impact on lever-2 address space is limited. Reversely, a larger δ may prohibit some “paths” from becoming paths. Thus, there may be less supernodes leading to smaller $Cm(1)$ and $Lm(1)$. However, combining clusters and paths requires larger $Lm(2)$. Also, increasing $Lm(2)$ by one results in doubling the level-2 address space. Hence, making δ just a little bit smaller than the average length of paths is
preferred.

B. Address Assignment

We propose a two-level addressing.
It has two purposes:
(i) to reduce address space and
(ii) (ii) to support ZigBeelike stateless routing.
In level-1 addressing, we regard each cluster/path as a supernode and use ZigBee-like addressing
to assign an m-bit address to each supernode.
In level-2 addressing, we again apply the ZigBee-like addressing on each individual cluster/path to assign an n-bit address to each node.
The concatenation of the level-1 and the level-2 addresses forms a node v’s network address, denoted by $(L1(v), L2(v))$.
During this process, we will also construct a *Descendant Table (DT)*, which allows an entry node to reach the entry nodes of its child supernodes.

C. Routing

With our two-level addressing, we also design a two-level routing approach consisting of a *level-1 routing* and a *level-2 routing*. The former can be imagined as routing in GL, which can assist in routing packets to the supernode containing the destination. The later is to route packets simply within the supernode (the same cluster/path). Therefore, suppose that a node v receives a packet destined to dst. If $L1(dst) = L1(v)$, v can simply adopt the level-2 routing to transmit packets to dst. Otherwise, v will first perform the level-1 routing until $L1(dst) = L1(v)$. Then v also applies the level-2 routing to transmit packets to dst.
Note that the concept of our level-1 routing is to determine which cluster/path the packets should be forwarded to and how to forward the packet to that cluster/path. When routing to cross the cluster/path, it still applies the level-2 routing.
Based on our two-level addressing, given a source x and a destination y both in the same cluster/path, the distance $P(2)(L2(x), L2(y))$ between them can be easily determined

CONCLUSION

We contribute in formally defining the PCC-WSN topology. Also, we have proposed a formation scheme to divide nodes into several paths and clusters. Then a two-level ZigBee-like hierarchical address assignment and routing schemes for PCC-WSN are conducted.The proposed address assignment scheme assigns each node both level-1 and level-2 addresses as its network address. With such a hierarchical structure, routing can be easily done based on addresses of nodes. We also show how to allow nodes to utilize shortcuts. With our design, not only network addresses can be efficiently utilized and the spaces required for the network addresses can be significantly reduced, but also the network scale can be enlarged to cover wider areas without suffering from address shortage. We have also verified our schemes by simulation programs. In the future, it deserves to consider applying this work to real cases such as habitat monitoring in a wildlife park or structure monitoring in an amusement park.

REFERENCES

[1] C.-T. Cheng, C. K. Tse, and F. C. M. Lau, “A clustering algorithm for wireless sensor networks based on social insect colonies,” *IEEE Sensors J.*, vol. 11, no. 3, pp. 711–721, Mar. 2011.
[4] C.-T. Cheng, C. K. Tse, and F. C. M. Lau, “A delay-aware data collection network structure for wireless sensor networks,” *IEEE Sensors J.*, vol. 11, no. 3, pp. 699–710, Mar. 2011.
[5] A. Chen, T. H. Lai, and D. Xuan, “Measuring and guaranteeing quality of barrier-coverage in wireless sensor networks,” in *Proc. ACM Int. Symp. Mobile Ad Hoc Netw. Comput.*, 2008, pp. 421–430.
[6] H.-S. Ahn and K. H. Ko, “Simple pedestrian localization algorithms based on distributed wireless sensor networks,” *IEEE Trans. Ind. Electron.*, vol. 56, no. 10, pp. 4296–4302, Oct. 2009.
[7] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson,“Wireless sensor networks for habitat monitoring,” in *Proc. ACM Int. Workshop Wireless Sensor Netw. Appl.*, 2002, pp. 88–97.
[8]Akyildiz, I.F, Su, W. Sankarasubramaniam, Y.; Cayirci, E. “Wireless sensor networks: A survey”. Comput. Netw. 2002, 38, 393–422.
[9]Al-Karaki, J.N.; Kamal, A.E.”Routing techniques in wireless sensor networks: A survey”.IEEE Wirel. Commun. 2004, 11, 6–28.
[10]”Sustainable WirelessSensor Networks” Seah, W., Tan, Y. Eds.; InTech Open Access Publisher: Rijeka, Croatia, 2010.
[11] Li, C., Zhang, H.X.; Hao, B.B.; Li, J.D. “A survey on routing protocols for large-scale wireless sensor networks”. Sensors 2011, 11, 3498–3526.
[12]Haas, Z.J.; Halpern, J.Y.; Li, L.”Gossip-Based Ad Hoc Routing”. In Proceedings of the 19th Conference of the IEEE Communications Society (INFOCOM), New York, NY, USA, 23–27 June 2002; pp. 1707–1716.
[13] Kulik, J.; Heinzelman, W.R.; Balakrishnan, H.”Negotiation based protocols for disseminating information in wireless sensor networks”. Wirel. Netw. 2002, 8, 169–185.
[14] Intanagonwiwat, C.; Govindan, R.; Estrin, D.; Heidemann, J. “Directed diffusion for wireless sensor networking.IEEE/ACM Trans”. Netw. 2003, 11, 2–16.
[15] Braginsky, D.; Estrin, D.”Rumor Routing Algorithm for Sensor Networks. In Proceedings of the First ACM International Workshop on Wireless Sensor Networks and Applications” (WSNA), Atlanta, GA, USA, 28 September 2002; pp. 22–31.
[16] Karp, B.; Kung, H. “GPSR: Greedy Perimeter Stateless Routing for Wireless Networks’. In Proceedings of the Sixth Annual International Conference on Mobile Computing and Networking (MOBICOM), Boston, MA, USA, 6–11 August 2000; pp. 243−254.
[17] Niculescu, D.; Nath, B. “Trajectory Based Forwarding and Its Applications. In Proceedings of the Ninth Annual International Conference on Mobile Computing and Networking (MOBICOM)”, San Diego, CA, USA, 14–19 September 2003; pp. 260–272.
[18] Shah, R.C.; Rabaey, J.M. “Energy Aware Routing for Low EnergyAd HocSensor Networks. In Proceedings of the Wireless Communications and Networking Conference (WCNC)”, Orlando, FL, USA, 17–21 March 2002; pp. 350–355.
[19] Schurgers, C.; Srivastava, M.B. “Energy Efficient Routing in Wireless Sensor Networks. In Proceedings of Military Communications Conference on Communications for Network-Centric Operations”: Creating the Information Force, McLean, VA, USA, 28–31 October 2001; pp. 357–361.
[20] Heinzelman, W.R.; Chandrakasan, A.; Balakrishnan, H. “Energy-Efficient Communication Protocol for Wireless Microsensor Networks”. In Proceedings of the 33rd Annual Hawaii International Conference on System Sciences, Maui, HI, USA, 4–7 January 2000; pp. 10–19.
[21]Younis, O.; Fahmy, S. “HEED: A hybrid, energy-efficient, distributed clustering approach for ad-hocsensor networks”. IEEE Trans. Mobile Comput. 2004, 3, 366–379.