2014 IEEE 11th International Conference on Mobile Ad Hoc and Sensor Systems

Backbone Discovery In Thick Wireless Linear Sensor Networks

Imad Jawhar[1], Xin Li[2], Jie Wu[3] and Nader Mohamed[1]

[1]College of Information Technology, UAE University, Alain, UAE
[2]College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, China
[3]Department of Computer and Information Sciences, Temple University, Philadelphia, Pennsylvania, USA

***Abstract*— Wireless sensor networks (WSNs) constitute an important area of research that is emerging. This is taking place due to the rapid and significant developments, which have led to sensing devices with increasingly smaller size, faster processing, lower energy consumption, as well as larger storage and communication capacities. In addition, as the amount of physical, chemical and biological conditions that are able to be sensed increases, WSNs are finding numerous applications in areas such as environmental, military, health care, and infrastructure monitoring. Many of these applications involve lining up the sensors in a linear form, making a special class of these networks, which are defined as Linear Sensor Networks (LSNs). In a previous paper, we introduced LSNs and provided a classification and motivation for designing networking protocols that can take advantage of the predictable linearity of the topology in order to optimize the performance, reliability, fault tolerance, energy consumption, and network lifetime. In this paper, we provide a topology discovery protocol for thick LSNs where, due to the nature of the monitored structure or area, and the deployment strategy, the nodes are assumed to exist between two parallel lines that extend for a relatively long distance compared to their transmitting range. As a result of the discovery process, a small percentage of the deployed nodes are selected to be a part of a backbone, which can be used for efficient communication between the other nodes in the LSN. The protocol takes advantage of the linearity of the network in order to reduce the amount of exchanged control messages, reduce energy consumption, and increase scalability. Two different strategies for topology discovery are presented, and simulated in order to verify and compare their operation, and efficiency.**

Keywords: Ad hoc and sensor networks, wireless networks, routing, topology discovery.

I. INTRODUCTION

The research area of wireless sensor networks (WSNs) have received a lot of interest lately due to significant advancements in the field of electronics. This paved the road for the design of low cost, small, and capable sensing devices with increasingly higher processing, storage, sensing and communication capabilities. In addition, WSNs have a great potential for use in a large amount of existing and future applications in numerous areas such as environmental, civil, health care, military, monitoring, and infrastructure surveillance. In the latter category, a considerable number of the infrastructures that are monitored have a linear structure which extends over relatively long distances. This causes the wireless sensors to be aligned in a linear topology. New frameworks and protocols are needed to take better advantage of the linearity of the network structure in order to increase routing efficiency, enhance reliability and security, and improve location management. In a previous paper [1], we introduced a classification of LSNs from a hierarchical and topological points of views.

We propose a topology discovery algorithm for thick LSNs, where the sensor nodes are deployed between two parallel lines that can stretch for a long distance (e.g. tens or hundreds of kilometers). As a result of the proposed topology discovery algorithms, a small percentage of the deployed sensor nodes are selected to form a backbone network along the linear topology, which can be used to efficiently to route sensing data (collected from the surrounding nodes and transmitted to the nearest backbone node) along the linear network to the sink or sinks located at the end of the network or network segment.

Some researchers studied the characteristics of one-dimensional ad hoc networks. Diggavi et. al. studied the characteristic of wireless capacity with the existence of mobility in one-dimension [2]. Ghasemi et al. provided an approximation formula for the connectivity probability of one-dimensional ad hoc wireless networks [3]. Miorandi et al. analyzed the connectivity issue in one-dimensional ad hoc networks using a queuing theory approach [4]. On the other hand, many researchers have investigated topology control (TC) techniques in wireless ad hoc networks. In [5], Santi et al. present a survey of these algorithms, which have the primary goal of reducing energy consumption, and radio interference. In [6], Ramanathan et al. study the optimization problem of creating a desired topology by adjusting the transmit power of the nodes. In another paper [7], the authors study power assignments to maintain fault tolerance in wireless devices and present algorithms which can be used to minimize power while maintaining $k-edge$ connectivity with guaranteed approximation factors. In [8], a topology discovery algorithm for WSNs is presented. The algorithm determines a set of nodes which can act as cluster heads in the network. In [9], Wang presents an overview of the different types of topology algorithms for multidimensional WSNs that have been proposed in research.

The algorithms that are mentioned above are primarily designed for multi-dimensional WSNs. They do not take advantage of the predictable topology of a thick LSN in

This work was supported in part by UAEU Research grant 01-03-9-12/07

978-1-4799-6036-1/14 $31.00 © 2014 IEEE
DOI 10.1109/MASS.2014.129

order to optimize their performance. On the other hand, the algorithms presented in this paper are designed to take advantage of the linearity of the network in order to reduce topology discovery control overhead, and increase operation efficiency, and scalability.

A. Thick LSN Architecture

In thick LSNs, the sensor nodes are scattered in a 2-dimensional random form between two parallel lines which extend for a long distance. In this type of network, the sensor nodes have the responsibility of both sensing the information as well as routing it through their neighbor nodes along the “thick line" of sensor nodes to finally reach the sink node at the end of the network. In this case, the sensor nodes have sensing, aggregation, compression, as well as routing responsibilities.

The thick LSN topology can be present in many applications such as when the WSN is responsible for monitoring a geographic area. For example, the network can have the responsibility of monitoring international borders between countries [10] and detect illicit activities. Such activities can involve border crossings by smugglers of different illegal goods or substances, military crossings by individuals, or vehicles, etc. The inexpensive sensors can be deployed by throwing them from an airplane moving at a constant but low speed or an unmanned aerial vehicle (UAV). The dropped sensors end up in a semi-random geographic form and could follow a linear structure. The sink nodes can also be deployed at various locations and are separated by some specified average distance. This deployment of the sink nodes can be done in many different ways. They could also be thrown from a low-flying airplane, placing them at locations which are separated by approximately the same average distance, or they can be installed [9] in a precise fashion by the network personnel if the terrain is easily accessible.

Some potential applications for linear sensor networks are the following:

- Above-ground oil, gas, and water pipeline monitoring.
- Underwater oil, gas, and water pipeline monitoring.
- Railroad/subway monitoring.
- Terrestrial border monitoring.
- Sea-coast monitoring.
- River monitoring.
- Other applications.

B. Why new architectures and protocols are needed?

There are many reasons why a new framework and architecture are needed for different categories of thick LSNs.

- *Speed-up route the route discovery and maintenance:* Network protocol design take advantage of the linear nature of the network that significantly increase the efficiency of the route discovery and maintenance processes.
- *Reduce control overhead and bandwidth utilization for route discovery:* Due to the fact that there is a prior knowledge of the linear nature of the network topology, the route discovery algorithms, can be better adapted and focused in order to reduce the number of control message exchanges that are used in the route discovery process. For example, in route discovery, a directional approach can be used to progressively discovery nodes in one direction, and not waist time, energy, and control overhead to consider nodes in the opposite direction of the discovery process.
- *Increased routing fault tolerance and reliability:* More specialized protocols for thick LSNs would have the ability to take advantage of the structure and achieve significantly increased reliability. Towards this end, several solutions are presented in [11].
- *Reduce control overhead for route maintenance:* The number of messages that are used in the route repair and maintenance process is significantly reduced due to the prior knowledge of the linear nature of the topology. Therefore, it is not necessary to do a global or even local flooding of maintenance control messages in order to discover alternative nodes in all directions to replace failed nodes in the route. Consequently, only nodes in a certain "direction" can be considered. Also, the selection of the replacement node/s can be done in a more efficient and effective manner to meet certain desired criteria such as short hop to save energy or longer hop to reduce message end-to-end delay.
- *Increased efficiency of location management:* In many applications, location management of nodes is important. Location management algorithms might rely on some GPS-capable nodes to be used as reference points for other less expensive nodes, which do not have this feature.

The rest of the paper is organized as follows. Section II presents the topology discovery algorithm for thick LSNs. Section III provides simulation results and analysis of some aspects of the discovery process, and section IV concludes the paper.

II. TOPOLOGY DISCOVERY ALGORITHM

A. LSN definitions

Before we can discuss the various topology discovery algorithms in greater detail, it is important to define some basic and necessary definitions that will be used. It is worthy of noting that although our discovery algorithm focuses on the the nodes in one segment, in the often occurring case of very long LSNs, the network can be extended to contain multiple segments in order to provide more efficient and reliable coverage as discussed in [1]. This section presents these definitions along with the various parts in the LSN model used in this paper. We define the following parameters:

- *Network primary and secondary edges:* These are the edges of the LSN. The primary edge is where the initial discovery message is started and the secondary is the opposite edge.
- *Forward and backward directions:* The forward direction is from the primary to the secondary edge and the backward direction is the opposite one.
- *Forward nearest neighbor (FNN) and backward nearest neighbor (BNN):* FNN_x is the neighbor node to a node

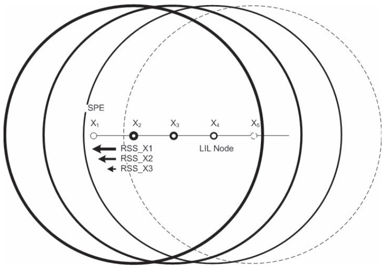

Fig. 1: Illustration of the reception of the acknowledgement message from x_2, x_3, and x_4 with varied RSS values in response to the neighborhood discovery message sent by x_1.

x, which is in its forward direction in the discovered backbone, and BNN_x is the one in the backward direction.

- S: number of segments in the network.
- L_s: physical segment length.
- L_n: physical network length.
- *Node ID*: For each node, there is a unique ID.
- *Node Location*: In our simulation, the nodes are generated randomly in the area of $[0, 0] \times [10, 1]$. Each node is assigned a location.
- *Neighborhood*: In the simulation setting, we use the received signal strength (RSS) model, as indicated in Figure 1, to measure the strength of a communication signal. We define the neighborhood as nodes that are within the transmission range of each other. We define the neighborhood as the area where the RSS strength is stronger than a threshold θ; in that case, we say that the two nodes are neighbors for each other. As is well known, RSS is related to the transmission energy and distance. We will use θ as the threshold of distance under a given energy in this paper.
- *Node Type*: When all of the nodes are scanned, we classify the nodes into some types: (1) *Relay node*, which is a part of the discovered backbone, and is used to relay data. (2) *Sensing node*, which only acts as sensing node, but not relay node. (3) *Intersection node*, which collects the data from sensing nodes. (3) *End node*, the edge of the available path.
- *Distance*: For each pair of nodes, the distance between the two nodes is defined as the Euclidean distance on 2-dimensional space
- *Medium Access Control (MAC) Protocol*: The MAC that is used is assumed to be the carrier sense multiple access with collision avoidance (CSMA/CA) protocol. However, other MAC protocols can also be used with the appropriate adjustments.

B. Node Distribution and Density

As mentioned above, the nodes are generated randomly in the area of $[0, 1] \times [0, 10]$, with various densities. Here, we define the density, represented by d, as the number of nodes in the area of square with a length equals to 0.1. Figure 2 shows the blocks. We designate the blocks that have the same x-coordinate as a $layer$. This provides additional information, which is used for data propagation.

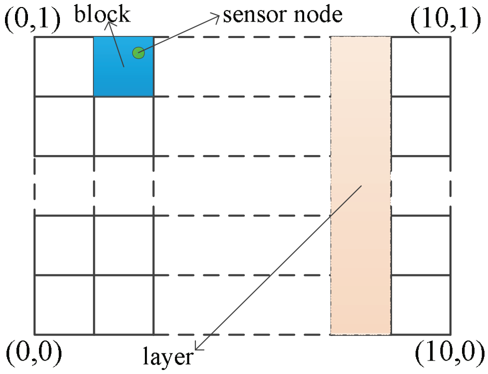

Fig. 2: Illustration of the area partition, the area is split to blocks, the number of sensor nodes in each block is known as *density*.

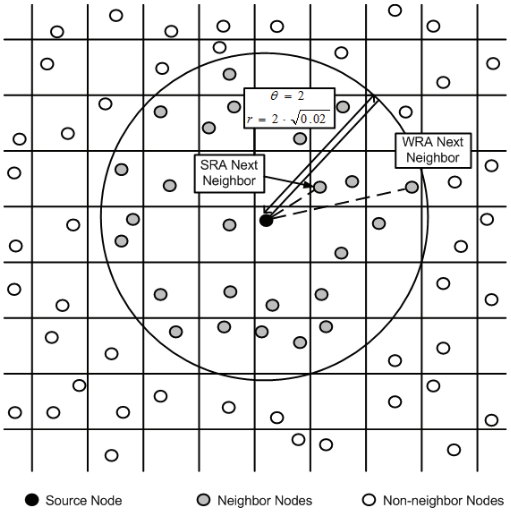

Fig. 3: Node neighborhood and next-hop selection. Here: $d = 1$.

C. Transmission Range

We use θ to indicate the transmission range r. Specifically, we let $r = \sqrt{0.02} \cdot \theta$, where $\sqrt{0.02}$ is the length of the diagonal of one block. To guarantee that there is a feasible path from the source sensor node to the destination sensor node, θ should be no less than 2. In that case, $r = 2 \cdot \sqrt{0.02}$, which means that the transmission range of the node covers two full blocks diagonally. However, we can always find a feasible path, even when $\theta = 1$.

D. Next-Hop Neighbor Selection

According to the performance requirements, there may be various backbone selection approaches. We will present two different stratgies.

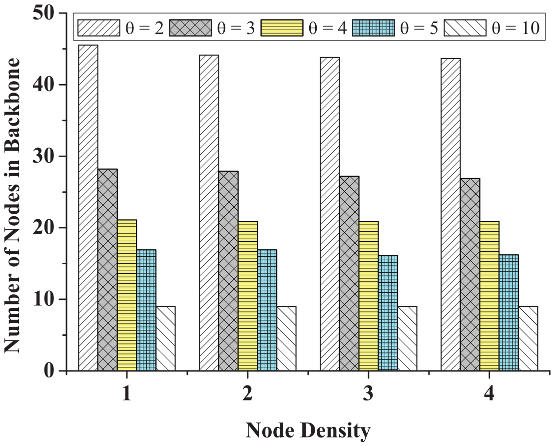

Fig. 4: Backbone selection: threshold and node density. Strategy: shortest hop

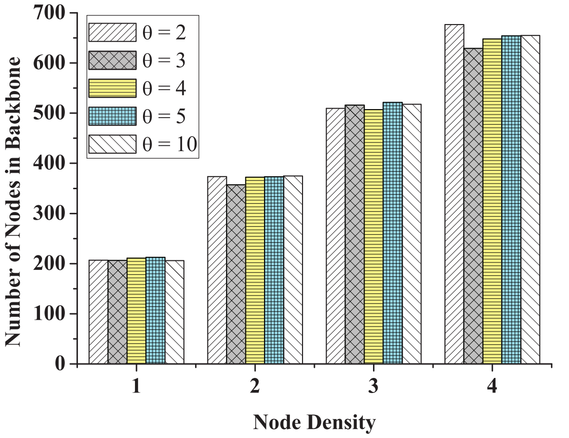

Fig. 5: Backbone selection: threshold and node density. Strategy: strongest link

1) Weakest RSS (WR) Strategy: In this strategy, the next-hop neighbor with the weakest RSS is selected. On average, this approach leads to the selection of the neighbor that is farthest from the current node with the most progress towards the sink. Consequently, the data can be transferred from the source node to destination node (sink) as fast as possible leading to lower end-to-end delay. However, the energy consumption of each node in backbone would be large due to the longer transmission range.

2) Strongest RSS (SR) Strategy: In this strategy, each node selects the neighbor with the strongest RSS as the next-hop node. Generally, this leads to the selection of nodes with lower distance from the transmitting node. In addition, the transmission range can be reduced, which results in increased energy savings. However, this comes at the price of increased end-to-end delay.

In order to avoid the selection of nodes in the backward direction, we take the layer into account in both strategies. This means that each sensor includes its layer information in its broadcasted messages. Consequently, each sensor node does not select the neighbors with lower layers as the next-hop node.

E. Discovered Backbone Caching

Once the discovery message reaches the end node, the latter sends a DCOMP (Discovery Complete) message. This message is unicast back to the first node along the discovered nodes in the backbone. As DCOMP message is propagated, each intermediate node in the backbone, can do one of the following actions, depending on the routing strategy that will be used subsequently:

1) Partial Backbone Caching: Another name for this caching type is *k-neighborhood caching*. The node only caches k neighbor node IDs in the *forward* direction and k neighbor node IDs in the *backward* direction. Later on, when the routing of the data is performed, the node can use this information to route any data it receives to the next-hop neighbor. For example, if node i receives a data packet from node $i-1$, which is located in its backward direction (i.e. its BNN node), it will normally route this data to node $i+1$, which is located in its forward direction (i.e. its FNN node). In the case of failure of node $i+1$, node i can take advantage of the linearity of the structure and re-route the data to overcome this failure using one of three different strategies that are briefly discussed in a subsequent section in this paper.

2) Full Backbone caching: Each backbone node caches the entire list of nodes in the backbone. Using this strategy, more advanced techniques can be applied for the routing process. For example, this way, a node can forward messages in either direction in order to reach the sink by satisfying different criteria, depending on the application's quality of service (QoS) requirements such as end-to-end delay, and bandwidth. For example, a node can cache accumulated past data on end-to-end delay experienced by packets in each direction and choose the one with lower delay. The overall packet delay is a summation of all of the delay types experienced by the packets including queueing delay, transmission delay, processing delay, and back-off delay. In addition, if a MAC protocol using a sleep strategy is deployed, then the sleep delay would also be significant and should be considered in the routing strategy. On the other hand, another approach might aim at lowering the end-to-end energy consumption by the packet to reach a sink, or provide load balancing for energy consumption of the backbone nodes in either direction. Each of these strategies would require each node to cache appropriate information about the nodes in both directions in its routing table in order to achieve the required optimization. We currently do not use these extensions in our protocol. However, we intend to consider the implementation of some of these techniques in the future.

F. Routing in Thick LSNs Using the Discovered Backbone

As mentioned earlier, the nodes can partially or fully cache the IDs of the nodes in the discovered backbone in order to use them for routing data to the sinks using different routing strategies. In a previous paper, [11], we discuss three different routing strategies that can be used in LSNs in order to transmit data from the nodes to the sinks and overcome intermediate node failures. These strategies are: Redirect Always (RA), Jump Always (JA), and Smart Redirect and Jump (SRJ). Having the information about the discovered backbone in its cache, a backbone node will normally send its data to its neighboring node to the "nearest" sink. However, if the message encounters a failed node on its way, it can use one of the above three algorithms to overcome this failure.

1) The RA Algorithm:: A node normally routes its data to the neighbor that is in the direction of the nearest sink. However, if that node fails, then the sending node will transmit its data to the neighbor that is in the opposite direction towards the alternative sink. It is worthy of noting that detecting the failure of the next-hop node can be done through various means such as a periodic $HELLO$ message mechanism, or acknowledgement of data message transmission. The number of hops needed to reach the opposite sink is typically larger resulting in added end-to-end delay and overall increased energy consumption. However, this provides a simple and efficient alternative to dropping the message and declaring a network disconnection. This process is only possible due to the priori knowledge of the linearity of the network and the discovered backbone.

2) The JA Algorithm:: In this case, we assume that the nodes have the capability of extending their transmission range by increasing their transmission power. In this case, when a node detects a failure of its next-hop neighbor to reach the nearest sink, it increases its transmission range to reach the following backbone node. This mechanism allows the routing process to overcome node failure by a simple increase in the transmission range due to the linear nature of the network. This increased reliability comes at the cost of increased energy consumption. However, the overall end-to-end delay is not significantly affected. This added reliability can be desirable in some critical applications that cannot tolerate added delay or data loss.

3) The SRJ Algorithm:: This algorithm combines the features of both the RA and JA algorithms. Using SRJ, when a node wants to transmit a data message, it forwards it towards the nearest sink through the next-hop backbone node in that direction. However, if node failures are encountered, then it calculates the amount of total energy that would be consumed for all of the hops needed to reach the nearest sink, E_n, and the total energy needed to reach the alternate sink in the opposite direction, E_a. The node then forwards the data message to the next-hop neighbor that is in the direction that requires the least energy. This approach provides for a reduction of total energy consumption and extended network lifetime. It requires full caching of the backbone information by the nodes. In addition, the cached information can include other parameters that can be used in the routing decision such as node residual energy, individual hop transmission success rate, and total delay. The SRJ algorithm can provide improved network performance. However, this comes at the cost of increased routing complexity and node memory requirements.

The choice among the three mentioned algorithms depends on the particular LSN application that is used, and the QoS requirements of the associated network traffic.

III. ANALYSIS AND SIMULATION

In this seciton, we provide an evaluation of the proposed algorithms for backbone discovery in thick LSNs. We examine the algorithms from the following aspects:

1) The total number of hops in the discovered backbone from the source sensor node at one end of the LSN to the destination node (sink) at the other end. This is done with a given transmission range and a predetermined RSS threshold.
2) How the threshold affects the discovery of the backbone.
3) How the node density affects the discovery of the backbone.
4) How the node selection strategy affects the discovery of the backbone.

We conduct extensive simulations based on various parameter settings. The results shown in Figure 4 indicate how the number of hops from the source to the destination changes along with the various thresholds and node densities. In this simulation, the target is to select a backbone with minimal hops to reach the destination node. Hence, the neighbor node with weak RSS strength is selected. In this Figure, we can see that the density has a reduced effect on the minimal number of hops from source to destination. This is because the neighbor with lower RSS strength and longer distance, is selected for each hop. This provides more opportunities to quickly reach the destination, though there are more neighbors when the density is large. For the effect of the threshold θ, we set the transmission range to a larger value resulting in an ability for the signal to reach more distant nodes with each hop. This leads to a lower number of hops in the backbone. Consequently, we see that the number of hops is inversely proportional to the value of the threshold. However, this comes at the cost of higher energy consumption resulting in an earlier expected failure of the discovered backbone.

Figure 6 shows an example of the backbone. The selection of the next-hop neighbor is based on the WR strategy, which leads to the minimal number of nodes in the backbone. In the corresponding simulation, we let $d = 1$. Each black point in the figure is a sensor node, and each red circle (or blue square) represents one node of the backbone. The left-most node is selected as the source node, while the right-most one is selected as the destination. For the red line, we let $\theta = 3$, which resulted in 28 sensor nodes to be included in the backbone. For the blue line, we let $\theta = 10$, and this resulted in 9 sensor nodes in the backbone. As we can see, when the transmission is large enough, the path tends to be a straight line.

Figure 5 shows the results when we use another node selection strategy. We select the neighbor node with the strongest RSS signal, which indicates that the links in the backbone have the strongest capacity to transfer data, but the number of hops will increase. From the figure, we see that the node transmission range has less of an effect on the number of nodes in the backbone. This is because the neighbor node with a shorter distance from the source is selected, though the number of neighbors will be large when the threshold increases. This implies that we can adjust the transmission range to be shorter,

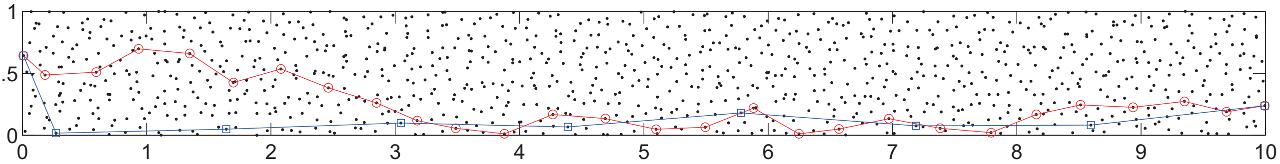

Fig. 6: Backbone selection. Here, we let $d = 1$. For the red line case, $\theta = 3$. For the blue line case, $\theta = 10$. For each case, the shortest path is selected with minimal relay nodes.

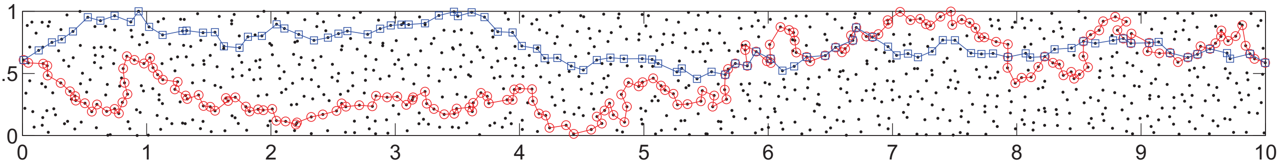

Fig. 7: Backbone selection: when the RSS signal is considered. Here $d = 1$, $\theta = 1.2$. Red line: each node selects the next-hop relay node with layers no less than it. Blue line: each node selects the next-hop relay node with layers higher than it.

which is an effective way to save energy consumption for the nodes in backbone, and reduce interference. However, when the density increases, there are more nodes in the area, and more nodes will be selected according to the selection strategy. In fact, having a large number of nodes in the backbone is unnecessary, which motivates us to let more nodes switch to the sleep mode to save energy.

Figure 7 shows an example of the backbone when the SR strategy is used. As explained earlier, with this approach, the node with the strongest RSS is selected as the next-hop relay neighbor. In this simulation, we let $d = 1$ and $\theta = 1.2$, which is an efficient way to save energy consumption in the backbone nodes since the transmission power is exponentially proportional to the transmission range. The red-colored backbone in the figure, which includes 195 nodes in this case, corresponds to the selection strategy where the next-hop relay node is in a layer that is more than or equal to that of the sending node. On the other hand, the blue-colored backbone, which includes 100 nodes, corresponds to the selection strategy where the next-hop relay node is in a layer that is strictly more than that of the sending node. The latter selection strategy is a frequently used method to select one node from each cluster, which corresponds to one layer in our paper.

It is worthy of noting that the results above are for the case of thick LSNs. However, the same algorithms work for the case of thin LSNs. The thickness of the LSN varies according to the requirements of the corresponding application.

In addition, according to the relationship of energy consumption, P, and transmission range, r, where $P \propto r^{\alpha}$, $(2 \leq \alpha \leq 4)$, we can easily get the result for how the energy is consumed as the transmission range r changes; we use θ to represent r in our paper.

IV. CONCLUSIONS

After stating some of the applications for thick LSNs in order to motivate the research, we presented a topology discovery algorithm for this type of WSNs. As a result of the discovery process, some nodes are selected to construct a backbone inside a network segment in a thick LSN. This backbone can later be used for efficient routing of messages between the nodes and the sinks that can be located on either or both ends of the network. For long thick LSNs, which might extend for tens or hundreds of kilometers, multiple segments separated by sinks can be used in order to provide added efficiency, reliability, and scalability to the network and the associated routing protocol. Two different strategies for backbone discovery are presented and analyzed. These strategies depend on the criteria for next-hop neighbor selection using the received RSS by the forwarding node. The proposed algorithms can constitute a good foundation for further future research in this important area.

REFERENCES

[1] I. Jawhar, N. Mohamed, and D. P. Agrawal. Linear wireless sensor networks: Classification and applications. *Elsevier Journal of Network and Computer Applications (JNCA)*, 34:1671–1682, 2011.

[2] S. Diggavi, M. Grossglauser, and D. Tse. Even one-demensional mobility increases adhoc wireless capacity. *IEEE Transactions on Information Theory*, 51(11), November 2005.

[3] A. Ghasemi and S. Nader-Esfahani. Supporting aggregate queries over ad-hoc sensor networks. *IEEE Communications Letters*, 10(4):251–253, April 2006.

[4] D. Miorandi and E. Altman. Connectivity in one-dimensional ad hoc networks: a queuing theoretical approach. *Wireless Networks*, 12(5):573–587, September 2006.

[5] P. Santi. Topology control in wireless ad hoc and sensor networks. *ACM Computing Surveys (CSUR)*, 37:164–194, March 2005.

[6] R. Ramanathan and R. Rosales-Hain. Topology control of multihop wireless networks using transmit power adjustment. *In Proc. IEEE Infocom 2000*, pages 404–413, March 2000.

[7] M. Hajiaghayi, N. Immorlica, and V. Mirrokni. Power optimization in fault-tolerant topology control algorithms for wireless multi-hop networks. *in: Proc. MobiCom*, September 2003.

[8] Badri Nath B. Deb, S. Bhatnagar. A topology discovery algorithm for sensor networks with applications to network management. *IEEE CAS workshop*, September 2002.

[9] Y. Wang. Topology control for wireless sensor networks. *Chapter 5, Wireless Sensor Networks and Applications, Springer*, pages 113–140, 2008.

[10] A. D'Costa, V. Ramachandran, and A. M. Sayeed. Distributed classification of gaussian space-time sources in wireless sensor networks. *IEEE Journal on Selected Areas in Communications*, 22(6):1026–1036, August 2004.

[11] I. Jawhar, N. Mohamed, K. Shuaib, and N. Kesserwan. An efficient framework and networking protocol for linear wireless sensor networks. *Accepted for publication in The Ad Hoc and Sensor Wireless Networks Journal, Old City Publishing, London, UK*, 2008.