# LTRT: An Efficient and Reliable Topology Control Algorithm for Ad-Hoc Networks

Kenji Miyao, *Student Member, IEEE,* Hidehisa Nakayama, *Member, IEEE,*
Nirwan Ansari, *Fellow, IEEE,* and Nei Kato, *Senior Member, IEEE*

***Abstract*—Broadcasting, in the context of ad-hoc networks, is a costly operation, and thus topology control has been proposed to achieve efficient broadcasting with low interference and low energy consumption. By topology control, each node optimizes its transmission power by maintaining network connectivity in a localized manner. Local Minimum Spanning Tree (LMST) is the state-of-the-art topology control algorithm, which has been proven to provide satisfactory performance. However, LMST almost always results in a 1-connected network, without redundancy to tolerate external factors. In this paper, we propose Local Tree-based Reliable Topology (LTRT), which is mathematically proven to guarantee $k$-edge connectivity while preserving the features of LMST. LTRT can be easily constructed with a low computational complexity of $O(k(m + n \log n))$, where $k$ is the connectivity of the resulting topology, $n$ is the number of neighboring nodes, and $m$ is the number of edges. Simulation results have demonstrated the efficiency of LTRT and its superiority over other localized algorithms.**

***Index Terms*—Ad-hoc networks, topology control, reliability, minimum spanning tree, $k$-edge connectivity.**

## I. Introduction

RECENT advances in wireless and mobile technologies have fostered the development of ad-hoc networks. Since nodes operate with limited battery power, reducing energy consumption to prolong lifetime of the network has always been an important issue. In ad-hoc networks, transmissions are classified into three types, namely, unicast, multicast, and broadcast transmissions. In this paper, we consider the latter. Broadcast transmissions in ad-hoc networks are used for, for example, sending control packets, distributing cryptographic keys, and so forth. Broadcast by flooding usually consumes much energy and also leads to high MAC-level interference. Therefore, it is not readily applicable in ad-hoc networks owing to resource constraints of mobile nodes.

Topology control is one means to broadcast efficiently [1]. By topology control, each node transmits packets by using relatively lower power to prevent interferences, and to reduce energy consumption. Algorithms are generally localized, i.e., each node uses only the information that is one-hop away. The problem of minimizing the total energy consumption is NP-hard in the three dimensional space [2]. Even in the two dimensional case, it is still NP-hard [3]. Many globalized algorithms such as [4], [5] have also been proposed, but they are not scalable. If the ad-hoc network consists of thousands of nodes, it is difficult to calculate the optimal transmission ranges and to pass on the information to the concerned nodes. Also, collecting the information of all the nodes will incur high overheads. Therefore, topology control algorithms have to be localized. However, determining the transmission power from local information is rather difficult, since adjustment of the power without global knowledge may not preserve the network connectivity.

Localized topology control algorithms have mathematically proved to be able to maintain the global connectivity. Each node can decide the transmission range from local information in order to maintain global network connectivity. Among the localized topology control algorithms, the most cost efficient algorithm is Local Minimum Spanning Tree (LMST) [6], which is localized MST-based algorithm. Although LMST is cost-efficient, LMST almost always results in only one fixed path between every pair of nodes. Control of transmission power reduces energy consumption of each node easily, but lowering the transmission power always implies the loss of fault-tolerance. So, if there is a link failure, the network may be split and some nodes will not be able to receive packets. The problem is that LMST has almost always only one path between every pair of two nodes in the network. If there are two or more paths between them, network reliability will indeed be enhanced.

In this paper, we propose the Local Tree-based Reliable Topology (LTRT) algorithm which is motivated by LMST and the Tree-based Reliable Topology (TRT) [7]. LTRT is a localized version of TRT which guarantees 2-edge connectivity. We combine the idea of TRT in LMST to guarantee $k$-edge connectivity of the resulting topology. Indeed, improving reliability leads to an increase of energy consumption. However, LTRT can maintain the extent of such an increase within a tolerable limit. LTRT can achieve nearly optimal performance at a much lower computational cost.

The remainder of the paper is organized as follows. Section II reviews some related works. We define the network model in Section III. In Section IV, we introduce the LTRT algorithm

Manuscript received January 19, 2009; revised July 14, 2009; accepted August 25, 2009. The associate editor coordinating the review of this paper and approving it for publication was X. Zhang.

K. Miyao and N. Kato are with the Graduate School of Information Sciences, Tohoku University, Sendai, 980-8579 Japan (e-mail: {miyao, kato}@it.ecei.tohoku.ac.jp).

H. Nakayama is with the Department of Electronics and Intelligenft Systems, Tohoku Institute of Technology, Sendai, 982-8577 Japan (e-mail: hidehisa@m.ieice.org).

N. Ansari is with the Advanced Networking Laboratory, Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07102 USA (e-mail: nirwan.ansari@njit.edu).

The preliminary version of this work was presented at IEEE Workshop on Security, Privacy and Authentication in Wireless Networks (SPAWN) 2008.

Digital Object Identifier 10.1109/TWC.2009.12.090073

and derive the properties of LTRT. Performance comparisons of LTRT with other existing algorithms are illustrated in Section V. Finally, Section VI concludes the paper.

## II. RELATED WORKS

Topology control has been widely studied. Cone-Based distributed Topology Control (CBTC($\alpha$)) [8] is among the first algorithms that adjusts the transmission power to save energy consumption. In CBTC($\alpha$), a node $u$, which transmits with the minimum power $p_u$, is required to ensure that in every cone of degree $\alpha$ around $u$, there is some node that $u$ can reach with the power $p_u$. The authors [8] analytically showed that if $\alpha < 5/6\pi$, the network connectivity is preserved.

Relative Neighborhood Graph (RNG) (first appeared in [9]) is also used to reduce the number of links between a node and its neighbors [10]. An edge belongs to the RNG only if it is not the longest leg of any triangle it may form in the original graph.

Li *et al.* [6] proposed a Minimum Spanning Tree based algorithm for topology control. LMST is a "localized" algorithm to construct MST based topology in ad-hoc networks by using only information of nodes which are one-hop away. Every node knows its position by GPS and has its ID for identification. The idea of LMST is simple. Each node calculates MST independently from the information of one-hop nodes and only keeps one-hop on-tree nodes as neighbors. The procedure of constructing LMST is composed of two phases. First, each node broadcasts a "Hello" message using the maximal transmission power, which contains its ID and position, and obtains the information of its one-hop nodes. Each node then has its local graph in this phase. In the next stage, each node applies Prim's algorithm independently to obtain its local MST and keeps its on-tree hop-one nodes as its neighbors. If every link has a unique weight (i.e., different links have different weights), the locally calculated MST is also unique and the connectivity can be guaranteed. The topology of LMST may not be a spanning tree but may have some redundant edges. LMST has several noteworthy features. The node degree of any node is bounded by 6; this can help reduce MAC-level contention and interference. The resulting topology can be converted into the one with only bi-directional links by removing all uni-directional links. Note that the topology of the resulting LMST might be split by a single link failure. This might limit its applicability since the topology in an ad-hoc network should have some redundancy because of its unsure links.

In recent years, some new approaches have been proposed. In [11], the authors assumed that nodes may act in their self-interest. They modeled interactions among nodes as a *game*, and analyzed the problem as a non-cooperative game. In [12], the authors proposed an algorithm to optimize the traditional topology control scheme. In this algorithm, each node iteratively increases its transmission power. This algorithm starts from a symmetric, connected topology, which assumes to be the output of a topology control. This can be applicable to many topology control schemes.

Among many localized topology control algorithms, LMST achieves the best performance with regard to energy efficiency. Based on the derived topology by a topology control algorithm, some optimization algorithms can be applied. RNG based Broadcast Oriented Protocol (RBOP) and LMST based Broadcast Oriented Protocol (LBOP) [13] can save more energy. In RBOP and LBOT, the broadcast is initiated at the source, and is propagated, following the rules of neighbor elimination, on the topology derived from RNG and LMST. TR-LBOP and TRDS are proposed in [14], in which the authors claimed that in a dense network, other topology control is no longer effective, and their protocols achieve more efficient transmission than normal topology control. Broadcast on LMST (BLMST) [15] is a flooding algorithm applied to the network topology derived by LMST with the optimization that if a node has received a broadcast message from all its neighbors, it will not relay the message.

The optimization of broadcasting mentioned above can be applicable to most of the topology control algorithms and save energy consumption significantly. Optimization of broadcasting, though interesting, is beyond the scope of this paper. Here, we focus on topology control itself rather than on broadcasting processes.

Although LMST achieves good efficiency, it considers only 1-connectivity. Therefore, fault-tolerant topology control algorithms have been proposed to mitigate this shortcoming. To achieve reliability, the $k$-connectivity approach is adopted. However, finding the minimum-cost $k$-connected subgraph is proved to be NP-hard, for which some approximation algorithms have been proposed. Bahramgiri *et al.* [16] proved that CBTC($\alpha$) preserves $k$-connectivity if $\alpha < \frac{2\pi}{3k}$. However, when $\alpha < \frac{2\pi}{3k}$, the topology has many redundant edges and consequently is not energy efficient. Fault-tolerant Local Spanning Subgraph (FLSS$_k$) [17] also guarantees $k$-connectivity if the network has $k$-connectivity. As compared to CBTC($\alpha$), FLSS$_k$ shows much better performance. The computational complexity is, however, $O(m(n+m))$ where $n$ is the number of nodes and $m$ is the number of edges. The authors derived that if $k \leq 3$, the complexity is reduced to $O(m)$ because connectivity testing can be operated in $O(1)$ when $(k \leq 3)$ [18]. However, this reduced complexity ignores the preprocessing cost of the connectivity testing algorithm. Therefore, the best achievable complexity is $O(m)$ for any value of $k$ [19]. The actual computational complexity is $O(m(m+n))$ regardless of the value of $k$. Since $m \simeq n^2$ in a dense network, the complexity is approximately $O(n^4)$. This high computational cost is not suitable for highly dense networks or mobile networks.

The above proposed algorithms cannot achieve realistic topology control. On the other hand, the proposed algorithm, LTRT, preserves $k$-edge connectivity with much lower computational cost, $O(k(m + n \log n))$; LTRT is consequently more suitable for realistic networks than other fault tolerant topology control algorithms.

## III. NETWORK MODEL

We consider multi-hop wireless networks, and assume that each node is able to gather its own location information via GPS or several localization techniques for wireless networks such as [20] [21].

We represent a network as an undirected graph $G = (V, E)$ where $V$ is the set of nodes and $E \subseteq V^2$ is the set of edges. Each node $v \in V$ has a unique ID which identifies the node, denoted by $id(v)$. We assume that each node can control the power of its transmissions to save energy consumption. Furthermore, we employ the unit disk graph (UDG) model, which can be used for approximating the original network. In this model, every node is embedded in the plane. Each node $v$ has the same maximum transmission radius $r_{max}$. Each node $u$ can transmit packets within its transmission radius, $r(u)$, $0 \le r(u) \le r_{max}$. Let $d(v_1, v_2)$ be the Euclidean distance between two vertices $v_1, v_2 \in V$. An edge exists between two nodes $v_1, v_2 \in V$ if and only if $d(v_1, v_2) \le r_{max}$.

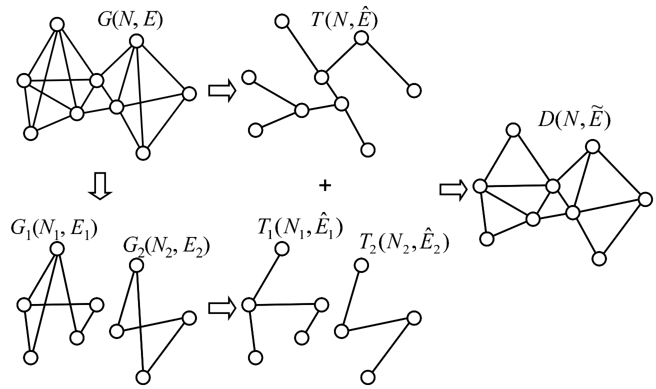

Fig. 1. Construction of TRT.

### A. Weight Function

Two edges with different nodes must have different weights because different weights guarantee the network connectivity. Given two edges $(u_1, v_1)$, $(u_2, v_2) \in E$, the weight function $w$ satisfies:

$$\begin{aligned}
& w(u_1, v_1) > w(u_2, v_2) \\
\Leftrightarrow\ & d(u_1, v_1) > d(u_2, v_2) \\
or\ & (d(u_1, v_1) = d(u_2, v_2) \\
& \&\& \max\{id(u_1), id(v_1)\} > \max\{id(u_2), id(v_2)\}) \\
or\ & (d(u_1, v_1) = d(u_2, v_2) \\
& \&\& \max\{id(u_1), id(v_1)\} = \max\{id(u_2), id(v_2)\} \\
& \&\& \min\{id(u_1), id(v_1)\} > \min\{id(u_2), id(v_2)\}).
\end{aligned}$$

### B. Neighbor Set

Node $v$ is a neighbor of node $u$ if there exists $(u, v) \in E$. The neighbor set of node $u$ is denoted as $N(u) = \{v \in V | (u, v) \in E\}$. Since network $G(V, E)$ is symmetric, (i.e., every edge is bidirectional,) $v \in N(u) \Leftrightarrow u \in N(v)$ is always implied. Node $v$ becomes a neighbor of node $u$ by means of the algorithm $ALG$, if there exists an edge $(u, v)$ in the resulting topology generated by ALG. The neighbor set of node $u$ generated by $ALG$ is denoted as $N_{ALG}(u)$. Note that $|N_{ALG}(u)| \le |N(u)|$.

### C. Bi-Directionality

A topology generated by $ALG$ is bidirectional if and only if for any two nodes $u, v \in V$, $u \in N_{ALG}(v)$ implies $v \in N_{ALG}(u)$

### D. $k$-vertex connectivity and $k$-edge connectivity

A graph $G(V, E)$ is $k$-vertex connected if the removal of any $(k-1)$ nodes does not partition the network, and is $k$-edge connected if the removal of any $(k-1)$ edges does not partition the network. In other words, there are $k$-vertex-disjoint paths for any $v_1, v_2 \in V$, or there are $k$-edge-disjoint paths for any $v_1, v_2 \in V$. In the case where $k = 1$, vertex connectivity and edge connectivity represent the same property, i.e., 1-vertex connectivity is the same as 1-edge connectivity. Generally, $k$-vertex connectivity is *stronger* than $k$-edge connectivity, i.e., $k$-vertex connectivity implies $k$-edge connectivity. Therefore, most studies of network connectivity in topology control focus on vertex connectivity. However, if a node is dropped, the topology has to be recalculated, thus incurring a large computational complexity. Consequently, it is sufficient to employ $k$-edge connectivity, a weaker one, which leads to a "low cost" topology.

### E. Transmission range

Each node can adjust the transmission range to minimize the energy consumption. Transmission range is controlled to the level that can reach the furthest neighbor. Therefore, when an algorithm $ALG$ is applied, the transmission range of each node is:

$$r(u) = \max\{d(u, v) | v \in N_{ALG}(u)\}$$

## IV. LTRT: Local Tree-Based Reliable Topology

In this section, we propose LTRT, which is a localized version of TRT. LTRT always generates a $k$-edge connected network if the original network is $i$-edge connected, where $i \ge k$. Also, the computational complexity is low.

Before explaining the Localized TRT algorithm, we will briefly review TRT. Next, we explain the algorithm of LTRT, and then prove that LTRT achieves $k$-edge connectivity if the original network is $i$-edge connected, where $i \ge k$. We also analyze the computational complexity; since devices in an ad-hoc network are always constrained with low computational power and energy, computational complexity is a crucial concern.

### A. TRT: Tree-based Reliable Topology

Ansari *et al.* [7] introduced the concept of Reliable Topology (RT) which guarantees 2-edge connectivity, and proposed an algorithm to construct RT by combining spanning trees, referred to as Tree-based Reliable Topology (TRT).

TRT is constructed as follows. Given $G(V, N)$, a connected network topology. We first calculate one of its spanning trees, $T(N, \hat{E})$. Then, we remove all the links in $\hat{E}$ from $G(V, N)$, and denote the resulting network as $G(N, E - \hat{E})$ which consists of $n$ ($n \ge 1$) connected sub-networks which are $G_1(N_1, E_1)$,$G_2(N_2, E_2)$,. . .,$G_n(N_n, E_n)$. This follows by the computations of $T_1(N_1, \hat{E}_1)$,$T_2(N_2, \hat{E}_2)$,. . .,$T_n(N_n, \hat{E}_n)$, which are the spanning trees of $G_1(N_1, E_1)$,

$i \leftarrow 1$
Send $id(u)$ and $position(u)$ with maximal power
Receive the message and construct local graph $G_u(N(u), E(u))$
**while** $true$ **do**
  Calculate MST of $G_u(N(u), E(u))$
  $ID(u) \leftarrow \{id(v) | v \in N_{MST}(u)\}$
  Broadcast $ID(u)$
  $E_i(u) \leftarrow \{(v, w) | w \in N(v) \cap v \in N(w)\}$
  $N_i(u) \leftarrow \{v | (u, v) \in E_i(u)\}$
  **if** $i = k$ **then**
    break
  **else**
    $E(u) \leftarrow E(u) \setminus E_i(u)$
    $i \leftarrow i + 1$
  **end if**
**end while**
$N_{LTRT}(u) \leftarrow \bigcup N_i(u)$
$r(u) \leftarrow \max\{d(u, v) | v \in N_{LTRT}(u)\}$

Fig. 2. Operation of each node $u$ in the LTRT algorithm.

$G_2(N_2, E_2)$,…,$G_n(N_n, E_n)$, respectively. The topology, $D(N, \tilde{E})$ constructed by combining $T(N, \hat{E})$, $T_1(N_1, \hat{E}_1)$, $T_2(N_2, \hat{E}_2)$, …, $T_n(N_n, \hat{E}_n)$ is referred to as a Tree-based Reliable Topology (TRT).

The construction procedure of TRT is illustrated by an example as shown in Fig. 1. Given a network $G(N, E)$ as shown in Fig. 1, we can construct one of its TRTs, $D(N, \tilde{E})$, by combining $T(N, \hat{E}), T_1(N_1, \hat{E}_1)$, and $T_2(N_2, \hat{E}_2)$, where $T(N, \hat{E})$ is one of the spanning trees of $G(N, E)$, and $T_1(N_1, \hat{E}_1)$ and $T_2(N_2, \hat{E}_2)$ are the spanning trees of $G_1(N_1, E_1)$ and $G_2(N_2, E_2)$, respectively, which are the remaining networks after having removed the links in $T(N, \hat{E})$ from $G(N, E)$.

It can be observed that if $n = 1$, $G(N, E \setminus \hat{E})$ is still a connected network; TRT is actually constructed by combining the two spanning trees of $G(N, E)$. Ansari *et al.* [7] suggested to deploy minimum spanning tree (MST) to construct TRT, i.e., in the process of constructing the TRT, all the spanning trees are the minimum spanning trees of the corresponding networks.

Actually, TRT preserves only 2-edge connectivity, but it can be extended to $k$-edge connectivity by repeating the process of MST calculation and link deletion. In the next section, we incorporate the TRT concept with LMST, to construct $k$-edge connected topology.

### B. Construction of LTRT

LTRT can be easily constructed by applying the topology construction phase of LMST $k$ times. The procedure of constructing LTRT is composed of four phases, namely, information exchange, topology construction, link deletion, and transmission radius control. The topology construction and link deletion phases are repeated until the topology achieves $k$-edge connectivity.

1) Information Exchange: Each node $u$ broadcasts a "Hello" message, which contains its ID $id(u)$ and position, and obtains the information of its neighbor set $N(u)$. Its edge set $E(u)$ is calculated from the positions of $N(u)$. Each node obtains its local graph $G_u(N(u), E(u))$ in this phase.

2) Topology Construction: Each node $u$ applies Prim's algorithm independently to obtain its local MST, $T_u{}'(N_u, E_u{}')$ by using the weight function $w$, with neighbor set $N_{MST}(u) = \{v | (u, v) \in E_u{}'\}$, and each node broadcasts its ID set $ID(u) = \{id(v) | v \in N_{MST}(u)\}$. Each node $u$ can obtain all neighbors of $v \in N_u$ from the broadcast, and then can fix the $i$th edge set $E_i(u)$. Topology $D_u(N(u), E_i(u))$ is an undirected graph constructed by eliminating unidirectional edges. This elimination of unidirectional edges does not destroy the connectivity of the network. If $i = k$, i.e., the topology construction phase is repeated $k$ times, we skip to Phase 4. Otherwise, the next step is Phase 3.

3) Link deletion: Each node $u$ deletes the links in $E_i(u)$ from its local graph $G_u(N(u), E(u))$, resulting in the topology, $G_u(N(u), E(u) \setminus E_i(u))$, and then we go back to Phase 2.

4) Transmission Radius Control: Each node $u$ can decide its neighbor set $N_{LTRT}(u) = \bigcup_{i=1}^{k} \{v | (u, v) \in E_i(u)\}$. Finally, the transmission radius $r(u) = \max(\{d(u, v) | v \in N_{LTRT}(u)\})$ is set, i.e., each node sets the transmission power to the level so that it may reach the furthest neighbor.

This algorithm is summarized in Fig. 2. The example of construction with $k = 2$ is illustrated in Fig. 3.

There are some noteworthy features in LTRT. LTRT is an undirected graph. The bulk of the computation is due to the cost of Prim's algorithm, and so the computational cost is rather small. Only $(k + 1)$ times of broadcasts per node are required, i.e., transmission with low overhead. If the weight of any link has a unique value, the choice on the minimum weight edge, $e$, is unique, and thus the topology calculated by Prim's algorithm is also unique. Therefore, we can always obtain a unique topology from LTRT.

The network can flexibly increase the connectivity if needed. To increase connectivity from $k$ to $(k+1)$, each node only needs to calculate MST once more, rather than another $(k + 1)$ times.

When a node joins or leaves the network, the topology will be recalculated. However, it is not necessary for every node of the network to recalculate its topology because the algorithm is localized. In such a situation, the nodes that are one-hop away from the joining node start the topology construction phase with $i = 1$ , and nodes that are two-hop away proceed to phase 3) with $i = 2$. This scheme is repeated by all $k$-hop nodes. Hence, only nodes that are $k$-hop or less away from the joining node need to recompute their topologies when the network is changed.

### C. $k$-edge connectivity

The topologies constructed in phase 2) is the same as that of LMST. The connectivity of LMST is proved in [6]. By using the connectivity of LMST, we show that LTRT achieves $k$-edge connectivity.

*Theorem 1:* Given a $k$-edge connected network $G(N, E)$, LMSTs $T_{i1}(N_{i1}, \hat{E}_{i1})$, $T_{i2}(N_{i2}, \hat{E}_{i2})$, …, $T_{in_i}(N_{in_i}, \hat{E}_{in_i})$, $i = 1, 2, \ldots, k$, and the LTRT, $D(N, \tilde{E})$, constructed by combining all LMSTs, LTRT $D(N, \tilde{E})$ is also $k$-edge connected.

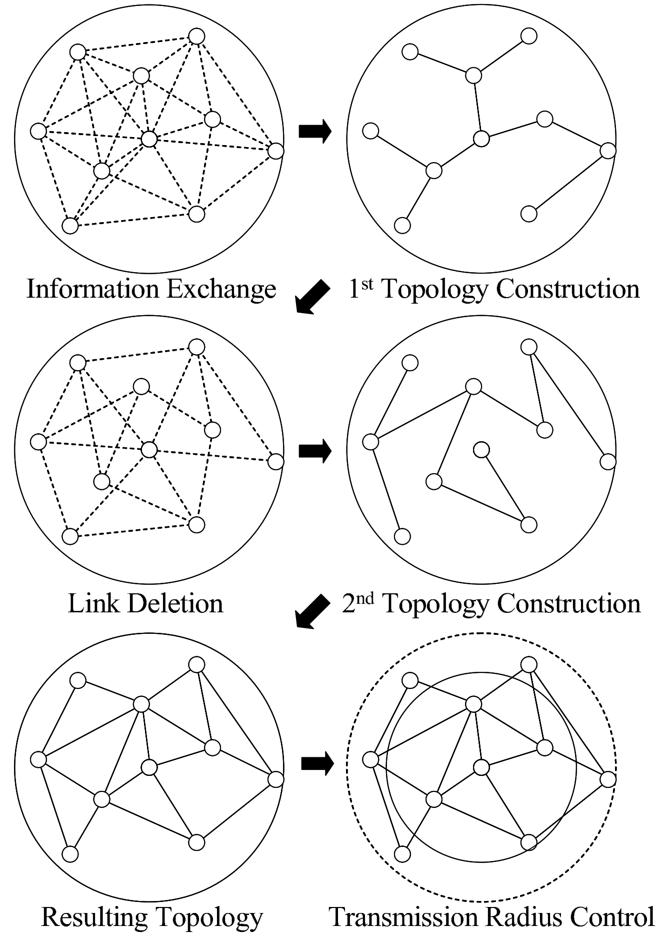


Fig. 3. Example of LTRT operation (when $k = 2$).

*Proof:* Given $D_l(N, \tilde{E}_l)$, constructed by combining $T_{i1}(N_{i1}, \hat{E}_{i1})$, $T_{i2}(N_{i2}, \hat{E}_{i2})$, …, $T_{in_i}(N_{in_i}, \hat{E}_{in_i})$, $i = 1, 2, \ldots, l$. We prove by induction.

$D_1(N, \tilde{E}_1)$ is the same as that of LMST. Therefore, $D_1(N, \tilde{E}_1)$ is 1-edge connected.

Assume that $D_{k-1}$ is $(k-1)$-edge connected for $k \geq 2$. If any set of $k$ edges $\{e_1, e_2, \ldots, e_{k-1}\}$ is not an edge-cut of $D_k(N, \tilde{E}_k)$, $D_k(N, \tilde{E}_k)$ is $k$-edge connected, i.e., LTRT $D(N, \tilde{E})$ is $k$-edge connected. We prove this by showing that any edge $\{e_1, e_2, \ldots, e_{k-1}\}$ is not an edge-cut of $D_k(N, \tilde{E})$. Assume edge set $\{e_1, e_2, \ldots, e_{k-1}\}$ is an edge-cut of $D(N, \tilde{E})$ that splits $D(N, \tilde{E})$ into $D^1(N^1, \tilde{E}^1)$ and $D^2(N^2, \tilde{E}^2)$ by removing $\{e_1, e_2, \ldots, e_{k-1}\}$, but does not split $G(N, E)$. For any $e \in \{e_1, e_2, \ldots, e_{k-1}\}$, $e \in \bigcup_{i=1}^{k-1} \bigcup_{j=1}^{n_i} \hat{E}_{ij}$ is obvious because $D_{k-1}(N, \hat{E})$ is $(k-1)$-edge connected and $D_{k-1}(N, \hat{E}^{k-1}) \subset D_k(N, \tilde{E}^k)$. However, there are some edges $\{e_1, e_2, \ldots, e_m\} \in (E \backslash \bigcup_{i=1}^{k-1} \bigcup_{j=1}^{n_i} \hat{E}_{ij})$ that connect between the set of $N_1$ and $N_2$ like those shown in Fig. 4, but are not the links of $\bigcup_{i=1}^{k-1} \bigcup_{j=1}^{n_i} \hat{E}_{ij}$, since $G(N, E)$ is $k$-edge connected. Since the topologies constructed in the $k$th round of topology construction phase are connected, one of $e_1, e_2, \ldots, e_n$ must be in one of $\hat{E}_{k1}, \hat{E}_{k2}, \ldots, \hat{E}_{kn_k}$. This, however, contradicts our assumption that $\{e_1, e_2, \ldots, e_{k-1}\}$ is the edge-cut of $D_k(N, \tilde{E}_k)$. Hence, none of the edges $\{e_1, e_2, \ldots, e_{k-1}\}$ is an edge-cut of $D_k(N, \tilde{E}_k)$, that is, $D(N, \tilde{E})$ is $k$-edge connected. ■

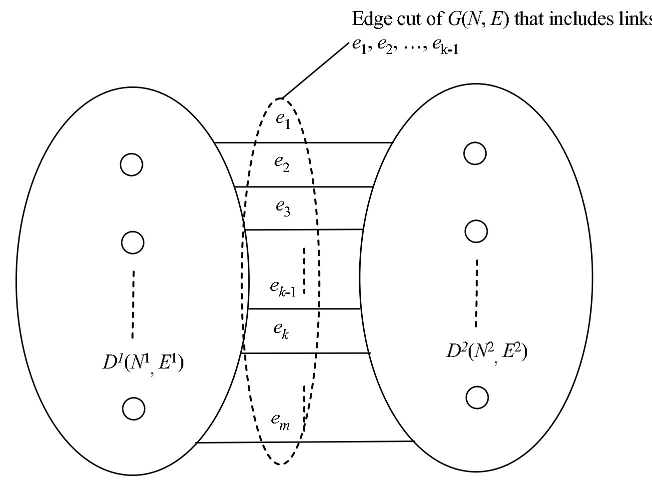


Fig. 4. An edge cut of $D(N, E)$.

### D. Complexity analysis

We show the computational complexity of LTRT construction to be $O(k(m + n \log n))$, where $k$ is the connectivity of the resulting topology, $n$ is the number of nodes which are one-hop away, and $m$ is the number of links in the local network $G_u(N, E)$. It is the similar to that of LMST.

In the information exchange phase, each node broadcasts and obtains the information. In this phase, adding the neighbors in the local graph costs $O(n)$. Each node calculates the length from the node positions to obtain link lengths in its local graph, thus costing $O(n^2)$. If each node $u$ calculates only the link lengths between the neighbor node $v$ and $u$, and broadcasts its length, the cost can be lowered to $O(n)$. In the topology construction phase, each node applies Prim's algorithm with complexity of $O(m \log n)$ [22]. If we employ Fibonacci heap, the complexity is $O(m + n \log n)$ [23]. In the link deletion phase, the deletion of a link from the network is $O(\log n)$ because there are $n$ links at most for each node. Since the node degree of any node is bounded by 6, the number of manipulations is $O(n)$. So, the deletion cost is $O(n \log n)$.

The topology construction phase and the link deletion phase are repeated $k$ times. Therefore, the computational complexity of the proposed algorithm is $O(k(m + n \log n))$. Since the cost of computation is almost the same as that of Prim's algorithm, the actual computational complexity is rather low, and the algorithm can be readily applied.

### E. Node movement

In our proposed algorithm, each node only uses the minimum power to reduce energy consumption effectively. However, in the mobile case, a small movement may result in the loss of connectivity. To avoid this problem, each node can adjust the update frequency of the topology and transmit with a larger transmission radius, according to the node's speed. In such a case, nodes will determine the update period and additional margin of transmission power (i.e., the extra power), by sharing the speed information, and by using a probabilistic scheme like that in [6]. If the nodes are moving too fast, this will incur too frequent updates, and the margin of transmission power will become too large. This problem

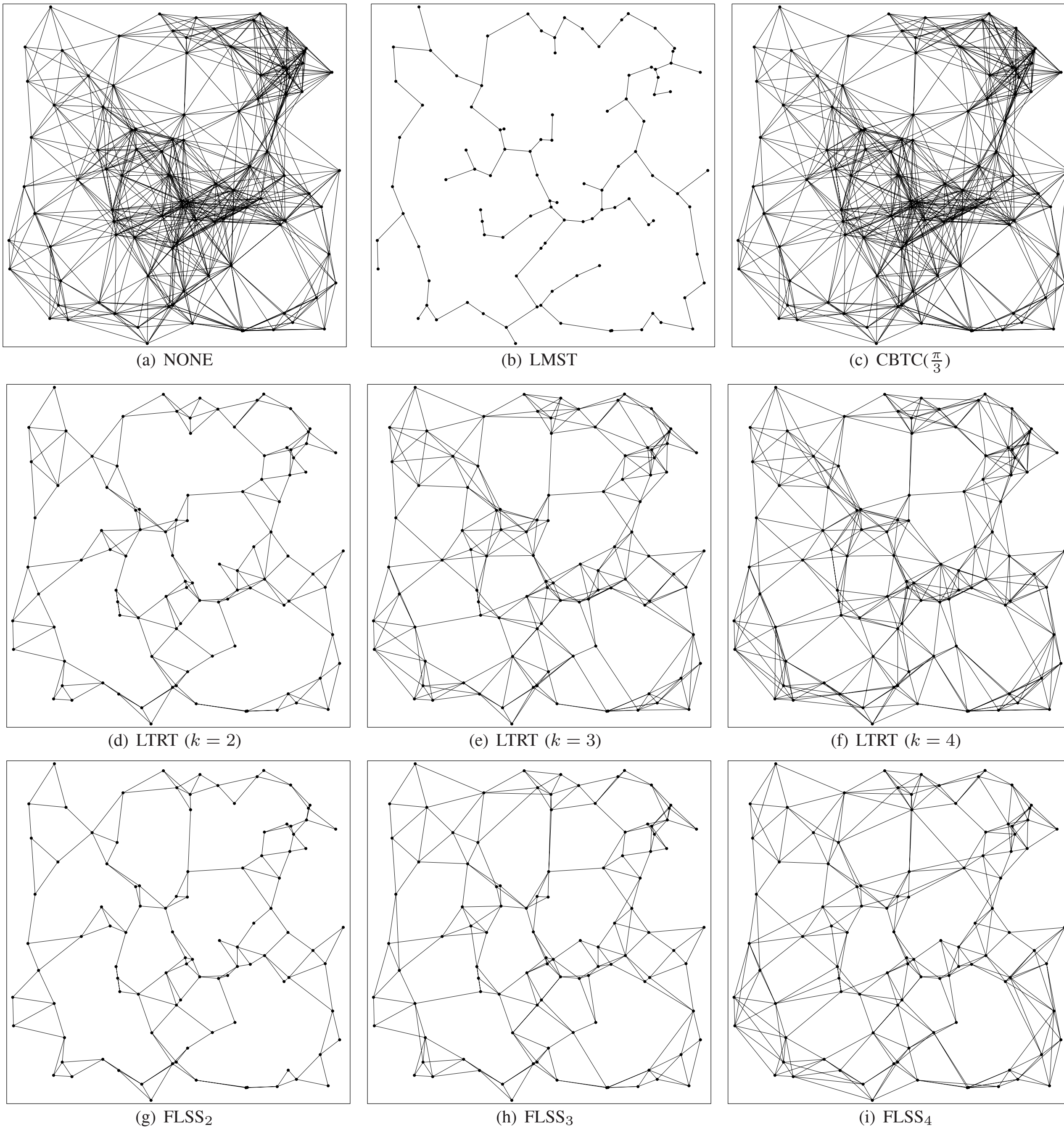


Fig. 5. The topologies derived by different algorithms.

may be mitigated by measuring the record of movements and using some statistical prediction algorithm.

LTRT may employ these probabilistic scheme and prediction algorithm to mitigate the problem of loss of connectivity in the mobile case. This issue is indeed challenging that will require further research effort and will be addressed in our future work.

## V. Performance Evaluation

In order to demonstrate the effectiveness of our proposed algorithm, we evaluate the performance of LTRT via extensive simulations, and compare it with state-of-the-art algorithms including LMST, TRT, CBTC($\frac{2\pi}{3k}$), and $\text{FLSS}_k$. CBTC($\frac{2\pi}{3k}$) and $\text{FLSS}_k$ are the typical algorithms to acquire $k$-vertex connected topology. To the best of our knowledge, FLSS shows the best performance in contrast with all other topology control algorithms. The topology of $\text{FLSS}_k$ is nearly optimal but the computational cost is rather high. We assume this algorithm to be nearly optimal and to perform the best in a localized manner.

Simulations have been conducted by using our self-made C++ simulator for topology control. We generated a network with nodes randomly placed in a square region, and compared the performance of the topologies derived under each algorithm. The length of the square region is $1000[\mathrm{m}]$. Each node has a maximum transmission radius of $r_{max} = 250[\mathrm{m}]$. We

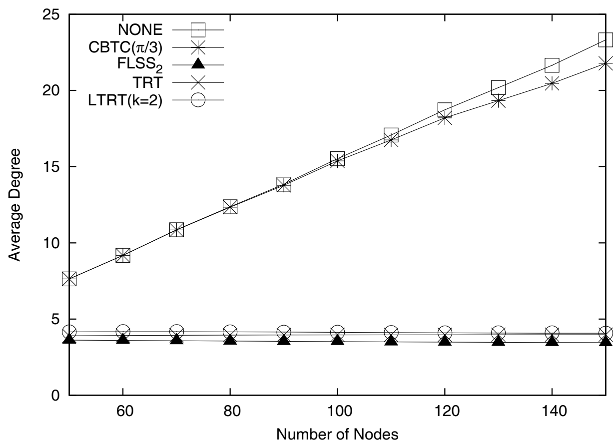


Fig. 6. Average degree.

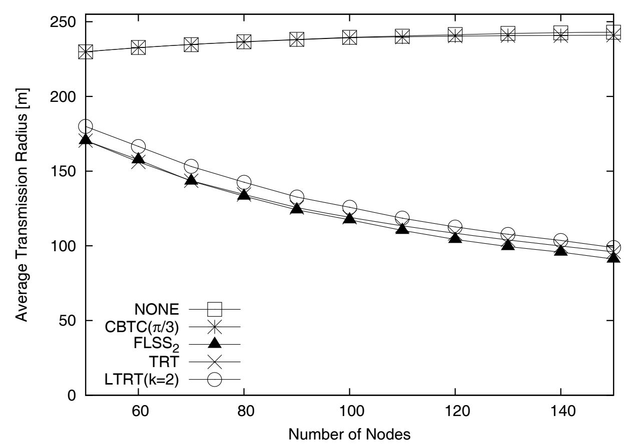


Fig. 8. Average radius.

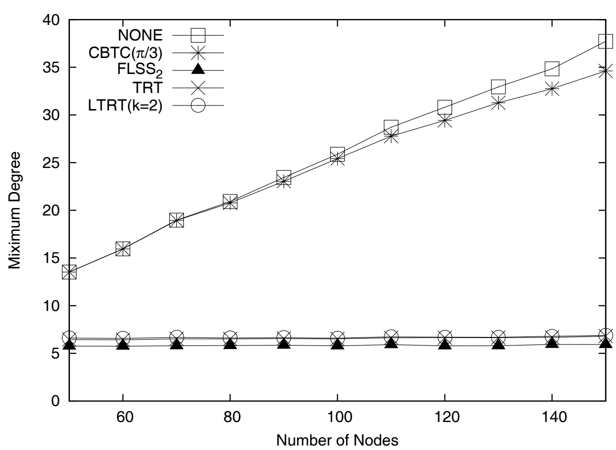


Fig. 7. Maximum degree.

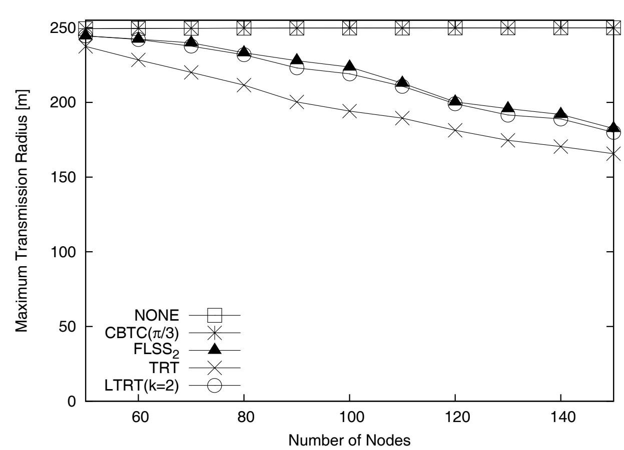


Fig. 9. Maximal radius.

vary the number of nodes from $50$ to $150$ to change the node density. Simulations are executed $100$ times. Resulting values are obtained by averaging over $100$ runs of simulations.

*Topologies:* Fig. 5 shows the resulting topologies of the 100-node network constructed by the respective algorithms. Fig. 5(a) illustrates that the network introduces high interference if no topology control algorithm is applied. Among the algorithms, LMST acquires an MST-like topology with the least number of edges, but link failures can easily fragment the network since the network is only 1-connected network. CBTC($\frac{\pi}{3}$) has more edges than that of LTRT or FLSS$_k$. Note that LTRT, though results in more edges than those of FLSS$_k$, is quite compatible with FLSS$_k$, which provides a nearly optimal solution.

*Node degree:* The node degree is defined as the number of neighbors of a node and is an indication of the level of MAC interference. If the node degree is smaller, the potential interference is lower. Fig. 6 shows the average node degree obtained by different algorithms, which ensure 2-connectivity, namely, CBTC($\frac{\pi}{3}$), FLSS$_k$, TRT, and LTRT ($k = 2$). FLSS$_2$ demonstrates the best performance while CBTC($\frac{\pi}{3}$) achieves much higher node degree than other algorithms. LTRT performs nearly the same as that of TRT, and the value is about $0.5$ higher than that of FLSS. Fig. 7 illustrates maximum value of node degree. FLSS$_2$, LTRT, and TRT achieve the smallest maximum degree which is independent of the node density. These results indicate that the MAC level interference can be lowered by LTRT even in the highly dense network.

*Transmission radius:* Next, we compare the transmission radius of the topologies derived by different algorithms. The transmission radius is the longest link length of each node that is directly proportional to energy consumption. Fig. 8 shows the average radius. It shows that the value obtained by LTRT is close to that of TRT, and about $5[\%]$ higher than that of FLSS for any node density. Fig. 9 illustrates the average of the maximal transmission radius. Nodes in the sparse region consume more energy than other nodes, and this result shows the transmission radius of such nodes. Since TRT is the global algorithm, TRT achieves the best performance. In addition, LTRT outperforms FLSS$_2$ to some extent.

*Energy Expended Ratio (EER):* We define EER as follows:

$$EER = \frac{E_{ave}}{E_{max}} \times 100[\%],$$

where $E_{ave}$ is the average transmission power over all the nodes in the network, and $E_{max}$ is the maximal transmission power that can reach the transmission range of $250[m]$. We adopt the model used in [6], [8], [17], $E = r^2$ as the energy model, where $r$ is the transmission radius. Fig. 10 illustrates EER obtained by different algorithms. This feature is almost

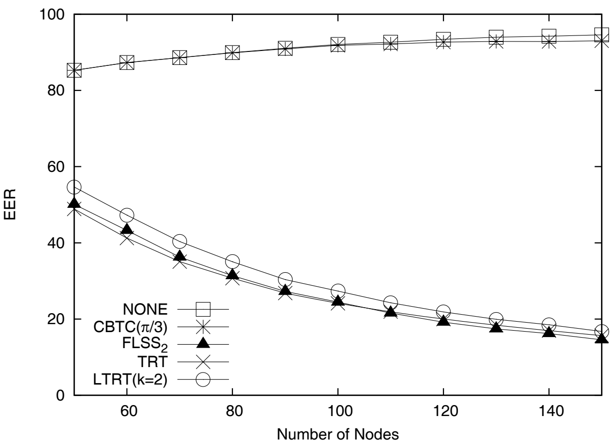


Fig. 10. Energy Expended Ratio.

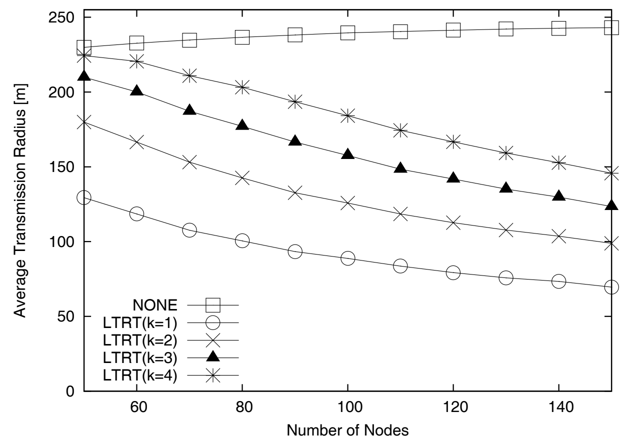


Fig. 12. Average transmission radius.

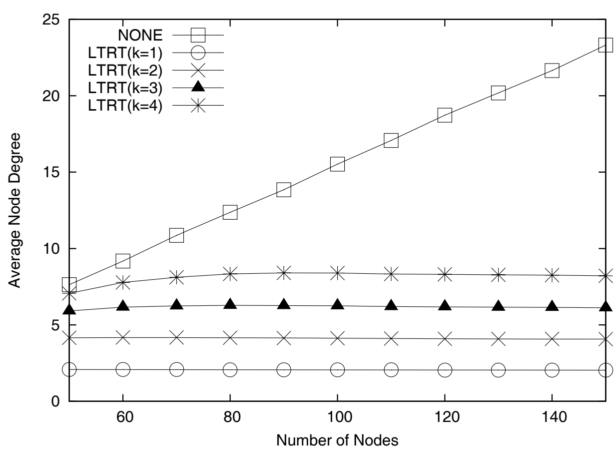


Fig. 11. Average node degree.

the same as the average transmission radius. The difference of the values between FLSS and LTRT diminishes as the density increases. This is attributed to the fact that the information of neighboring nodes can be readily acquired by each node as the network becomes denser.

*Trade-off between connectivity and fault-tolerance:* Finally, we compare the node degree and transmission radius with varying connectivity. We vary the number of connectivity $k$ of LTRT from $1$ to $4$. When $k = 1$, LTRT is the same as LMST.

Figs. 11 and 12 illustrate the node density and the transmission radius, respectively. The average node degree increases by about 2 as the connectivity $k$ is incremented by one. The average transmission radius increases from $20\%$ to $40\%$. Maintaining network reliability requires high energy, but LTRT can minimize the increase of energy consumption and interference.

*Discussion:* From these results, LTRT outperforms CBTC($\frac{2\pi}{3k}$), and it performs almost the same as FLSS, which is $k$-vertex connected and near optimal. Since the computational cost of LTRT is much lower than that of $\text{FLSS}_k$, LTRT is more desirable in the sense that LTRT achieves comparable performance as that of $\text{FLSS}_k$, i.e., it provides a nearly optimal solution, but at a much lower computational cost.

## VI. Conclusion

We have considered the broadcast transmission in ad-hoc networks. The objective of topology control is to minimize transmission power of each node while maintaining network connectivity. If the topology is adequately controlled, the resulting topology preserves network connectivity, but usually at the cost of transmission reliability. In this paper, we have proposed LTRT, which preserves $k$-edge connectivity of the network to ensure reliable transmission. LTRT incorporates the idea of LMST with TRT. $k$-edge connected LTRT is easily constructed by repeating the topology construction and link deletion phase. Since the computational cost of LTRT is the same as that of required by Prim's algorithm for the most part, the actual computational cost is rather low. Therefore, LTRT is readily applicable to resource constrained ad-hoc networks. In fact, LTRT can be applied to various types of networks because it is computationally cost effective and yields high energy saving with reliable transmission. For example, LTRT can be constructed without much delay because of its low computational cost even if the processing capability of devices is restricted. Even when the network environment changes, LTRT can adaptively change the network connectivity.

We have evaluated the performance of LTRT through extensive simulations, and demonstrated that LTRT achieves comparable performance to that of the near-optimal algorithm, but at a much lower computational cost. Our evaluation has demonstrated the effectiveness of LTRT, and we can say that LTRT is more readily applicable to real networks than any other existing topology control algorithms.

## Acknowledgements

This work was supported through the International Communications Research Grant (A Study on Reliable Ad hoc Networks, 2009), International Communications Foundation (ICF).

## References

[1] S. Guo and O. W. W. Yang, "Energy-aware multicasting in wireless ad hoc networks: a survey and discussion," *Comput. Commun.*, vol. 30, no. 9, pp. 2129-2148, 2007.

[2] L. M. Kirousis, E. Kranakis, D. Krizanc, and A. Pelc, “Power consumption in packet radio networks," *Theoretical Computer Science*, vol. 243, no. 1-2, pp. 289-305, 2000.

[3] A. E. Clementi, P. Penna, and R. Silvestri, “On the power assignment problem in radio networks," *Mobile Networks Applications*, vol. 9, no. 2, pp. 125-140, 2004.

[4] E. L. Lloyd, R. Liu, M. V. Marathe, R. Ramanathan, and S. Ravi, “Algorithmic aspects of topology control problems for ad hoc networks," *Mobile Networks Applications*, vol. 10, no. 1, pp. 19-34, 2005.

[5] J. Wieselthier, G. Nguyen, and A. Ephremides, “On the construction of energy-efficient broadcast and multicast trees in wireless networks," in *Proc. IEEE INFOCOM 2000*, vol. 2, 2000, pp. 585-594.

[6] N. Li, J. Hou, C. Sha, and L. Sha, “Design and analysis of an MST-based topology control algorithm," *IEEE Trans. Wireless Commun.*, vol. 4, no. 3, pp. 1195-1206, May 2005.

[7] N. Ansari, G. Cheng, and R. Krishnan, “Efficient and reliable link state information dissemination," *IEEE Commun. Lett.*, vol. 8, no. 5, pp. 317-319, May 2004.

[8] L. Li, J. Y. Halpern, P. Bahl, Y. Wang, and R. Wattenhofer, “Analysis of a cone-based distributed topology control algorithm for wireless multi-hop networks," in *Proc. ACM PODC 2001*, Aug. 2001, pp. 264-273.

[9] G. T. Toussaint, “The relative neighborhood graph of a finite planar set," *Pattern Recognition*, vol. 12, pp. 261-268, 1980.

[10] J. Cartigny, D. Simplot, and I. Stojmenovic, “Localized minimum-energy broadcasting in ad-hoc networks," in *Proc. IEEE INFOCOM 2003*, vol. 3, Mar. 2003, pp. 2210-2217.

[11] R. Komali, A. MacKenzie, and R. Gilles, “Effect of selfish node behavior on efficient topology design," *IEEE Trans. Mobile Computing*, vol. 7, no. 9, pp. 1057-1070, Sep. 2008.

[12] M. Cardei, J. Wu, and S. Yang, “Topology control in ad hoc wireless networks using cooperative communication," *IEEE Trans. Mobile Computing*, vol. 5, no. 6, pp. 711-724, 2006.

[13] J. Cartigny, F. Ingelrest, D. Simplot-Ryl, and I. Stojmenovic, “Localized LMST and RNG based minimum-energy broadcast protocols in ad hoc networks," *Ad Hoc Networks*, vol. 3, no. 1, pp. 1-16, 2005.

[14] F. Ingelrest, D. Simplot-Ryl, and I. Stojmenovic, “Optimal transmission radius for energy efficient broadcasting protocols in ad hoc and sensor networks," *IEEE Trans. Parallel Distributed Systems*, vol. 17, no. 6, pp. 536-547, 2006.

[15] N. Li and J. Hou, “BLMST: a scalable, power-efficient broadcast algorithm for wireless networks," in *Proc. QSHINE 2004*, Oct. 2004, pp. 44-51.

[16] M. Bahramgiri, M. Hajiaghayi, and V. Mirrokni, “Fault-tolerant and 3-dimensional distributed topology control algorithms in wireless multi-hop networks," in *Proc. IEEE ICCCN 2002*, Oct. 2002, pp. 392–397.

[17] N. Li and J. C. Hou, “Localized fault-tolerant topology control in wireless ad hoc networks," *IEEE Trans. Parallel Distributed Systems.*, vol. 17, no. 4, pp. 307-320, 2006.

[18] G. Di Battista, R. Tamassia, and L. Vismara, “Output-sensitive reporting of disjoint paths," *Algorithmica*, vol. 23, no. 4, pp. 302-340, 1999.

[19] S. Even and R. E. Tarjan, “Network flow and testing graph connectivity," *SIAM J. Computing*, vol. 4, no. 4, pp. 507-518, 1975.

[20] L. Lazos and R. Poovendran, “SeRLoc: secure range-independent localization for wireless sensor networks," in *Proc. ACM WiSe ’04*, 2004, pp. 21-30.

[21] A. Caruso, S. Chessa, S. De, and A. Urpi, “GPS free coordinate assignment and routing in wireless sensor networks," *Proc. IEEE INFOCOM 2005*, vol. 1, pp. 150-160, Mar. 2005.

[22] R. Prim, “Shortest connection networks and some generalizations," *Bell System Technical J.*, vol. 3, pp. 1389-1401, 1957.

[23] M. Fredman and R. Tarjan, “Fibonacci heaps and their uses in improved network optimization algorithms," *J. ACM*, vol. 34, no. 3, pp. 596-615, July 1987.


**Kenji Miyao** received the B.E. degree in information engineering from Tohoku University, Sendai, Japan, in 2008. Currently, he is pursuing the M.S. degree in the Graduate School of Information Sciences (GSIS) at Tohoku University. His research interests are in the areas of ad-hoc networks. He received the Presidential Award from Tohoku University in March 2007.

**Hidehisa Nakayama** received the B.E., M.S., and Ph.D. degrees in information sciences from Tohoku University, Sendai, japan, in 2000, 2002, and 2005, respectively.

He is a Senior Assistant Professor at the Tohoku Institute of Technology. He has been engaged in research on intelligent sensor technology, wireless mobile ad hoc network, computer networking, character string analysis, pattern recognition, and image processing.

Dr. Nakayama is a member of IEEE Communications Society, the Institute of Electronics, Information and Communication Engineers (IEICE), and the Information Processing Society of Japan (IPSJ). He received the Best Paper Award for Young Researcher of IPSJ Tohoku Chapter in 2000 and the Best Paper of Pattern Recognition Award in SCI 2003.

**Nirwan Ansari** (S’78-M83-SM’94-F’09) received the B.S.E.E. (summa cum laude) from the New Jersey Institute of Technology (NJIT), Newark, in 1982, the M.S.E.E. degree from University of Michigan, Ann Arbor, in 1983, and the Ph.D. degree from Purdue University, West Lafayette, IN, in 1988. He joined NJIT’s Department of Electrical and Computer Engineering as Assistant Professor in 1988, and has been Full Professor since 1997. He has also assumed various administrative positions at NJIT. He authored Computational Intelligence for Optimization (Springer, 1997, translated into Chinese in 2000) with E.S.H. Hou, and edited Neural Networks in Telecommunications (Springer, 1994) with B. Yuhas. His current research focuses on various aspects of broadband networks and multimedia communications. He has also contributed over 300 technical papers, over one third of which in refereed journals/magazines.

He is a Senior Technical Editor of the IEEE COMMUNICATIONS MAGAZINE, and also serves on the Advisory Board and Editorial Board of five other journals. He has been serving the IEEE in various capacities such as Chair of IEEE North Jersey COMSOC Chapter, Chair of IEEE North Jersey Section, Member of IEEE Region 1 Board of Governors, Chair of IEEE COMSOC Networking TC Cluster, Chair of IEEE COMSOC Technical Committee on Ad Hoc and Sensor Networks, and Chair/TPC Chair of several conferences/symposia. Some of his recent awards and recognitions include an IEEE Fellow (Communications Society), IEEE Leadership Award (2007, from Central Jersey/Princeton Section), the NJIT Excellence in Teaching in Outstanding Professional Development (2008), IEEE MGA Leadership Award (2008), the NCE Excellence in Teaching Award (2009), and designation as an IEEE Communications Society Distinguished Lecturer.

**Nei Kato** received his M.S. and Ph.D. Degrees in information engineering from Tohoku University, Japan, in 1988 and 1991, respectively.

He joined Computer Center of Tohoku University at 1991, and has been a full professor with the Graduate School of Information Sciences since 2003. He has been engaged in research on computer networking, wireless mobile communications, image processing and neural networks. He has published more than 160 papers in journals and peer-reviewed conference proceedings.

Nei Kato currently serves as Vice Chair of Technical Committee of Satellite Communications, IEICE, a technical editor of IEEE Wireless Communications(2006 ), an editor of IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS (2008 ), a co-guest-editor for the IEEE WIRELESS COMMUNICATIONS MAGAZINE special issue on wireless communications for E-healthcare. He has served as a symposium co-chair for GLOBECOM07 and ChinaCom08 and ChinaCom09. He is serving as IEEE WCNC2010 TPC Vice Chair(Network Track) and ICC 2010 Ad Hoc, Sensor and Mesh Networking Symposium Co-chair. His awards include Minoru Ishida Foundation Research Encouragement Prize(2003), Distinguished Contributions to Satellite Communications Award from the IEEE Communications Society, Satellite and Space Communications Technical Committee(2005), the FUNAI information Science Award(2007), the TELCOM System Technology Award from Foundation for Electrical Communications Diffusion(2008), and the IEICE Network System Research Award(2009). Besides his academic activities, he also serves on the expert committee of Telecommunications Council, Ministry of Internal Affairs and Communications, and as the chairperson of ITU-R SG4, Japan. Nei Kato is a member of the Institute of Electronics, Information and Communication Engineers (IEICE) and a senior member of IEEE.