

RESEARCH

Open Access

# Adaptive neighbor-based topology control protocol for wireless multi-hop networks

Zeeshan Hameed Mir<sup>1</sup> and Young-Bae Ko<sup>2\*</sup>

## Abstract

Topology control protocols have been proposed to construct efficient network topologies with several design goals, e.g., network-wide connectivity, minimal energy cost, symmetry, lower nodal degree, and therefore higher spatial reuse or lower interferences. Neighbor-based topology control protocols are simple and assume that each node in the network is connected to its  $k$  least-distant neighbors. There have been several empirical and theoretical research efforts that recommend a network-wide optimal value of the local parameter  $k$ . However, since most of the design goals often run against each other the suggested lower and upper bounds on the values of  $k$  are not sufficient to provide a controllable trade-off among various design goals. In this article, an adaptive neighbor-based topology control protocol is presented where the neighboring nodes collaborate and provide feedback on the network connectivity to decide on their respective transmission ranges. Since every node adaptively adjusts its number of neighbors, the parameter  $k$  acts as a performance knob to choose a set of backbone nodes and to form a hierarchical topology structure consisting of symmetric links. Through extensive simulation-based study, it is shown that the value of  $k$  can be tuned to generate fully connected network topologies while offering an efficient trade-off among various design goals.

**Keywords:** topology control, neighbor-based, wireless multi-hop networks

## 1. Introduction

Topology control [1,2] leads to a simpler network topology with several design goals such as network-wide connectivity, minimal energy cost, symmetry, lower nodal degree, and therefore higher spatial reuse or lower interferences. In *neighbor-based* [3,4] topology control protocols, each node connects to its  $k$  least-distant neighbors. The neighbor-based protocols are often characterized by their simplicity and the use of minimum amount of information needed by nodes to construct the network topology. However, finding an optimal value of  $k$  such that some or most of the design goals are achieved has been a challenging task.

The network topology induced by setting a lower value of  $k$  is either not fully connected and/or consists of asymmetric links. On the other hand, the recommended upper bound on the value of  $k$  causes significant redundancy in nodal degree which is a measure of

the spatial reuse and interference. There have been several research efforts both empirically and theoretically to find an optimal, network-wide value of the local parameter  $k$ . However, since most of the design goals such as connectivity, energy cost, symmetry, and nodal degree often run against each other the suggested lower and upper bound on the values of  $k$  are not sufficient to provide a controllable trade-off among various design goals.

While the focus of most of the previous investigations rested on finding the number of neighbors that are necessary for connectivity [5,6] and on how different network models and design goals influence the value of  $k$  [3,4], they miss the potential of collaboration among neighboring nodes where nodes provide feedback on the network connectivity to decide on their respective transmission ranges. Nodes start with connecting to the least-distant neighbors, check for local network connectivity information and adjust the number of neighbors to achieve an efficient trade-off among various design goals.

This study describes an adaptive neighbor-based topology control (ANTC) protocol that constructs fully

\* Correspondence: youngko@ajou.ac.kr

<sup>2</sup>School of Information and Computer Engineering, Ajou University, Suwon, Republic of Korea

Full list of author information is available at the end of the article

connected network topologies while it addresses number of design goals efficiently. The basic idea is to select a subset of nodes that serve as the network backbone and to form a hierarchical topology structure consisting of symmetric links. The process of backbone node selection is carried out in a distributed manner without requiring global network connectivity knowledge. The proposed ANTC protocol runs in three phases. First each node discovers its one-hop neighbors. Next the sink or base station node initiates the topology construction (TC) phase by broadcasting a control message. Among other attributes, the control message contains information on local neighborhood connectivity. On receiving the control message, each node checks for network connectivity requirements and adjusts its number of neighbor accordingly. The control message is then re-broadcasted exactly once by means of controlled flooding. This process is realized by mean of node coloring algorithm where initially all nodes are WHITE in color. During the TC phase nodes change their color according to the feedback information on network connectivity. The backbone nodes are colored BLACK whereas the other nodes are either RED or BLUE. Both RED and BLUE nodes are linked symmetrically with the BLACK node, while BLUE nodes are also candidate BLACK nodes. In the last phase, topology maintenance is performed in order to avoid disconnected networks due to any control message loss or collision.

The simulation results validate our claim that the proposed ANTC protocol shows a versatile performance for the given set of design goals. The following factors make our work extremely useful for wireless ad hoc and sensor networks.

- A practical and distributed protocol that construct fully connected network topologies consisting of symmetric links. The local topology parameter  $k$  act as a performance knob that offers an efficient trade-off among several conflicting design goals.
- A hierarchy of backbone nodes are selected using the communication overhead of at most  $3n$  messages, where  $n$  is the total number of nodes in the networks. Lower message complexity makes the ANTC protocol scalable and suitable for energy and resource constraint sensor networks.
- The hierarchy of backbone nodes implicitly setups the forwarding path towards the common sink or base station node.
- Through extensive simulation-based study the quality of the generated topology is evaluated against several criteria such as network-wide connectivity, minimal energy cost, higher spatial reuse, or lower interference by means of lower nodal degree.

The rest of this article is organized as follows. Section 2 gives a brief overview of the related work and expands

on the problem statement. Section 3 presents an ANTC protocol in detail which is followed by the performance evaluation which is given in Section 4. Finally Section 5 concludes the article.

## 2. Related work and motivation

### 2.1. Related work on topology control

In this section, we will discuss a number of neighbor-based topology control protocols found in the literature and comment on how they construct network topology with the desired set of designs goals.

In  $k$ -Neighbor [3,4] and XTC [7] protocols, each node creates an ordered neighbor list, which ranks all one-hop neighboring nodes with respect to their distance, energy, or link quality. Both protocols are simple to implement, localized, and communication efficient. The  $k$ -Neighbor protocol simply chooses the value of  $k$  to be 6 and 9 to achieve weak and strong requirements on connectivity, respectively. However, the final topology is constructed by keeping only symmetric links with fewer than  $k$  least-distant neighbors. In XTC, each node locally traverses the neighbor list in non-decreasing order of their ranks. At some arbitrary node  $u$  if a candidate node  $v$  can be reached through an intermediate neighbor  $w$  with better rank (i.e., link quality), then the link between node  $u$  and  $v$  is marked redundant and is therefore not included in the final topology.

There is a class of neighbor-based topology control protocols that tries to keep the number of neighbors within a certain minimum and maximum threshold values  $k_{\min}$  and  $k_{\max}$ . The example protocols include Local Information No Topology (LINT)/Local Information Link-state Topology (LILT) [8], MobileGRID [9], Novel Topology Control Protocol (NTC) [10], and Cooperative Nearest Neighbor (CNN) [11]. Each protocol consists of two phases which adaptively adjust the transmission ranges by including or excludes links from the final topology. Both phases are responsible for maintaining the nodal degree within certain configurable bounds. Mainly the algorithms differ in actually what triggers the range adjustment at each node. For example, in LILT, the absence/presence of routing updates determines whether the node is disconnected or not. LINT, on the other hand, periodically checks for number of active neighbors and tries to keep the neighbor count around a desired value. Likewise LINT, the MobileGRID protocol, utilizes the nodal degree for maintaining a specific contention index. The nodal degree value is preconfigured and depends on the node density, network area, and transmission range. NTC and CNN are especially designed to mitigate the network partitioning problem which usually arises when the wireless nodes operate at lower values of  $k$ . For uniform node distribution, the topology constructed by the local connections

can perform well. However, with clustered node distribution the NTC and CNN algorithms may result into partitioned network for the given lower bound of  $k$ .

The quality of a generated topology can be measure with respect to several design goals such as connectivity, energy cost, symmetry, spatial re-use and interference, etc. More recently, Banner and Orda [12] proposed a topology control protocol that caters for a wide array of design goals. The authors argue that the previous work concentrated more on finding nodes' transmission range by keeping in mind the rarely occurring worst-case scenarios. Most of the previous approaches towards topology control would certainly yield optimal performance for only a small subset of the design goals which are often conflicting in nature. Through both analytical and simulation studies they demonstrate that their protocol is capable of achieving *average performance* for a variety of topology related parameters simultaneously.

## 2.2. Motivation

In neighbor-based topology control protocols, the main focus is on finding a network-wide optimal value of the local parameter  $k$  that could result in an *efficient topology*, i.e., fully connected and energy minimal. To cater different network models and design goals, the value for  $k$  is often too large to achieve several conflicting design goals, simultaneously. Consider a 10-node network given in Figure 1 where dashed lines represent the asymmetric links and the solid lines represent the symmetric links. Here, node 1 is the sink node or base station to collect the data generated by all the other nodes. As illustrated in Figure 1d, it is not until  $k$  set to 4, that a fully connected network topology consisting of symmetric links is obtained. However, to account for worst-case scenarios the network operates at even higher values of  $k$ , i.e., 9 or  $(n - 1)$ . Clearly, the larger values of  $k$  result in significantly large number of redundant links and thus higher interferences among the neighboring nodes. Moreover, larger transmission ranges result in higher energy cost while having longer *forward progress* [10] and thus lower path length towards the final destination. Conversely, the Minimum Spanning Tree (MST) [13] generates sparser network topology (given in Figure 1f) which results in fewer links and lower energy consumption. However, the path length among source-destination pairs increases significantly.

For the recommended value of  $k$ , all the nodes are forced to have larger neighborhood because some of the nodes might have too few neighbors or asymmetric links. However, only a small subset of nodes is required to extend their number of neighbors provided that each node is aware of the local network connectivity for the selected value of  $k$ . For example in Figure 1c, each node establishes links to their three least-distant neighbors

(i.e.,  $k = 3$ ). However, there are only two pairs of nodes (8, 9) and (4, 6) that are linked asymmetrically. Considering distance as the metric to decide on which pair, nodes 4 and 6 may collaborate to increase their transmission range enough to have an efficient topology. Likewise in Figure 1b with  $k = 2$  and in Figure 1a with  $k = 1$  there are at least three and five pairs of nodes, respectively.

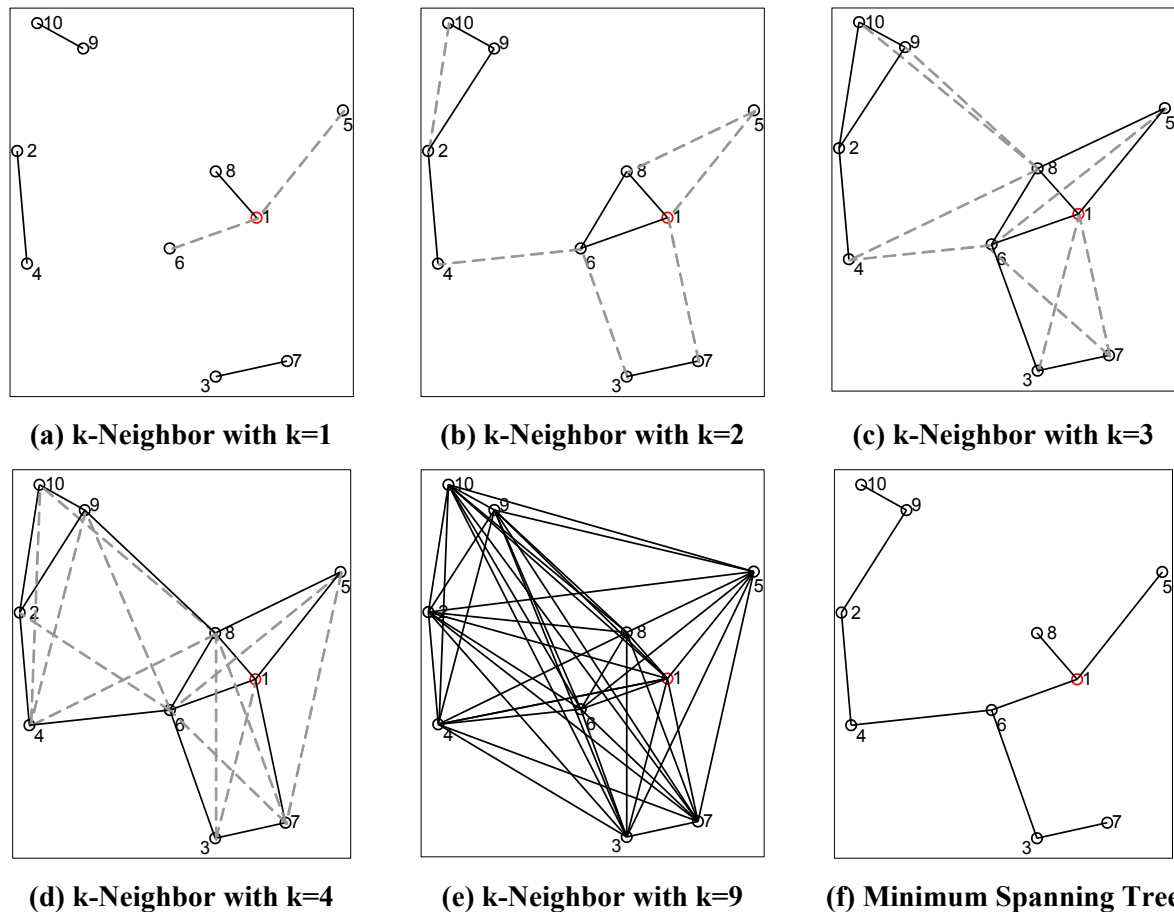
Interestingly, as the  $k$  increases, so does the number of symmetric links and as a result the number of node pairs decreases. It is also noteworthy that how the selection of node pairs could possibly effect the forward progress and nodal degree at each node. For example in Figure 1b for node 9, selecting node 8 instead of node 2, save half of the hop distance towards the sink. Therefore, the value of  $k$  certainly acts as a performance knob which can be tuned to achieve an efficient trade-off among various design goals concurrently. Moreover, instead of following a network-wide value of the local parameter  $k$ , all the nodes locally adjust their number of neighbors accordingly.

## 3. Adaptive neighbor-based topology control (ANTC)

Consider a wireless network comprising of  $n - 1$  ordinary nodes and a single sink or base station node (thus total  $n$  nodes). All the nodes are stationary and deployed randomly throughout the two-dimensional space. It is assumed that all the nodes are equipped with homogeneous, single RF radio with an omni-directional antenna and are capable of adjusting their RF output power at several discrete levels [14,15]. Furthermore, the underlying topology is fully connected at maximum transmission range  $Tx_{max}$ . The main objective of the ANTC protocol is to provide a controllable and efficient trade-off among various design goals such as network-wide connectivity, minimal energy cost, symmetry, and lower nodal degree for *any* given value of the local parameter  $k$ , with  $0 < k \leq n - 1$ . The proposed ANTC protocol consists of three phases.

### 3.1. Neighbor discovery phase

Initially, each node  $i$  announces its presence using MAC layer beaconing at  $Tx_{max}$ . A simple yet practically reliable one-hop broadcast mechanism [16] can be used to reduce the chances of a node remain undiscovered by its neighbors. The idea is to make sure that all the nodes receive the discovery message and construct a complete neighbor list. Upon successful receiving the ANNOUNCE message from node  $i$ , the neighboring node  $j$  estimates its distance to node  $i$ . An all neighbor list  $N_j^*$  is maintained which consists of two attributes, a unique neighbor identity and their distance. The



**Figure 1** Topology snapshot by applying  $k$ -Neighbor topology control protocol with  $k = 1, 2, 3, 4$ , and  $9$  for (a)-(e), respectively, and (f) MST. Symmetric links are represented by solid lines and asymmetric links are represented by dashed lines.

neighbor list is then stored in non-decreasing order of the distance. Algorithm 1 illustrates the above-mentioned procedure.

The higher cost of operating GPS-enabled nodes makes this option of acquiring the distance information nearly infeasible. Instead, techniques that utilize different physical measurements, e.g., Received Signal Strength Intensity (RSSI) [17] and Time of Arrival (ToA) [18] can be employed to estimate the distance between two nodes. The use of RSSI and ToA-based mechanisms often raised criticism by the researchers due to the distance measurement error which is usually caused by attenuation present in the atmosphere or non-line of sight (i.e., the obstacles). However, in the performance evaluation section it is shown that the ANTIC protocol achieves resilience to error in distance estimation at much lower cost than the previous work.

Although this article utilizes the estimated distance between two nodes, other metrics such as link quality, residual energy can also be combined.

### 3.2. TC phase

ANTIC is essentially a node coloring algorithm, which creates a single topology structure consisting of backbone nodes. The number of backbone nodes and other topology-related properties are decided based on the input parameter  $k$ . During the course of the TC phase, each node can have one of the four colors. Nodes modify their colors in response to the reception (or overhearing) of network connectivity information within the TC messages. As the TC phase proceeds, the nodes change their colors according to the following definitions. For ease of reference, Table 1 provides a list of parameters and algorithmic notations that are used throughout the article.

**Table 1 Algorithmic notation and data structures**

WHITE color	Ordinary node
RED color	$k$ -symmetric neighbor
BLUE color	$k$ -asymmetric neighbor
BLACK color	Backbone node
$n$	Total number of nodes in the network
$T_{x_{\max}}$	Maximum transmission range
$N_i^*$	Set of all one-hop neighbors of node $i$
$N_i^k$	Set of $k$ least-distant neighbors of node $i$
$Tx_i^k$	Transmission range of node $i$ required to reach the farthest neighbor in $N_i^k$
$d_{(i,j)}$	Estimated distance between nodes $i$ and $j$
$T_{\max}$	Maximum delay a node must wait, before initiating the next step
$T_{d, fwd}$	Forwarding delay for TC message
$T_{interval}$	Forwarding interval for TC message
$i.msgCache\_$	{ $nb.Identity\_;$ $nb.Distance\_;$ $nb.Color\_;$ $nb.Backbone\_;$ $nb.HopCount\_;$ } entry;

- **WHITE:** A node in its initial state which has not received and thus acted upon the information within the TC message.
- **BLACK:** The backbone nodes are colored BLACK, which guarantees symmetric link towards the common sink or base station node.
- **RED:** A RED colored node is *associated  $k$ -symmetrically* with a BLACK node. If a WHITE node  $j$  receives the TC message from node  $i$ , i.e. ( $j \in N_i^K$  and  $i \in N_j^k$ ), it turns into RED color. In other words, nodes  $i$  and  $j$  have symmetric links for the given value of  $k$ .
- **BLUE:** A BLUE node is *associated  $k$ -asymmetrically* with a BLUE or BLACK node. If a WHITE node  $j$  receives the TC message from node  $i$ , i.e. ( $j \notin N_i^k$  and  $i \in N_j^k$ ), it turns into BLUE color. In other words, node  $j$  has an asymmetric link with node  $i$  for the given value of  $k$ .

The sink or base station node initiates the TC phase by broadcasting the TC message at  $T_{x_{\max}}$ . The ANTC protocol exploits the broadcast nature of the wireless communication channel. Each node selects an appropriate set of neighbors by overhearing the ongoing communication among its neighboring nodes. The process of selecting next node to disseminate its TC message is bit a like peeling the onion layers inside-out. Nodes that are closer to the initiator are given higher priority as compare to the farther ones. The TC message is also required to maintain proper value of hop distance value towards the initiator. As the TC phase proceeds, the TC message disseminates in a controlled manner such that each node relays the TC message only once. Each node

$i$  maintains and updates the following attributes of the TC message.

- $i.Identity\_:$  Unique identity of node  $i$ .
- $i.Backbone\_:$  Identity of a neighboring node which guarantees symmetric link towards the common sink or base station node.
- $i.HopCount\_:$  Total distance in terms of number of hops from sink or base station to the current node.
- $i.Color\_:$  Color of node  $i$ .
- $k$ -Neighbor list ( $N_i^k$ ): A list of  $k$  least-distant neighbors of node  $i$ .

Following are the main highlights of the ANTC algorithm.

- Algorithm 2 illustrates the TC message attribute initialization at sink node  $i$ . The sink node turns BLUE, before it broadcast the TC message into its immediate one-hop neighborhood.
- Lines 10-33 describe the steps taken by node  $j$  upon receiving the TC message from node  $i$ . There are three courses of actions  $j$  may take depending on its current status.
  - From lines 10-18, node  $j$  verifies whether it has a  $k$ -symmetric or  $k$ -asymmetric link with node  $i$ . If  $i$  and  $j$  are linked  $k$ -symmetrically, the receiver WHITE node becomes RED. In case of  $k$ -asymmetric link, the receiver WHITE node turns BLUE otherwise the color remains unchanged. This part of the pseudo-code also performs two important steps, i.e., first a soft state is maintained for each sender node and secondly asymmetric links are eliminated during this process.



- Lines 19-26 are followed if  $j$  has not yet sent its TC message. Before a node broadcasts its TC message further into the network, it has to decide on two aspects, i.e., (1) the backbone node and (2) the TC message forwarding sequence. Later in this section, more details will be provided.
- Last, lines 27-32 are only executed, if  $j$  overhears the TC message from one of its neighboring node which has selected  $j$  as the backbone node. If the sender's information is not already in the  $N_j^k$ , it is then included and the transmission range is extended to reach the most distant neighbor in the  $k$ -Neighbor list.
- The topology maintenance phase, ensures that TC phase remain resilient to TC message lose.

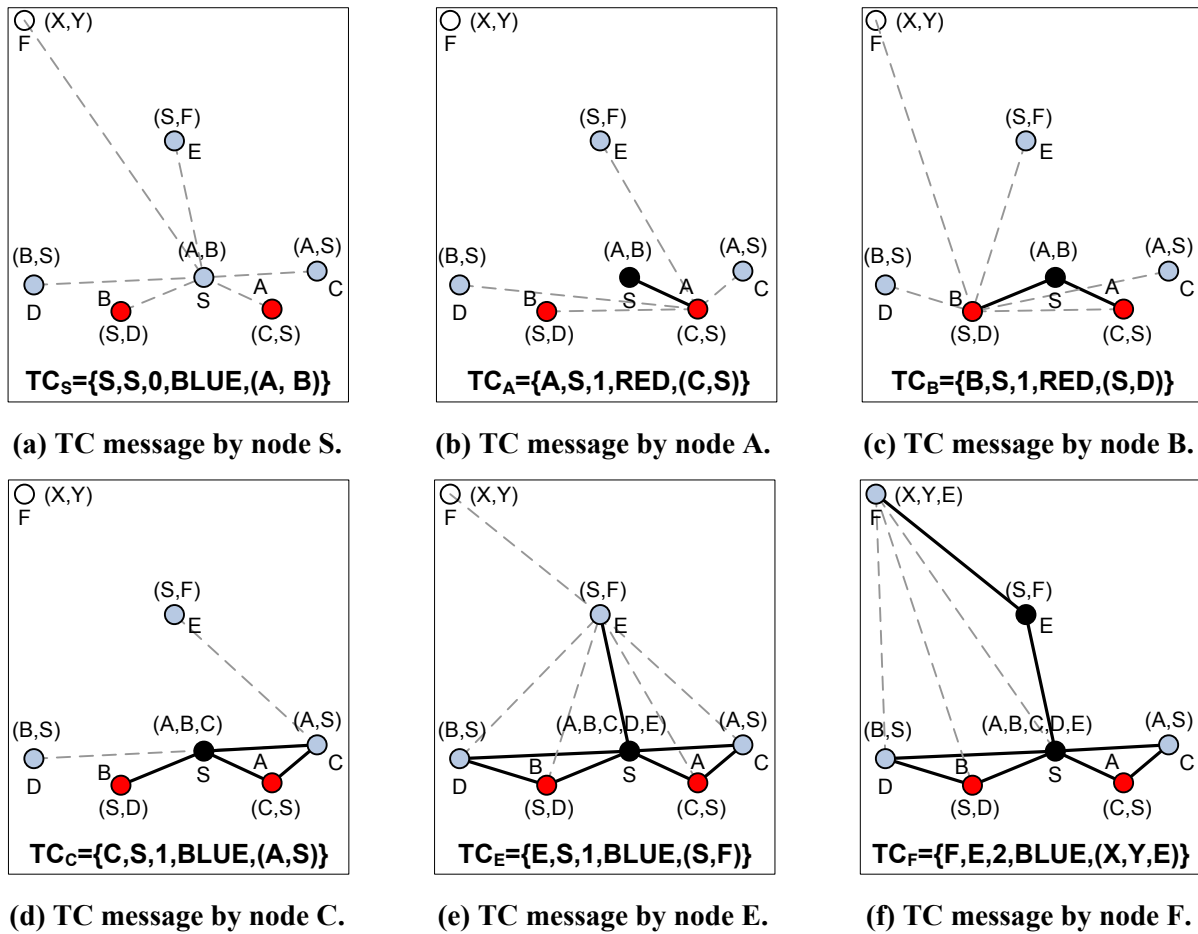
To make a single connected topology structure consisting of symmetric links towards a common sink, each node must select a backbone node. The intuition behind selecting a BLUE node as the backbone is very simple. First, for the given value of  $k$  the BLUE nodes are farthest from the sender node, which are also linked  $k$ -asymmetrically with another potential backbone node. Second, the fact that BLUE node would operate at larger transmission ranges therefore they give better performance trade-off between energy costs and hop distances. The use of distance information further helps in choosing better candidate among several BLUE nodes. Finally, in scenarios (especially for higher values of  $k$ ) where there exist fewer BLUE nodes, any least-distant neighbor which is minimal hops away from sink node is selected. In Algorithm 3, lines 1 through 11 describe the backbone node selection procedure. Lines 12-15 set the required TC message attributes before they are further broadcasted by the receiver node  $j$ . For example, the  $j$ .Backbone\_ attribute holds the selected backbone' identity, the hop count attribute gets an increment, and finally if nodes current color is WHITE, it changes into BLUE. Lines 16-19 simply update the  $k$ -Neighbor list and set the transmission range that is required to reach the farthest neighbor in  $N_j^k$ .

In ANTIC, each node decides backbone based on the neighbor's color, estimated distance, and hop distance towards the sink node. While the TC phase is in progress the exact path length from the sink to the current node cannot be known in advance. Therefore, the node color and the hop distance information are included within the TC message. The value in the hop count attribute is incremented each time a node forwards the TC message further into the network. To calculate correct accumulated value of the hop count, the forwarding sequence of TC message requires special consideration.

A distance-based message dissemination approach is utilized which introduces forwarding delay for each node based on their distance from the previous node. The delay heuristic is implemented with the help of a simple timer (see Lines 19-26, Algorithm 2) and it works in two ways. First, the TC message is flooded in a controlled manner, i.e., it ensures that each node disseminates the TC message only once. Up on receiving the first TC message each node set the timer value based on the parameter  $T_{d, fwd}$ , given at line 20. The timer value is set proportional to the distance between the receiving node and the previous node. Once set, the timer value cannot be re-initiated by overhearing of any subsequent TC message. Second, the coloring of the nodes and the increment in hop count value require a proper forwarding sequence, i.e., a node that is at  $n$ th hops distance from the previous node forwards the TC message before a node  $n + 1$ th hops away. Thus, the cumulative effect is to give higher priority to the nodes that are closer to the pervious node.

In practice, the actual transmission time of a TC message is determined by the combination of the timer value and the time required by functions performed by a particular medium access control protocols. For instance, in contention-based MAC protocols, random back-off is performed for each message to make sure its successful reception at the receiver. Consider an example where the forwarding interval ( $T_{interval}$ ) is 1 time unit and the nodes'  $T_{x_{max}}$  is set to 1.5 units. Let, nodes  $i$  and  $j$  are at the same distance from the previous node  $S$  (1 distance unit), they decide to forward the TC message at time 0.67 ( $= 1 \times 1/1.5$ ). However, in order to break the tie between nodes  $i$  and  $j$ , the actual sending can be differentiated by the CSMA/CA mechanism at the MAC layer. More importantly, nodes are not required any form of strict time synchronization for distance-based forwarding to work properly. Distance-based forwarding can lead to slightly longer time. However, since the TC phase last for very small fraction of overall network operational time, therefore its effect can be consider negligible on the network performance.

To exemplify the proposed heuristics, a simple topology given in Figure 2 is considered. In current example,  $k$  is set to 2 and the neighbor list is shown in parenthesis besides each node  $i$ . The sink node  $S$  initiates the TC phase by first turning its color to BLUE and then broadcasting a TC message at  $T_{x_{max}}$  (represented by the dotted lines). The contents of the TC messages are given within each figure. On reception, all the neighboring nodes calculate the timer value to send their respective TC messages. Based on the estimated distance from the sender node, each node will calculate its forwarding sequence. In this example, the forwarding sequence is represented by the alphabetical order given as node IDs.



**Figure 2** Diffusion process of  $TC_i$  message for  $k = 2$ ; with four types of information, of node  $i$  i.e.,  $\{i.Backbone, i.HopCount, i.Color, \text{ and } N_i^2\}$  at  $Tx_{max}$ . (a) TC message by node S. (b) TC message by node A. (c) TC message by node B. (d) TC message by node C. (e) TC message by node E. (f) TC message by node F.

Figure 2a shows the coloring of the nodes after the TC message is received by the neighbors from the sink node S. Since nodes A and B have 2-symmetric links with S, they turn RED. Similarly, nodes C, D, and E will turn BLUE because of their 2-asymmetric links. Whereas node F's color remains WHITE, because node F is neither linked  $k$ -symmetrically nor  $k$ -asymmetrically with node S.

In order to construct a fully connected backbone, each node has to be linked symmetrically with a special backbone node so that it guarantees a symmetric link towards the sink. For RED nodes, the selection of backbone node is simple, since they already have symmetric link with a BLUE node. Figure 2b illustrates the process where node A selects S and send the TC message. The decision regarding backbone selection is conveyed to the other nodes while they overhear the ongoing communication. Once node S finds out that node A has selected

it as the backbone node it changes its color to BLACK. In Figure 2c, node B repeats the same procedure.

For the given value of  $k$ , a node might have no  $k$ -symmetric neighbors thus resulting in either asymmetric links or network partitioning. Nodes C, D, and E represent the first case where they are associated  $k$ -asymmetrically with node S. These nodes select node S as the backbone node (the least-distant BLUE node) and convey this decision in their respective TC messages. On overhearing the TC messages, node S extends its neighbor list and sets its transmission range to the level that is required to reach the farthest neighbor in the neighbor list. The selected backbone node turns BLACK (if not already BLACK color). In Figure 2d, nodes C and D perform the above procedure with node S, followed by node E in Figure 2e. The nodes retain links with their  $k$ -symmetric neighbors. Node F represents the second case where the network is partitioned between nodes F and

*E*. In Figure 2f, the WHITE node *F* selects node *E* as the backbone node. Node *F* is associated *k*-asymmetrically with two other neighbors *X* and *Y* (not shown in the figure). Before the TC message is sent further into the network, node *F* turns BLUE and increments the hop count value by 1. In scenarios where there exists no BLUE node, any least-distant node which is minimum hops away from the sink is selected. The hop count value is incremented as the TC message is disseminated across the network. The forwarding sequence of the TC message ensures that path length value is incremented properly.

The proposed ANTC protocol results in a single tree-like structure, where RED and BLUE nodes are associated symmetrically with the set of BLACK nodes. The BLACK nodes are linked directly with relatively larger transmission ranges, thus forming a connected backbone. In order to control the number of backbone nodes and the associated links, the value of *k* is utilized as a performance knob. The performance evaluation section shows that the value of *k* provides an efficient trade-off among various design goals. The other advantage of topology constructing phase is that it also constructs an implicit data forwarding hierarchy consisting of backbone nodes towards the sink or base station node.

### 3.3. Topology maintenance phase

This section describes a scenario where the TC message is lost either due to collision or unsuccessfully reception at the receiver. The lost of TC message is significant, because it contains information regarding the backbone node that a potential BLACK node could not receive. The given scenario often leads to asymmetric links where the selected backbone node remains unable to extend its transmission range up to the requesting node. Each node *i* checks whether it has received/overheard the TC message from all its one-hop neighbors. If not, then it broadcasts the Topology Repair Request (TRREQ) message at  $Tx_{max}$ . On reception, only those nodes *j* would respond with the Topology Repair Reply (TRREP) message that has a *k*-asymmetric links with the node *i*. Finally, on receiving the TRREP message node *i* extend its transmission range to include node *j*. Algorithm 4 describes the above procedure.

## 4. Performance evaluations

### 4.1. Simulation environment

To evaluate ANTC and other neighbor-based topology control protocols, extensive simulations are performed using NS-2 [19] simulator. In the simulation study, the network size *n* is varied from 100 to 250 in increments of 50 and 500 nodes. The network sizes represent sparse, moderate, and densely populated networks. All the nodes are placed randomly over a  $1000 \times 1000$  m<sup>2</sup>

area with the sink or base station node positioned at the center. The maximum transmission range  $Tx_{max}$  is set to 250 m. In our simulations,  $T_{interval}$  is set to 1 s and  $Tx_{max}$  is set to an arbitrary constant large enough to avoid interruption with the ongoing TC phase.

For a comparative study, following neighbor-based topology control protocols are considered where all the resultant topology instances are fully connected and consisting of symmetric links.

- **MST:** Most of the topology control protocols idealize topologies generated by the MST algorithms. However, the requirements of global information at some arbitrary common node make the MST algorithm less feasible to implement especially for large-scale sensor networks.
- **CNN with *k* set to 5:** In CNN, all the nodes are connected to their  $k = k_{min} = k_{max}$  least-distant neighbors. The CNN protocol is evaluated with  $k = 5$  (given as *k*-CNN).
- ***k*-Neighbor with *k* set to 9:** The results obtain for *k*-Neighbor protocol suggest that to achieve strong requirements on connectivity the value of *k* must be set to 9. The final topology is referred as *k*-Neighbor.

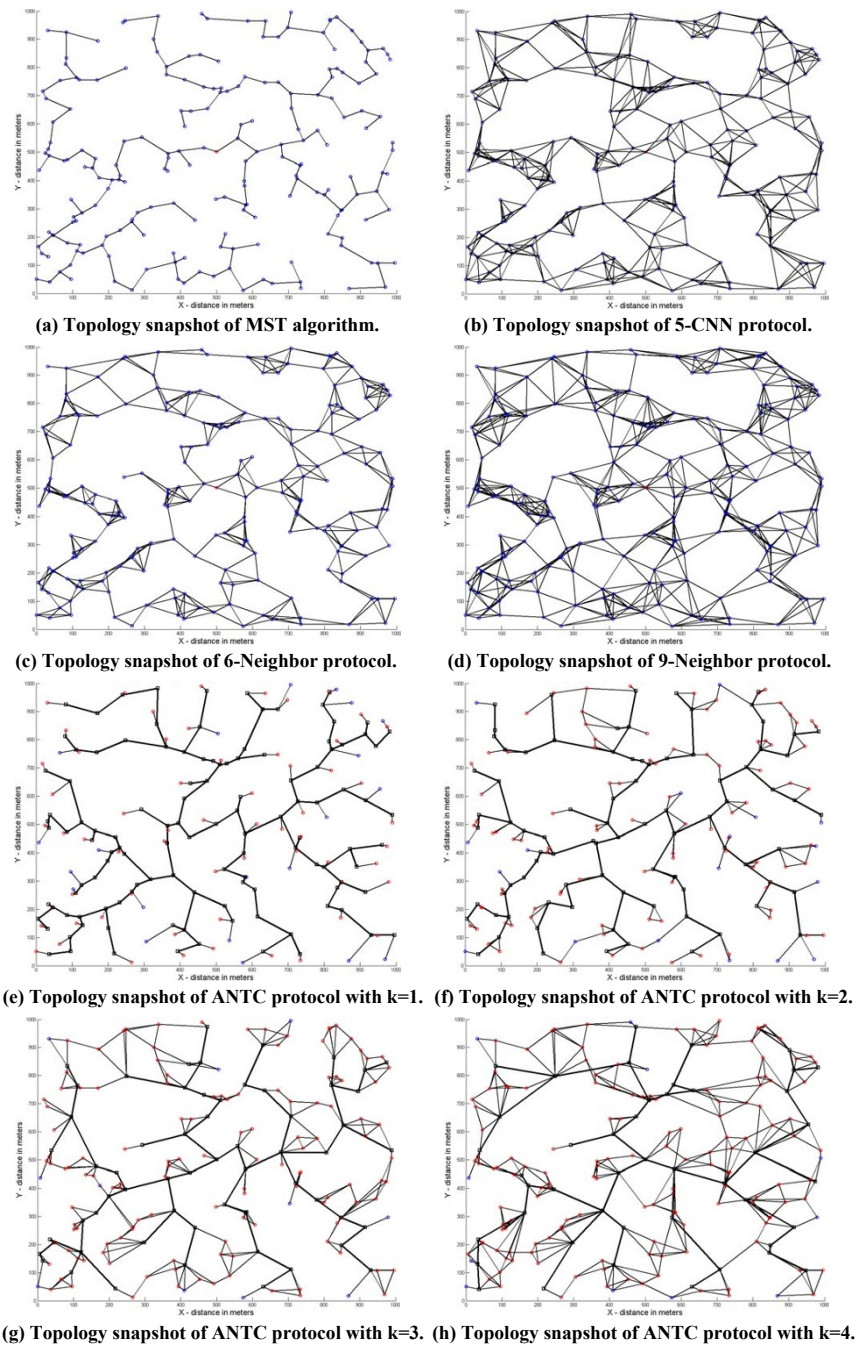
Following three metrics are used for performance evaluations. The final values are obtained by averaging 100 different simulation runs over 100 different randomly generated network topologies. All the measurements are averaged over the network sizes.

- **Total energy cost** is defined as  $\sum_{i=0}^n (Tx_i^k)^\alpha$ , where  $Tx_i^k$  represents the transmission range assigned to reach the *k*th neighbor in the neighbor list at the end of individual topology control protocol and  $\alpha$  is the *path loss constant* typically with a value between 2 and 6 [20].
- **Nodal degree** is defined as the number of direct neighbors. We consider both logical and physical nodal degrees, where later is consider as a better measure of expected contention.
- **Path length** is defined as the number of hops towards the common sink or base station node. A shortest path algorithm is executed over the topology controlled network to measure path length between a node and the sink.

### 4.2. Simulation results

Figure 3 illustrates the topology snapshots obtained by several topology control protocols. In *k*-Neighbor, the sample topologies are generated for  $k = 6$  and 9, in CNN the value for *k* is set to 5 and finally the ANTC

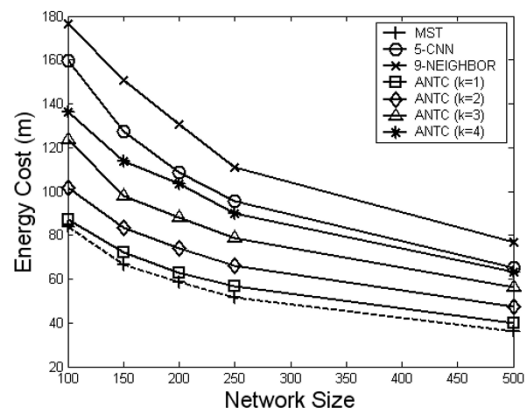




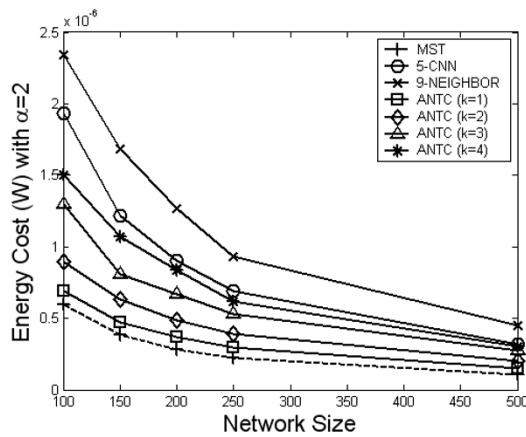
**Figure 3** Sample topology snapshot with 200 nodes: (a) Topology snapshot of MST algorithm, (b)  $k$ -CNN with  $k = 5$ , (c), (d)  $k$ -Neighbor with  $k = 6$  and 9, (e)-(h) ANTC protocol with  $k = 1, 2, 3$ , and 4. Solid black lines represent backbone links.

protocol is executed with  $k = 1, 2, 3$ , and 4. The network consists of 200 nodes. The topology generated by the MST algorithm is the sparsest. For the neighbor-based protocols, the parameter value  $k$  determines the topology resolution and other topological properties. In ANTC as the value of  $k$  increases, a denser

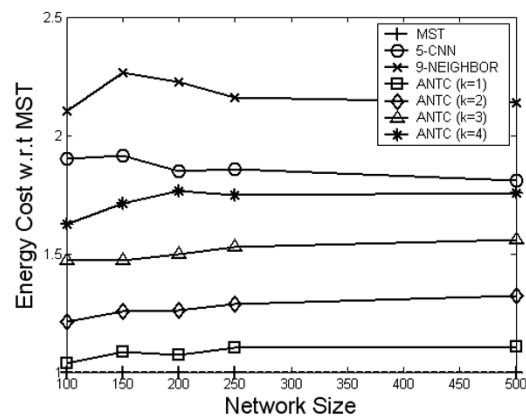
communication graph starts to appear. Intuitively,  $k$  acts as a *performance knob* which can be used to tune various design goals such as energy cost, nodal degree, and path length. For *any* given value of local parameter  $k$ , the proposed ANTC protocol constructs fully connected network topologies consisting of symmetric links.



(a)



(b)



(c)

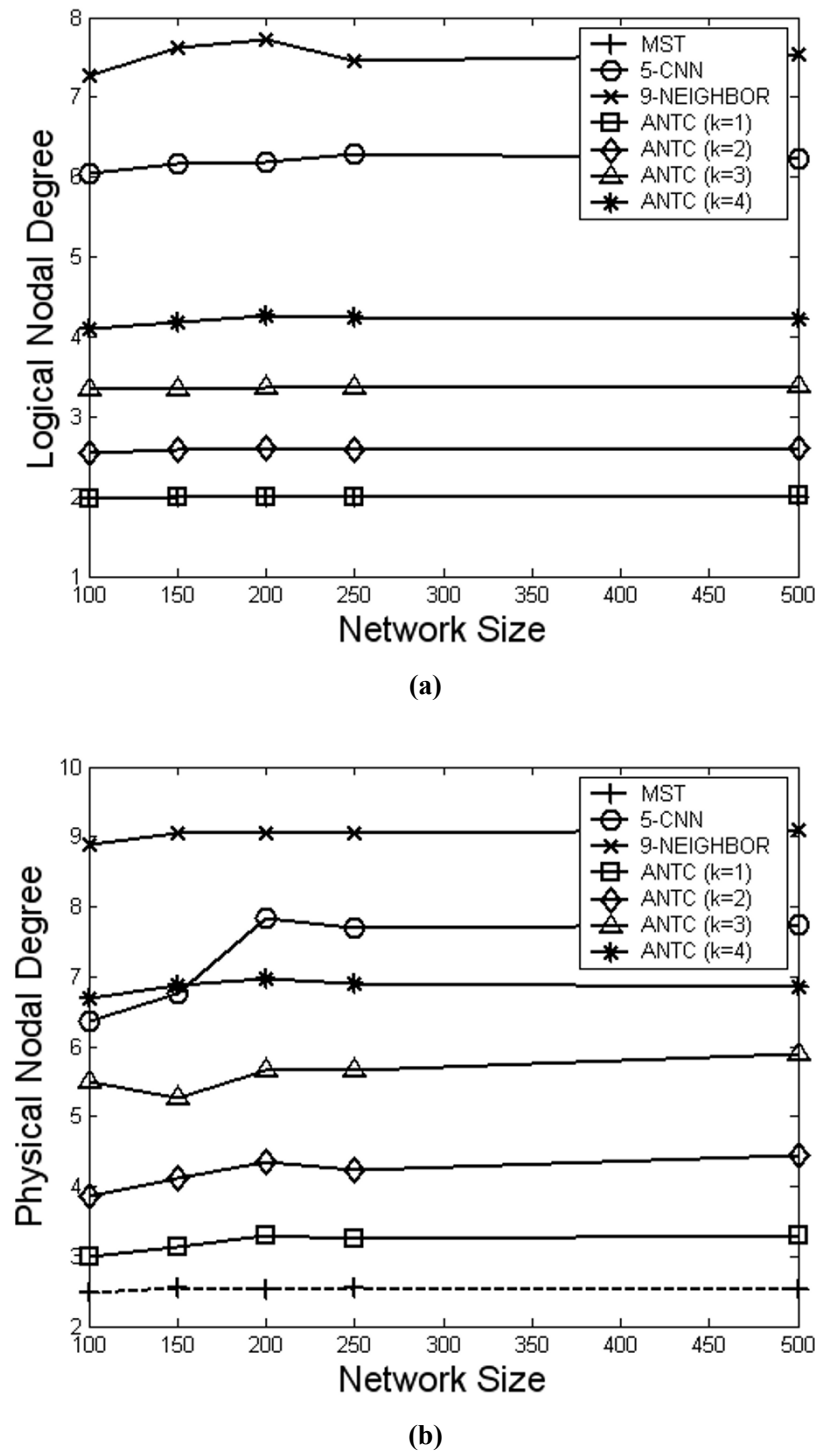
**Figure 4** Energy cost comparison among MST,  $k$ -CNN,  $k$ -Neighbor, and the proposed ANTC protocols for different network sizes. (a) Transmission range assignment (m), (b) with respect to MST, and (c) for  $\alpha = 2$  (W).

#### 4.2.1. Performance comparison with various network size $n$

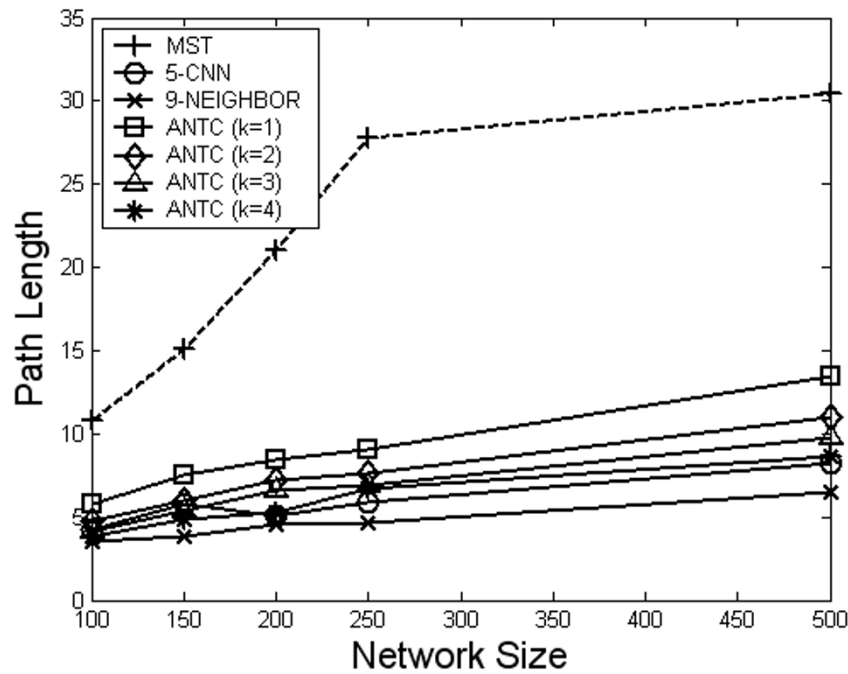
Figure 4a reports on the energy cost for all the protocols. Since the energy cost measurements are averaged over network sizes, the per node energy cost decreases as the network size increases for any given value of  $k$ . MST generated topologies are optimal in terms of energy cost. For the lowest value of  $k = 1$ , ANTC performed comparable to that of MST. The cost of maintaining a fully connected topology is highest for the 9-Neighbor protocol. Figure 4b shows the energy cost normalized with respect to MST. Overall the energy cost increases with the increase in the value of  $k$ . For the intermediate value of  $k$  the ANTC performance lies right between the two extremes of MST and 9-Neighbor. Compared to the 9-Neighbor protocol when  $n = 100$ , ANTC with  $k = 1, 2, 3$ , and  $4$  provides an improvement of 102, 73, 43, and 30%, respectively. For all the network sizes, ANTC with  $k = 4$  performed reasonably better than the 5-CNN. Figure 4c plots the trends for all the other protocols with path loss constant value set to 2, i.e.,  $\alpha = 2$ .

Figure 5a,b shows the average logical and physical nodal degrees for all the protocols. For any given network size, the MST and ANTC with  $k = 1$  performed identically. The logical and physical nodal degrees increase consistently for all the value of  $k$ . Generally, as the  $k$  and  $n$  pair grows the network topology becomes increasingly denser. Since logical nodal degree is considered as the lower bound to the physical degree, the physical degree is always higher than the logical degree. The 9-Neighbor protocol yields almost three times higher nodal degree as compared to ANTC protocol with  $k = 1$ . Whereas the logical and physical nodal degrees of 5-CNN protocol are nearly twice of the ANTC protocol with  $k = 1$ . In ANTC, only the nodes with asymmetric links tend to extend their transmission ranges to make a symmetric links with BLUE node, whereas the RED nodes are already associated with the least-distant backbone nodes resulting in lower average logical and physical degree.

Figure 6 shows average path length for the MST,  $k$ -CNN,  $k$ -Neighbor, and the ANTC protocols. MST generated topology results in highest number of hops. Despite that the ANTC protocol with  $k = 1$  is comparable with MST in terms of energy cost and nodal degree, almost half of the number of hops are required to reach the common destination. In ANTC, generally, the path length decreases with the increase in  $k$  because of the corresponding increase in transmission ranges results in more forward progress towards the sink or base station. For the lower values of  $k$ , the difference in



**Figure 5** Nodal degree comparison among MST, *k*-CNN, *k*-Neighbor, and the proposed ANTC protocols for different network sizes. (a) Logical, and (b) physical nodal degrees.



**Figure 6** Path length comparison among MST,  $k$ -CNN,  $k$ -Neighbor, and the proposed ANTC protocols for different network sizes.

path length is considerable. For example, for the first four values of  $k$  the percent decrement in hop counts is within the range of 6 to 20%. For most of the network sizes, the 9-Neighbor protocol performed better, however at the expense of higher energy cost and nodal degree. The performance of 5-CNN and the ANTC protocol with  $k = 4$  is comparable. The main reason being that in ANTC, a node only selects a backbone node that is closer in terms of hop count and distance towards the common sink.

These results demonstrate that the proposed ANTC protocol achieves an efficient trade-off among various design goals. The value of  $k$  acts as a performance knob which can be tuned to construct network topologies with variety of different topological properties. For example, ANTC performed comparable with the optimal MST topologies in term of energy efficient and nodal degree while maintaining shorter path lengths.

#### 4.2.2. Impact of $k$ and $n$ pair with respect to node color

To further study the energy cost distribution, a closer look on nodes with respect to their color is presented. Figure 7a,c,e shows that generally the BLACK nodes operate at higher transmission range as compared to the RED and BLUE nodes for the given values of  $k$  and  $n$  pair. This is due to the fact that the backbone nodes have to extend their transmissions ranges beyond the current value of  $k$  to construct a fully connected network topology. However, despite this, the BLACK nodes operate at as high as 70% of  $Tx_{\max}$  for the sparsely node

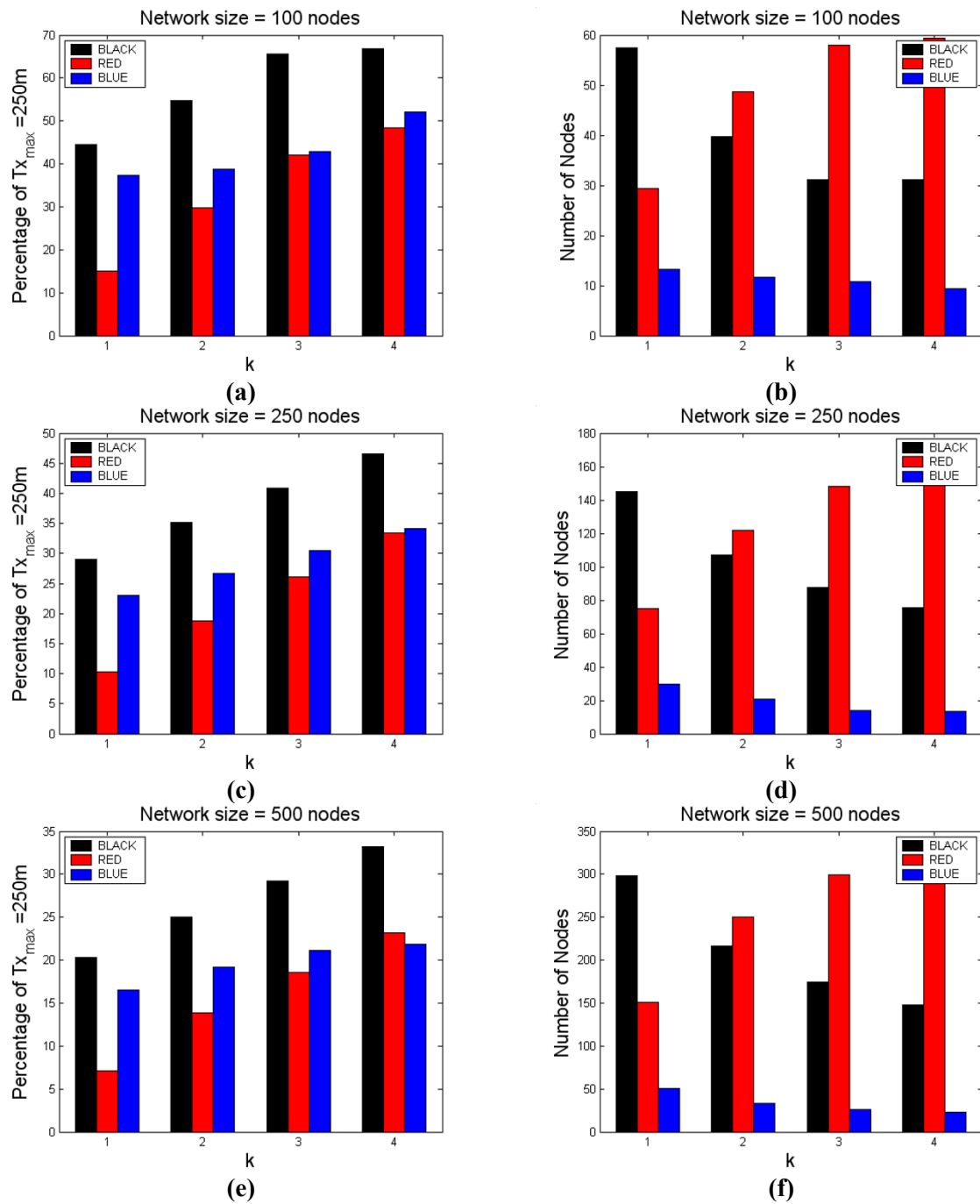
deployment (Figure 7a). The energy cost of backbone node decreases to 45 and 35% of  $Tx_{\max}$  as the network size grows from moderate (Figure 7c) to highly dense network (Figure 7e), respectively. The energy cost of RED nodes varies between 10 and 50% of  $Tx_{\max}$  depending on the values of  $k$  and  $n$  pair. An increase in  $k$  results in corresponding increase in transmission range, therefore more distant nodes become  $k$ -symmetric to the sender node. Whereas most BLUE nodes have an average transmission range of 30% of  $Tx_{\max}$ . Since BLUE nodes are  $k$ -asymmetric nodes, therefore they have to extend their range to create symmetric links.

Since, the BLACK nodes dominate the energy cost, it is therefore desirable to minimize the number of BLACK nodes in the network. Interestingly, as given in Figure 7b,d,f, the number of BLACK nodes decreases as the value of  $k$  increases for all network sizes. Thus, allowing less number of nodes to extend their transmission ranges. The number of RED nodes on the other hand increases. This is mainly because at higher values of  $k$ , the nodes tend to operate at higher transmission range, thus leading to more neighbors nodes to turn RED, i.e.,  $k$ -symmetric nodes. Finally, the numbers of BLUE nodes are nearly 10% of the network size  $n$ , which gradually decreases as the value of  $k$  increases.

#### 4.2.3 Impact of distance measurement error (DME)

It is typically assumed that the distance estimations are error free which is far less realistic than the actual





**Figure 7** (a, c, and e) Percentage of  $Tx_{max}$  assigned to and (b, d, and f) number of BLACK, RED, and BLUE nodes for different values of network size  $n$  and  $k$  pair.

situation. Specially, the assumption that the distance can be estimated by mean of measuring certain physical phenomena (i.e., RSSI and ToA), requires extra considerations. In  $k$ -Neighbor TC protocol, the recommended value of  $k$  in presence of distance measurement error is even higher than the actual value needed to achieve the

strong connectivity requirement for variety of network sizes. For network sizes  $n \in [100, 500]$ , the preferred value of  $k$  is found to be 9 and 10, for ToA and RSSI errors, respectively [3]. Resilience to the error in distance measurement is achieved by simply connecting to the even farther neighbor in the neighbor list.

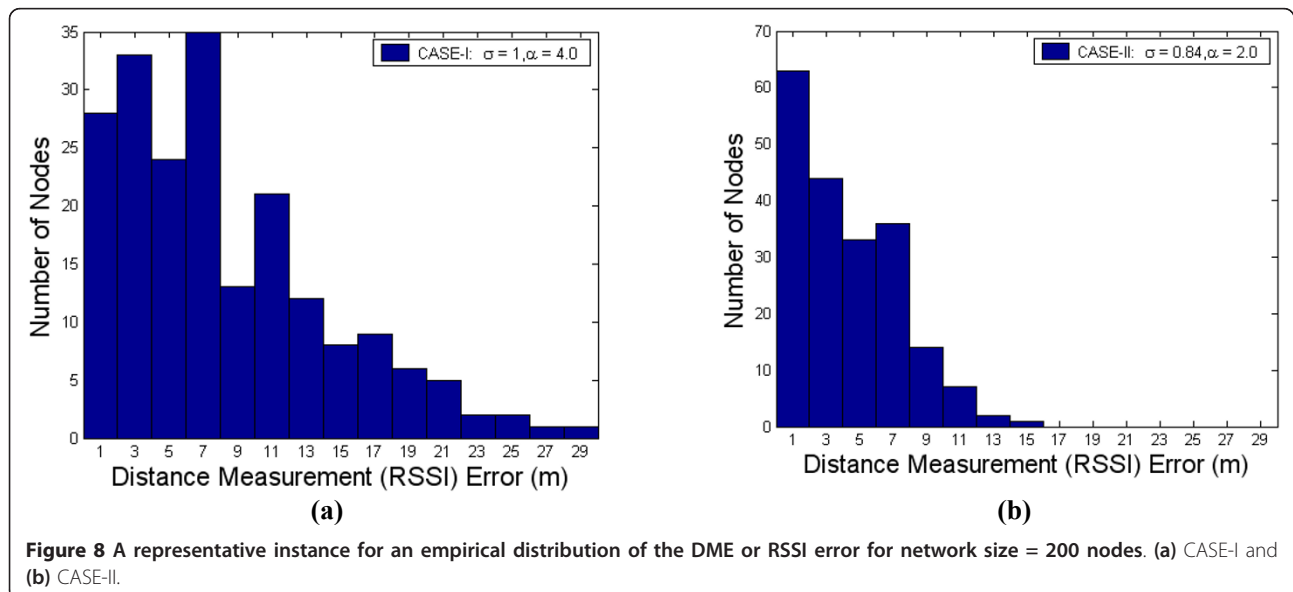
To show the impact of DME on proposed ANTC protocol performance, the error model given in [3,21] is used with two different settings of the parameter values. The distance measurement between two nodes  $i$  and  $j$  is given by  $\hat{d}_{(i,j)} = d_{(i,j)} + \text{RSSI}^e$ , where  $\hat{d}_{(i,j)}$  and  $d_{(i,j)}$  are defined as the measured and correct distances, respectively. The factor  $\text{RSSI}^e$  is the ranging error (also DME or RSSI error) is given as  $\frac{X\sigma}{d_{(i,j)}(1 - 10\frac{\alpha}{10\alpha})}$ , where  $X\sigma$  is a zero-mean Gaussian random variable with standard deviation  $\sigma$  and  $\alpha$  is the path loss constant. For CASE-I, the parameter values are  $\sigma = 1$  and  $\alpha = 4$  and for CASE-II we set  $\sigma = 0.84$  and  $\alpha = 2$  in the RSSI error equation (Equation 5 in [21]). CASE-I and CASE-II yield almost 70% of the distance estimations errors within 6 and 10% of the correct distances, respectively. Figure 8 shows one of the instances for an empirical distribution of distance measurement or RSSI error, (a) CASE-I and (b) CASE-II.

A transmission range shorter than the actual would certainly results in asymmetric links or network to partition. Like  $k$ -Neighbor protocol, ANTC also let all the nodes extend their transmission ranges up to the farther neighbor in the neighbor list. However, unlike the  $k$ -Neighbor protocol, where all nodes follow a network-wide value of the local parameter  $k$ , in ANTC nodes operate at different number of least-distant neighbors according to their color for the given value of  $k$ . Each node  $i$  maintains an all neighbor ordered list  $N_i^*$  with respect to the distance. The farthest node in the  $k$ -Neighbor list  $N_i^k$  is located at some  $k$ th entry in the list  $N_i^*$ , which may not be the actual  $k$ th neighbor in  $\hat{N}_i^*$

list. The list  $\hat{N}_i^*$  holds faulty distance measurements, obtained by applying the above-mentioned error model during the post-processing phase. To accommodate error in distance estimation, each node adjusts its transmission range to reach the  $\hat{k}th = k + \eta$ th neighbor in  $\hat{N}_i^*$ , where  $\eta$  is a small positive integer constant set according to the desired application requirements. The value of  $\eta$  is increased until the specified requirement on the connectivity is achieved i.e., more than 95% of the nodes are connected in a single structure consisting of symmetric links.

As suggested in [3], since the neighbor-based protocols work on the notion of “nearest” neighbors and the fact that error cause by ToA are positive, we have not included results for ToA errors. The simulation results given in Table 2 are obtained for the network size of 200 nodes with  $Tx_{\max}$  set to 250 m. For both CASE I and CASE II, the preferred value of  $\hat{k}th$  depends on the parameter  $k$ . For the lower value of  $k$ , the  $\eta$  is higher because the nodes tend to operate at lower transmission ranges and thus have fewer links. Consequently a discrepancy in distance estimates more often results in partitioned network topology. In ANTC, on the average an increment of 2 (i.e.,  $\hat{k}th = 5$ ) and 3 (i.e.,  $\hat{k}th = 6$ ) is recommended, for CASE-I and CASE-II, respectively. Despite this, the final value of  $\hat{k}th$  is considerably lower than the preferred value given in  $k$ -Neighbor protocol (i.e.,  $\hat{k}th = 10$ ).

It is noteworthy that as the value of  $k$  increases, the network finds sufficient redundant links to maintain the prescribed requirements on the connectivity. Complying



**Table 2 The value of  $\hat{k}th$  for CASE-I and CASE-II with RSSI error**

$k$	$kth$ without DME (i.e., no error)	$\hat{k}th$ CASE-I	$\eta$	$\hat{k}th$ CASE-II	$\eta$
1	3.28	5.28	2	6.28	3
2	4.34	6.34	2	6.34	2
3	5.67	6.67	1	6.87	1
4	7.84	8.62	1	8.79	1

with the results obtained for the  $k$ -Neighbor protocol, our protocol also requires no further stretch in transmission range for higher value of  $k$ . Since, the solution for distance measurement error take into account the farthest neighbor in the list, the backbone nodes and their associated nodes relationship remains intact. Once again the proposed ANTC protocol exhibits better performance to accommodate distance measurement error at comparatively much smaller cost.

## 5. Conclusion

Topology control protocols have been utilized in wireless multi-hop networks to achieve variety of different design goals. In this article, an ANTC protocol is presented, aimed at constructing efficient network topologies. The proposed ANTC protocol exploits the potential of collaboration among neighboring nodes where nodes provide feedback on the network connectivity information to decide on their respective transmission ranges. Based on the local connectivity information each node selects a backbone node that guarantees hierarchical topology structure consisting of symmetric links. The process of backbone node selection is carried out in a distributed manner without requiring global network connectivity knowledge. To evaluate the performance of ANTC protocol against other protocols, we have performed extensive simulation based study. The results demonstrate that ANTC achieves an efficient trade-off among various design goals. The value of  $k$  acts as a *performance knob* which can be tuned to construct network topologies with up to 100% and threefold improvements in terms of energy cost and average nodal degrees, respectively, while maintaining shorter path lengths.

### Algorithm 1: Neighbor discovery phase

*/\*All  $n$  nodes exchange the ANNOUNCE messages\*/*  
*For node  $j$  receives ANNOUNCE message from node  $i$ :*

*/\*Node  $j$  upon receiving an ANNOUNCE message from node  $i$  stores one-hop neighbor identity and estimated distance in neighbor list  $N_j^*$ \*/*

*receiveANNOUNCE ( $i.Identity\_$ ) {*  
1.  $N_j^*.insert (i.Identity\_ , d_{(i,j)});$

*/\*Sort neighbor list, in non-decreasing (ASCENDING) order of distance\*/*

2.  $N_j^*.sort (N_j^*.Distance\_ , ASC);$   
3. }

### Algorithm 2: TC phaseFor the sink node $i$ :

*/\*Initialize\*/*

1. Set:  $i.Identity\_ = Sink;$   
2. Set:  $i.Color\_ = BLUE;$   
3. Set:  $i.Backbone\_ = Sink;$   
4. Set:  $i.HopCount\_ = 1;$   
5. Set:  $i.isBroadcast\_ = 0;$   
*/\*Broadcast the TC message at  $T_{x_{max}}$ \*/*  
6. Call: sendTC ( $i.Identity\_ , i.Color\_ , i.Backbone\_ , i.HopCount\_ , N_i^k$ );

*/\*Upon receiving the TC message\*/*

7. Set:  $i.isBroadcast\_ = 1;$   
8. Wait:  $T_{max}$  units;  
9. Goto: Topology maintenance phase;  
*For the other node  $j$ :*

*/\*Upon receiving the TC message\*/*

10. receiveTC( $i.Identity\_ , i.Color\_ , i.Backbone\_ , i.HopCount\_ , N_i^k$ ) {

*/\*Add TC message attribute into the msgCache\_ data structure\*/*

11.  $j.msgCache_.insert (i.Identity\_ , d_{(i,j)} , i.Color\_ , i.Backbone\_ , i.HopCount\_);$

12. if ( $j \in N_i^K$  and  $i \in N_j^K$ ) {

13. if ( $j.Color\_ == WHITE$ )

14. Set:  $j.Color\_ = RED;$  }

15. elseif ( $j \notin N_i^K$  and  $i \in N_j^K$ ) {

16.  $N_j^K.erase (i.Identity\_);$

17. if ( $j.Color\_ == WHITE$ )

18. Set:  $j.Color\_ = BLUE;$  }

19. if ( $j.isBroadcast\_ == 0$ ) {

20. Set:  $T_{d,fwd} = T_{interval} \times d_{(i,j)} / T_{x_{max}}$  units;

21. Initiate: diffusionTimer( $T_{d,fwd}$ ) ;

22. On expire:

23. Call: findBackbone();

24. Call: sendTC ( $j.Identity\_ , j.Color\_ , j.Backbone\_ , j.HopCount\_ , N_j^k$ );

25. Set:  $j.isBroadcast\_ = 1;$

26. }

27. if ( $i.Backbone\_ == j.Identity\_$ ) {

28. Set:  $j.Color\_ = BLACK;$

29. if ( $i \notin N_j^K$ )

30.  $N_j^K.insert (i.Identity\_ , d_{(i,j)});$

31. Set:  $T_{x_j^k} = \max(N_j^K(1).Distance\_ , N_j^K(N_j^K.size()).Distance\_);$

32. }

33. }

### Algorithm 3: Procedure for finding the backbone node

```

findBackbone() {
  /*Initialize*/
  1. entry.Distance_ = 250; entry.HopCount_ = 255;
  2. for ( $k \in j.\text{msgCache}_j$ ) {
    /* Find the least-distant neighbor with color = BLUE */
  3. if ( $k.\text{Distance}_k < \text{entry}.\text{Distance}_j$ ) and ( $k.\text{Color}_k == \text{BLUE}$ )
  4. Set: entry = k;
  5. }
  6. if (entry == NULL) {
  7. for ( $l \in j.\text{msgCache}_j$ ) {
    /* Find the least distant neighbor with least hops towards the sink */
  8. if ( $l.\text{Distance}_l < \text{entry}.\text{Distance}_j$ ) and ( $l.\text{HopCount}_l \leq \text{entry}.\text{HopCount}_j$ )
  9. Set: entry = l;
  10. }
  11. }
  12. Set: j.Backbone_ = entry.Identity_;
  13. if (j.Backbone_  $\neq$  Sink) Set: j.HopCount_ = entry.HopCount_ + 1;
  14. if (j.Color_ == WHITE) Set: j.Color_ = BLUE;
  15. if (entry.Color_ == BLACK) Set: j.Color_ = RED;
  16. if (entry.Identity_  $\notin N_j^k$ )
  17.  $N_j^k.\text{insert}(\text{entry}.\text{Identity}_j, d_{(\text{entry}.\text{identity}_j)})$ ;
  18. Set:  $\text{Tx}_j^k = \max(N_j^k(1).\text{Distance}_j, N_j^k(N_j^k.\text{size}()).\text{Distance}_j)$ ;
  19. }

```

### Algorithm 4: Topology maintenance phaseFor node *i* sends Topology Repair Request (TRREQ) message:

```

1. if ( $i.\text{msgCache}_i.\text{size}() < N_i^k.\text{size}()$ )
2. Call: sendTRREQ( $N_i^k$ ); For the other node j sends Topology Repair Reply (TRREP) message:
  /*Upon receiving the TRREQ message*/
3. receiveTRREQ( $N_i^k$ ) {
4. if ( $j \notin N_i^k$  and  $i \in N_j^k$ )
5. Call: sendTRREP(j.Identity_);
6. } For node i receives TRREP message from node j:
7. receiveTRREP (j.Identity_) {
8. if ( $j \notin N_i^k$ )
9.  $N_i^k.\text{insert}(j.\text{Identity}_j, d_{(i, j)})$ ;
10. Set:  $\text{Tx}_i^k = \max(N_i^k(1).\text{Distance}_i, N_i^k(N_i^k.\text{size}()).\text{Distance}_i)$ ;
11. }

```

### Acknowledgements

This research was supported by the MKE(The Ministry of Knowledge Economy), Korea, under the ITRC(Information Technology Research Center) support program supervised by the NIPA(National IT Industry Promotion Agency" (NIPA-2012-(C1090-1221-0011)).

### Competing interests

The authors declare that they have no competing interests.

### Author details

<sup>1</sup>RFID/USN Research Division, Electronics and Telecommunications Research Institute (ETRI), Daejeon, Republic of Korea <sup>2</sup>School of Information and Computer Engineering, Ajou University, Suwon, Republic of Korea

Received: 30 October 2011 Accepted: 9 March 2012

Published: 9 March 2012

### References

- P Santi, Topology control in wireless ad hoc and sensor networks. *ACM Comput Surv.* **37**(2), 164–194 (2005). doi:10.1145/1089733.1089736
- P Santi, *Topology Control in Wireless Ad Hoc and Sensor Networks*, (John Wiley and Sons, Chichester, UK, 2005)
- DM Blough, M Leoncini, G Resta, P Santi, The K-neigh protocol for symmetric topology control in ad hoc networks, in *Proc of ACM MobiHoc 2003*, Annapolis, Maryland, USA, pp. 141–152 (1-3 June 2003)
- DM Blough, M Leoncini, G Resta, P Santi, The k-neighbors approach to interference bounded and symmetric topology control in ad hoc networks. *IEEE Trans Mobile Comput.* **5**(9), 1267–1282 (2006)
- P Wan, C Yi, Asymptotic critical transmission radius and critical neighbor number for k-connectivity in wireless ad hoc networks, in *Proc ACM MobiHoc 2004*, Roppongi Hills, Tokyo, Japan, pp. 1–8 (24-26 May 2004)
- F Xue, PR Kumar, The number of neighbors needed for connectivity of wireless networks. *Wirel Netw.* **10**(2), 169–181 (2004)
- R Wattenhofer, A Zollinger, XTC: a practical topology control algorithm for ad-hoc networks, in *Proc of WMAN 2004*, vol. 13. Santa Fe, New Mexico, p. 216a (26-30 April 2004)
- R Ramanathan, R Rosales-Hain, Topology control of multi-hop wireless networks using transmit power adjustment, in *Proc of IEEE INFOCOM 2000*, vol. 2. Tel Aviv, Israel, pp. 404–413 (26-30 March 2000)
- J Liu, B Li, MobileGrid: capacity-aware topology control in mobile ad hoc networks, *Proc of IEEE ICCCN 2002*, Miami, Florida, USA, pp. 570–574 (14-16 October 2002)
- L Hu, Topology control for multi-hop packet radio networks. *IEEE Trans Commun.* **41**(10), 1474–1481 (1993). doi:10.1109/26.237882
- M Gerharz, C de Waal, P Martini, P James, A cooperative nearest neighbors topology control algorithm for wireless ad hoc networks. *Springer Telecommun Syst J.* **28**(3-4), 317–331 (2005)
- R Banner, A Orda, Multi-objective topology control in wireless networks, in *Proc of IEEE INFOCOM 2008*, Phoenix, Arizona, USA, pp. 448–456 (13-18 April 2008)
- N Li, JC Hou, L Sha, Design and analysis of an MST-based topology control algorithm, in *Proc of IEEE INFOCOM 2003*, vol. 3. Urbana-Champaign, Illinois, USA, pp. 1702–1712 (30 March-3 April 2003)
- CC2420: Datasheet for Chipcon (TI) CC2420 2.4 GHz IEEE 802.15.4/ZigBee RF Transceiver <http://www.ti.com/lit/ds/symlink/cc2420.pdf>
- CC1000: Datasheet for Chipcon (TI) CC1000 315, 433, 868, 915 MHz ultra-low-power RF Transceiver <http://www.ti.com/lit/ds/symlink/cc1000.pdf>
- S Park, RR Palasdeokar, Reliable one-hop broadcasting (ROB) in mobile ad hoc networks, in *Proc of ACM PE-WASUN 2005*, Montreal, Quebec, Canada, pp. 234–237 (13 October 2005)
- P Bahl, VN Padmanabhan, RADAR: an in-building RF-based user location and tracking system, in *Proc of IEEE INFOCOM 2000*, vol. 2. Tel Aviv, Israel, pp. 775–784 (26-30 March 2000)
- L Girod, D Estrin, Robust range estimation using acoustic and multimodal sensing, in *Proc of IEEE/RSJ ICIRS 2001*, vol. 3. Maui, Hawaii, USA, pp. 1312–1320 (29 October-3 November 2001)
- NS-2, The Network Simulator <http://www.isi.edu/nsnam/ns>
- K Pahlavan, P Krishnamurthy, in *Principles of Wireless Networks- A Unified Approach*, ed. by K Pahlavan, P Krishnamurthy, Principles of Wireless



Networks - A Unified Approach, Prentice Hall, ISBN 0-13-093003-2, Hardcover, 584 pages, January 2002 (Prentice Hall, 2002).

21. S Slijepcevic, S Megerian, M Potkonjak, Location errors in wireless embedded sensor networks: sources, models, and effects on applications. *ACM Sigmobile Mobile Comput Commun Rev.* **6**(3), 67–78 (2002). doi:10.1145/581291.581301

doi:10.1186/1687-1499-2012-97

**Cite this article as:** Mir and Ko: Adaptive neighbor-based topology control protocol for wireless multi-hop networks. *EURASIP Journal on Wireless Communications and Networking* 2012 **2012**:97.

**Submit your manuscript to a SpringerOpen<sup>®</sup> journal and benefit from:**

- Convenient online submission
- Rigorous peer review
- Immediate publication on acceptance
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

---

Submit your next manuscript at ► [springeropen.com](http://springeropen.com)

---