

Topology Control Algorithm Using Fault-Tolerant 1-Spanner for Wireless Ad Hoc Networks*

Ruozi Sun, Yue Wang**, Jian Yuan, Xiuming Shan, Yong Ren

Department of Electronic Engineering, Tsinghua University, Beijing 100084, China

Abstract: A fault-tolerant 1-spanner is used to preserve all the minimum energy paths after node failures to cope with fault-tolerant topology control problems in wireless ad hoc networks. A fault-tolerant 1-spanner is a graph such that the remaining graph after node failures will not only remain connected, but also have a stretch factor of one. The fault-tolerant 1-spanner is used in a localized and distributed topology control algorithm, named the k -Fault-Tolerant 1-Spanner (k -FT1S), where each node constructs a minimum energy path tree for every local failed node set. This paper proves that the topology constructed by k -FT1S is a k -fault-tolerant 1-spanner that can tolerate up to k node failures, such that the remaining network after node failures preserves all the minimum energy paths of the remaining network gained from the initial network by removing the same failed nodes. Simulations show that the remaining network after removal of any k nodes still has the optimal energy efficiency and is competitive in terms of average logical degree, average physical degree, and average transmission radius.

Key words: fault-tolerant spanner; energy-efficiency; topology control; stretch factor

Introduction

Topology control in wireless ad hoc networks controls the network topology by adjusting the node transmission powers. Since network topology is the basis of good network performance, topology control is becoming a core issue in wireless networks and attracts much attention^[1].

Limited energy supplies of battery-powered wireless communication nodes and radio interference of wireless channels necessitate improving energy efficiency and reducing interference as the goals of topology control. From graph theory, topology control aims to construct a sparse spanning subgraph in an edge-dense

graph under different rules. The reduction of the number of paths between two nodes makes the network susceptible to node failures. Battery depletion or hardware failures cause frequent node failures in wireless ad hoc networks which are usually deployed as unmanned systems. The timely replacement of failed nodes is not feasible; thus, the network topology must be fault-tolerant so the network can maintain normal functions with node failures.

Most topology control studies considering fault-tolerance have used the connectivity to measure the fault-tolerant capability of the networks, such as CBTC_k^[2], FLSS_k^[3], k -UPVCS^[4], and LTRT^[5]. These algorithms all construct k -connected networks. In graph theory, the node connectivity of a graph is the minimum number of nodes which must be deleted to disconnect a network. A k -connected network can tolerate $k-1$ node failures, i.e., the failure of any $k-1$ nodes will not disconnect the network. Measuring fault-tolerance in terms of connectivity fails to consider the energy

Received: 2011-07-08; revised: 2012-02-20

* Supported by the National Natural Science Foundation of China (No. 60932005)

** To whom correspondence should be addressed.

E-mail: wangyue@tsinghua.edu.cn

efficiency of the remaining network after node failures. Network connectivity is not enough to guarantee high network energy efficiency. For example, the minimum spanning tree is connected, but its stretch factor could be $O(n)$ high, where n is the number of nodes. The stretch factor can be used to measure the network energy efficiency. For a k -connected network, the removal of high energy-efficient paths will reduce the energy efficiency of the remaining network. Though the network is connected, the nodes have to consume more energy for communications due to node failures. G_2^2 ^[6] and MBSS^[7] algorithms considered both fault-tolerance and network energy efficiency. They built 2-connected networks that preserved all the minimum energy paths. However, these two algorithms are not general since they can only tolerate one node failure. Additionally, they only aim to preserve all the minimum energy paths when no nodes fail, instead of preserving all the shortest paths after some node failures. The network is not only to remain connected after node failures, but also to have a stretch factor of one. The fault-tolerant spanner concept^[8] comes from computational geometry and was proposed in 1998. A fault-tolerant spanner can be informally defined to a graph such that the remaining graph after node failures is not only connected, but also has a predetermined stretch factor. A fault-tolerant spanner is a geometric structure that considers both the fault tolerance and the energy efficiency of the remaining network after node failures. Unfortunately, studies of fault-tolerant spanners have not given practical algorithms for constructing fault-tolerant spanners for topology control, since most algorithms are based on a complete geometry graph and are centralized, needing global information. Generally, topology control is based on unit disk graphs assuming that each node has the same maximal transmission range. Practical topology control methods should be localized and distributed because global information collection is costly.

This paper presents a localized, distributed topology control algorithm, called the k -Fault-Tolerant 1-Spanner (k -FT1S), to construct a fault-tolerant 1-spanner for wireless ad hoc networks. To the best of our knowledge, k -FT1S is the first algorithm using the fault-tolerant spanner concept for topology control and the first fully localized algorithm to construct a fault-tolerant 1-spanner on a unit disk graph.

The contributions of this paper are: (1) present a constraint condition for solving network fault-tolerance problems, (2) introduce the fault-tolerant spanner concept to characterize this constraint condition, (3) build a k -fault-tolerant 1-spanner using one-hop information in a unit disk graph, (4) theoretically prove that the topology generated by k -FT1S is really a k -fault-tolerant 1-spanner, and (5) present simulation results showing that the resulting networks preserve all the minimum energy paths after up to k node failures.

1 k -FT1S Algorithm

1.1 Fault-tolerant spanner

The graph distance is defined as:

Definition 1 Distance Let $G = (V, E)$ be an undirected graph with weight function W . For any two nodes $u, v \in V$, let $P_{\min}(u, v) = (v_0, v_1, \dots, v_L)$ be the shortest path between u and v in the sense of weight function W , where $v_l \in V$, $l = 0, 1, \dots, L$, $v_0 = u$, $v_L = v$. Then the length of the shortest path $\delta_G(u, v) = \sum_i W(v_i, v_{i+1})$

is called the distance between u and v in graph G .

Topology control aims to maintain high energy-efficient and good paths. The stretch factor can measure the network energy efficiency.

Definition 2 t -spanner and stretch factor Let G' be a spanning subgraph of G . If for any nodes $u, v \in G$, $\delta_{G'}(u, v) / \delta_G(u, v) \leq t$, where $\delta_{G'}(u, v)$ and $\delta_G(u, v)$ are the distances between u and v in G' and G respectively, then graph G' is called a t -spanner of G and the smallest real number t such that G' is a t -spanner of G is called the stretch factor of G' .

Topology control is the process of deleting edges in the maximal power network (the initial graph) for a given rule. The deletion of edges may increase the distance between two nodes. If the resulting graph G' is a t -spanner of G , then the distance between any two nodes in G' is at most stretched t times compared to that in the initial graph G . If the edge weights are defined as the power needed to communicate through the edge, the stretch factor can be an indicator of the network energy efficiency. If the resulting topology G' is a 1-spanner which preserves all the shortest paths, it is the theoretical optimum from the energy efficiency viewpoint. The network sparseness should be taken into account as well as the energy efficiency to reduce

the routing cost and radio interference. For a geometric or metric graph, bounded by the triangle inequality principle, except in the special case that all the nodes lay on the same line, the deletion of edges will increase the stretch factor. Fortunately, if edge weight is defined as the power needed for transmission on that edge, the triangle inequality no longer holds and the resulting topology can not only preserve all the shortest paths but also be sparse. Shen et al.^[9] proposed the Local Shortest Path (LSP) algorithm, which is a fully distributed and localized protocol. The nodes collaboratively construct a 1-spanner maintaining all the minimum energy paths using one-hop neighbor information. At the same time, the logical degree, physical degree, and transmission radius of the topology are greatly reduced compared with the initial network. However, since the network generated by the LSP algorithm is only 1-connected, one node failure may partition the network. The current network should accommodate multiple node failures.

Researchers have proposed many fault-tolerant algorithms based on k -connectivity such as CBTC_k^[2], FLSS_k^[3], and k -UPVCS^[4]. These algorithms are all based on k -connectivity, but the network topologies generated by these fault-tolerant algorithms do not preserve all the minimum energy paths of the initial network and the removal of nodes will degrade the network performance due to the increased network stretch factor. The network topologies constructed using the G_2^2 algorithm^[6] and the MBSS algorithm^[7] could maintain all the minimum energy paths and have 2-connectivity, but the authors did not discuss the stretch factor of the remaining network after the removal of failed nodes. Since both algorithms can only construct 2-connected networks, they are not general. Additionally, the algorithms do not perform well in the terms of network degree and transmission radius.

A k -fault-tolerant t -spanner can be defined as follows^[8].

Definition 3 k -fault-tolerant t -spanner and fault-tolerant stretch factor Let $G=(V,E)$ be an undirected graph. The weight function of G is W . Let t be a real number. Let $G'=(V',E')$ be a spanning subgraph of G , where $V'=V$, $E' \subseteq E$. If for any node set $F \subseteq V$, $|F| \leq k$, and for any two nodes $u, v \in V \setminus F$, $\delta_{G' \setminus F}(u, v) \leq t \delta_{G \setminus F}(u, v)$ holds, then G' is called a k -fault-tolerant t -spanner of G , and the smallest real number t such that G' is a k -fault-tolerant t -spanner

of G is called the fault-tolerant stretch factor of G' , where $\delta_{G' \setminus F}(u, v)$ and $\delta_{G \setminus F}(u, v)$ denote the distance between u and v in $G' \setminus F$ and $G \setminus F$ for weight function W .

From the above definition, the remaining graph $G' \setminus F$ obtained from G' by removing the failed node set F , together with its incident edges, is a t -spanner of the remaining graph $G \setminus F$ obtained from G by removing the same nodes and their incident edges. Thus, the k -fault-tolerant t -spanner has $(k+1)$ -connectivity and is a t -spanner of G at the same time. Researchers^[8,10-12] have proposed several centralized algorithms for constructing a k -fault-tolerant t -spanner in a planar point set or a geometric graph. Since these algorithms need global information which is costly to collect in wireless networks, they can not be used directly for topology control. The k -fault-tolerant t -spanner concept in computational geometry provides a new view for topology control but it does not provide a practical solution method for topology control. The ideal topology control algorithm should be localized, using only one-hop neighbor information. Each node in k -FT1S algorithm uses only one-hop neighbor location information to construct a k -fault-tolerant 1-spanner in the initial graph. Thus this provides network fault-tolerance and optimal network energy efficiency.

1.2 k -FT1S algorithm

The wireless ad hoc network is modeled as an undirected simple graph $G=(V,E)$, where V is the set of nodes, $E = \{(u, v) : d(u, v) \leq d_{\max}, u, v \in V\}$ is the set of edges, $d(u, v)$ is the Euclidean distance between u and v , and d_{\max} is the maximal transmission range of each node. The node locations are given. The visible neighborhood for each node u is defined as $NV_u = \{v \in V(G) : d(u, v) \leq d_{\max}\}$. Let $G_u = (NV_u, E_u)$ be the induced subgraph of graph G , where $E_u = \{(u, v) : d(u, v) \leq d_{\max}, u, v \in NV_u\}$. Let G' denote the constructed topology for the k -FT1S algorithm.

k -FT1S algorithm is as follows:

Step 1 Information collection

Each node u periodically broadcasts a Hello message with its maximal transmission power to get the visible neighborhood NV_u . Node u constructs its G_u using the neighbor location information. The weight function W of graph G_u is defined as $W(u, v) = d(u, v)^\alpha$, where $2 \leq \alpha \leq 5$.

Step 2 Topology construction

Let integer c be the number of node failures that the resulting topology can tolerate. Set $c=k$ for a k -fault-tolerant 1-spanner.

(1) For each node u , in its local graph G_u , any c nodes are arbitrarily selected from the node set $NV_u \setminus \{u\}$ as the local failed node set. There are $JJ = \binom{|NV_u|-1}{c}$ different such local failed node sets,

where $\binom{|NV_u|-1}{c}$ is the combinatorial number. Denote

the element index of the failed node set class as $j \in J = \{1, 2, \dots, JJ\}$. Denote the j -th local failed node set as $F_c^j = \{v_{j,1}, v_{j,2}, \dots, v_{j,c}\}$, $j \in J$, where $v_{j,l} \in NV_u \setminus \{u\}$, $l = 1, 2, \dots, c$.

(2) For any local failed node set F_c^j , $j \in J$, let F_c^j and its incident edges be removed from graph G_u , with the remaining graph denoted as $G_u^{F_c^j}$. The node set of graph $G_u^{F_c^j}$ is $V(G_u^{F_c^j}) = NV_u \setminus F_c^j$. For any node $w_m \in V(G_u^{F_c^j}) \setminus \{u\}$, $m = 1, 2, \dots, |NV_u| - c - 1$, calculate the shortest path $P_{\min}(u, w_m) = (u, w_{0,m}, w_{1,m}, \dots, w_m)$ from node u to node w_m , where $w_{h,m} \in V(G_u^{F_c^j}) \setminus \{u\}$, $h = 0, 1, 2, \dots$. If $w_{0,m} \neq w_m$ and $W(u, w_m) = W(P_{\min}(u, w_m))$, then let $w_{0,m} = w_m$. This operation guarantees the uniqueness of the shortest path. The second node $w_{0,m}$ in the path $P_{\min}(u, w_m)$ is selected as a neighbor of node u . After all the shortest paths from node u to node w_m , $m = 1, 2, \dots, |NV_u| - c - 1$ are calculated, obtain node u 's logical neighbor set for the local failed node set F_c^j with this neighbor set for F_c^j denoted as $N_u^{c,j} = \bigcup_{m=1}^{|NV_u|-c-1} \{w_{0,m}\}$.

(3) For all the local failed node sets F_c^j , $j = 1, 2, \dots, JJ$, node u executes the same calculation as in Step (2) to get node u 's ultimate logical neighbor set $N_u^c = \bigcup_{j=1}^{JJ} N_u^{c,j}$.

Step 3 Transmission power determination

Each node u adjusts its transmission radius to the minimum value needed to reach the farthest neighbor in N_u^c .

Each node u executes the calculations in Steps 1 to 3 to generate the final network topology G' and determine the transmission radius for each node.

1.3 Theoretical basis of k -FT1S

This part theoretically proves that the network

topology G' constructed using k -FT1S is really a k -fault-tolerant 1-spanner of the initial network G . The LSP^[10] algorithm is introduced for use in the conclusion. The LSP algorithm consists of three steps introduced using the same symbol definitions as in Section 1.2.

(1) Information collection: Each node u gets the edge weights of the induced subgraph G_u induced by NV_u using local neighbor information.

(2) Topology calculation: In graph G_u , node u calculates the shortest path from node u to node v_i for each node $v_i \in NV_u \setminus \{u\}$, $i = 1, 2, \dots, |NV_u| - 1$ and denotes the shortest path as $P(u, v_i) = (u, v_{0,i}, v_{1,i}, \dots, v_i)$, where $v_{l,i} \in NV_u \setminus \{u\}$, $l = 1, 2, \dots$. The second node $v_{0,i}$ in the path $P(u, v_i)$ is chosen as one neighbor of node u and then node u 's logical neighbor set is $N_u^0 = \bigcup_{i=1}^{|NV_u|-1} v_{0,i}$.

(3) Adjustment of the transmission power: The transmission power for each node u is adjusted to the minimum needed by node u to reach the farthest node in its neighbor set N_u^0 .

Each node executes the calculations in Steps 1 to 3 to construct the final topology G_{LSP} . LSP is equivalent to k -FT1S for $k=0$. Shen et al.^[9] proved the following conclusion for the LSP algorithm.

Conclusion 1 Let G be the initial graph. Let G_{LSP} be the topology constructed using the LSP algorithm. Then G_{LSP} is a 1-spanner of G .

The topology G' generated using k -FT1S should be a k -fault-tolerant 1-spanner of G based on the following Theorem 1.

Theorem 1 Assume that the initial graph G has $(k+1)$ -connectivity. Any node u of G executes the calculations in Step 2 in the k -FT1S algorithm with $c = c_1$ and $c = c_2$, where $0 \leq c_1 < c_2 \leq k < n_u - 1$, $n_u = |NV_u|$. If node u 's resulting neighbor set N_u^c for $c = c_1$ is $N_u^{c_1}$ and for $c = c_2$ is $N_u^{c_2}$, then $N_u^{c_1} \subseteq N_u^{c_2}$.

Proof Denote node u 's visible neighbor set as NV_u and its induced subgraph as G_u . Let $J = \{1, 2, \dots,$

$\binom{|NV_u|-1}{c_1}\}$ and $I = \{1, 2, \dots, \binom{|NV_u|-1}{c_2}\}$. For $c = c_1$,

arbitrarily select c_1 nodes from set $NV_u \setminus \{u\}$ as a local failed node set denoted as $F_{c_1}^j = \{v_{j,1}, v_{j,2}, \dots, v_{j,c_1}\}$, where $j \in J$, $v_{j,l} \in NV_u \setminus \{u\}$, $l = 1, 2, \dots, c_1$. Assume that

$F_0^j = \emptyset$. For $c = c_2$, arbitrarily select c_2 nodes from set $NV_u \setminus \{u\}$ as a local failed node set denoted as $F_{c_2}^i = \{v_{i,1}, v_{i,2}, \dots, v_{i,c_2}\}$, where $i \in I$, $v_{i,l} \in NV_u \setminus \{u\}$, $l = 1, 2, \dots, c_2$. For any local failed node set $F_{c_1}^j (F_{c_2}^i)$, its cardinality is $c_1 (c_2)$. Since $c_1 < c_2$, there is always an index subset $I_1 \subset I$ such that for any $i_1 \in I_1$, $F_{c_1}^j \subset F_{c_2}^{i_1}$ holds. The algorithm calculates the shortest path trees rooted at node u for all the local failed node sets $F_{c_2}^i$, $i = 1, 2, \dots, \binom{|NV_u| - 1}{c_2}$. For any $z \in N_u^{c_1}$, since

$c_2 < n_u - 1$, there is always an index $i_1 \in I_1$ such that node z is a node of the remaining graph $G_u^{F_{c_2}^{i_1}}$ which is obtained from graph G_u by removing the failed node set $F_{c_2}^{i_1}$ and its incident edges, $z \in V(G_u^{F_{c_2}^{i_1}})$. Since $G_u^{F_{c_2}^{i_1}} \subset G_u^{F_{c_1}^j}$, then $z \in N_u^{c_2, i_1}$, where $N_u^{c_2, i_1}$ is node u 's neighbor set for the local failed node set $F_{c_2}^{i_1}$. Since $N_u^{c_2, i_1} \subseteq N_u^{c_2}$, then $z \in N_u^{c_2}$, so $N_u^{c_1} \subseteq N_u^{c_2}$. \square

Theorem 2 If the initial graph G is $(k+1)$ connected, then the topology G' generated by the k -FTIS algorithm is a k -fault-tolerant 1-spanner of G .

Proof Denote the initial graph as $G = (V, E)$ and the topology constructed using k -FTIS as $G' = (V, E')$, where the cardinality of the vertex set is $|V| = n$. Arbitrarily select k nodes from node set V as a global failed node set denoted by $F = \{v_1, v_2, \dots, v_k\}$, where $v_l \in V$, $l = 1, 2, \dots, k$. Remove the global failed node set F and its incident edges from G and G' , so that the remaining graphs are $G^F = (V^F, E^F)$ and $G'^F = (V^F, E'^F)$, where $V^F = V \setminus F$ and E^F and E'^F are the edge sets obtained from E and E' by deleting the failed set F 's incident edges. Since graph G is $(k+1)$ connected, then graph G^F is connected. Now, we only need to prove that graph G'^F is a 1-spanner of graph G^F , which means that G'^F preserves all the shortest paths in G^F .

Denote node u 's visible neighbor set as $NV_u = \{v \in V: d(u, v) \leq d_{\max}\}$. The induced subgraph by NV_u in G is $G_u = (NV_u, E_u)$. For any $u \in V^F$, let its visible neighbor set in graph G^F be $NV_u^F = \{v \in V^F: d(u, v) \leq d_{\max}\}$ and the induced subgraph by NV_u^F in G^F is $G_u^F = (NV_u^F, E_u^F)$. $G^F = \bigcup_{u \in V^F} G_u^F = \left(\bigcup_{u \in \{u: NV_u \cap F \neq \emptyset\}} G_u^F \right) \cup$

second equation is the subgraph not related to the global failed node set F , and the second part is the subgraph related to F .

For any node $u \in V^F$ and $NV_u \cap F = \emptyset$, then $NV_u^F = NV_u$, $E_u^F = E_u$, and $G_u = G_u^F$. The k -FTIS algorithm will calculate the shortest path tree rooted at node u in graph G_u (also G_u^F) in Step 2 with node u 's neighbor set N_u^k . By Theorem 1, since $0 < k$, $N_u^0 \subseteq N_u^k$, where N_u^0 is the neighbor set generated by the LSP algorithm.

For any node $u \in V^F$ and $NV_u \cap F \neq \emptyset$, let $NV_u \cap F = F_a^{I_u}$, where $a = |F_a^{I_u}| \leq |F_k^j| = k$, $I_u \in \left\{1, 2, \dots, \binom{|NV_u| - 1}{a}\right\}$.

Since $u \notin F_k^j$, the algorithm will calculate the shortest path trees rooted at node u in all the graphs $G_u^{F_k^j}$, $j = 1, 2, \dots, \binom{|NV_u| - 1}{a}$ in Step 2 and the logical neighbor

set N_u^k . Since $G_u^{F_a^{I_u}} = G_u^F$, let N_u^{a, I_u} be node u 's sub-neighbor set generated by calculating the shortest path tree rooted at node u in graph $G_u^{F_a^{I_u}}$ (also G_u^F). Given $|F_a^{I_u}| = a \leq |F_k^j| = k$, when $a = k$, there is always a $\hat{j} \in \left\{1, 2, \dots, \binom{|NV_u| - 1}{a}\right\}$ such that $F_a^{I_u} = F_k^{\hat{j}}$. Then $N_u^{a, I_u} = N_u^{k, \hat{j}}$. Thus $N_u^{k, \hat{j}} \subseteq N_u^k$ and $N_u^{a, I_u} \subseteq N_u^k$. When $a < k$, by Theorem 1, $N_u^a \subseteq N_u^k$. Since $N_u^{a, I_u} \subseteq N_u^a$, then $N_u^{a, I_u} \subseteq N_u^k$. Define graph $G^{*F} = (V^F, E^{*F})$, where $E^{*F} =$

$$\bigcup_{u \in \{u: NV_u \cap F = \emptyset\}} \left(\bigcup_{v \in N_u^0} \{(u, v)\} \right) \cup \bigcup_{u \in \{u: NV_u \cap F \neq \emptyset\}} \left(\bigcup_{v \in N_u^{a, I_u}} \{(u, v)\} \right).$$

Since graph G^F is connected and G^{*F} is the topology constructed using the LSP algorithm in graph G^F , then G^{*F} is a 1-spanner of graph G^F . In

$$\text{graph } G'^F = (V^F, E'^F), E'^F = \bigcup_{u \in \{u: NV_u \cap F = \emptyset\}} \left(\bigcup_{v \in N_u^k} \{(u, v)\} \right) \cup$$

$$\bigcup_{u \in \{u: NV_u \cap F \neq \emptyset\}} \left(\bigcup_{v \in N_u^k} \{(u, v)\} \right). \text{ For any node } u \text{ such that}$$

$NV_u \cap F = \emptyset$, $N_u^0 \subseteq N_u^k$. For any node u such that $NV_u \cap F \neq \emptyset$, $N_u^{a, I_u} \subseteq N_u^k$. Therefore, $E^{*F} \subseteq E'^F$ and then $G^{*F} \subseteq G'^F$. Since G^{*F} is a 1-spanner of G^F , then G'^F is a 1-spanner of G^F . According to the definition of the k -fault-tolerant t -spanner, G' is a k -fault-tolerant 1-spanner of G . \square

Conclusion 2 If G is $(k+1)$ connected, then G' is also $(k+1)$ connected.

Conclusion 3 G' is a 1-spanner of G .

These two conclusions hold based on the definition of the k -fault-tolerant t -spanner.

1.4 Complexity analysis

In graph $G_u = (NV_u, E_u)$, let the number of vertices be $N = |NV_u|$ and the number of edges be $M = |E_u|$. k -FT1S will calculate the shortest path tree for $\binom{N-1}{k}$ cases. The computational complexity of the shortest path tree is $O(M + N \log N)$ by the Dijkstra algorithm^[13], so the computational complexity of the k -FT1S algorithm is $O(N^k M + N^{k+1} \log N)$. The computational complexity of k -FT1S is compared with FLSS $_k$ which is the best fault-tolerant topology control algorithm based on k -connectivity. The computational complexity of the FLSS $_k$ algorithm is $O(M^2 \sqrt{N})$ ^[3] and not a function of k . In a dense network, $M \approx N^2$, so complexity of the 1-FT1S algorithm is $O(N^3)$, while the corresponding complexity of FLSS $_2$ is $O(N^4 \sqrt{N})$. Thus the complexity of 1-FT1S is lower than that of FLSS $_2$. When $k=2$, the complexity of 2-FT1S is $O(N^4)$ and the complexity of FLSS $_3$ is $O(N^4 \sqrt{N})$, so the complexity of 2-FT1S is still lower than that of FLSS $_3$. For $k > 2$, the complexity of the k -FT1S algorithm becomes greater than that of the FLSS $_{k+1}$ algorithm.

2 Simulation Results

This section demonstrates the effectiveness of the k -FT1S algorithm based on five network performance metrics:

(1) Average logical degree: The logical degree of a node is defined as the number of logical neighbors in the resulting topology. A lower average logical degree usually means a sparse network and lower routing costs.

(2) Average physical degree: The physical degree is defined as the number of nodes within the node's transmission radius. A lower average physical degree usually implies less contention, lower interference, and higher network capacity.

(3) Average transmission radius: A smaller average transmission radius implies that the network can be constructed by lower energy consumption.

(4) Stretch factor: A smaller stretch factor means a higher network energy efficiency.

(5) Fault-tolerant stretch factor: A smaller fault-tolerant stretch factor means that the residual network obtained by removing k failed nodes and their incident edges has a higher energy efficiency.

These five performance metrics were calculated for networks constructed using the k -FT1S, FLSS $_k$, and G_2^2 . The FLSS $_k$ algorithm was shown to perform the best among various algorithms based on the k -connectivity^[3]. The network topologies constructed using FLSS $_k$ have lower logical degrees, lower physical degrees, and higher network capacities than other algorithms and the authors showed that the network topology generated by FLSS $_k$ has the minimum maximum transmission radius among all the fully localized algorithms. G_2^2 and MBSS construct bi-connected networks that preserve all the minimum energy paths, which are similar to the k -FT1S algorithm. However, these two studies did not analyze the performance of the remaining network obtained by removing the failed nodes and the algorithms only work for 2-connectivity networks. The networks generated by k -FT1S not only have arbitrary k -connectivity, but also are k -fault-tolerant 1-spanners. k -FT1S is not compared with MBSS because MBSS is not a fully localized algorithm and needs global information.

A total of n nodes are randomly distributed in a 1000×1000 region and each node's transmission range is $d_{\max} = 250$. The number of the node n varied from 100 to 200. Each data point is the average of 1000 simulation runs. All the algorithms constructed 2-connected networks.

The average logical degree of the topologies derived by k -FT1S, FLSS $_k$, and G_2^2 is shown in Fig. 1. The

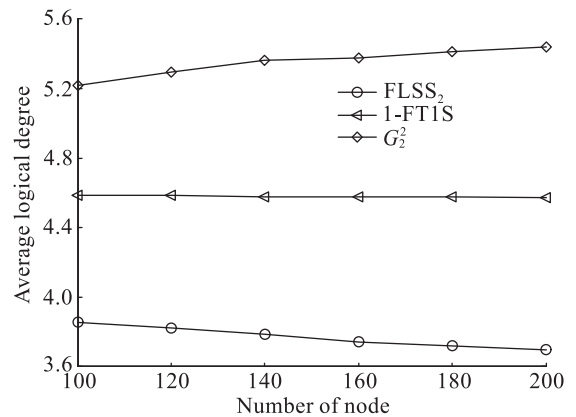


Fig. 1 Average logical degrees for 2-connected networks

logical degrees with k -FT1S are higher than with $FLSS_k$ while G_2^2 is the highest. In addition, in contrast to G_2^2 , the logical degrees for both the k -FT1S and $FLSS_k$ networks decrease with an increase in the number of nodes. The average physical degrees in k -FT1S, $FLSS_k$, and G_2^2 are shown in Fig. 2. The results are similar to those for the logical degree. The average transmission radius with k -FT1S is higher than with $FLSS_k$ and smaller than G_2^2 in Fig. 3. Thus, $FLSS_k$ performs the best, k -FT1S is in the middle, and G_2^2 is the worst.

The stretch factors for these three algorithms are shown in Fig. 4. The networks built using k -FT1S and G_2^2 are 1-spanners, while the stretch factor with $FLSS_k$ is more than 1 and increases with an increase in the number of nodes. Thus, $FLSS_k$ has the worst energy efficiency.

The fault-tolerant stretch factors for the three algorithms shown in Fig. 5 show that the stretch factor after the removal of failed nodes increases with $FLSS_k$, which implies degradation of the energy efficiency. The fault-tolerant stretch factor with k -FT1S is still 1,

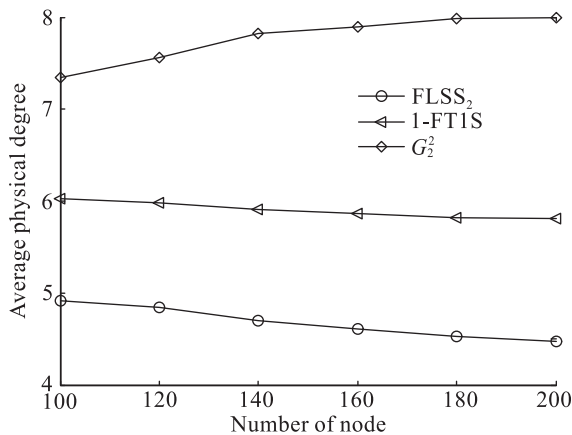


Fig. 2 Average physical degrees for 2-connected networks

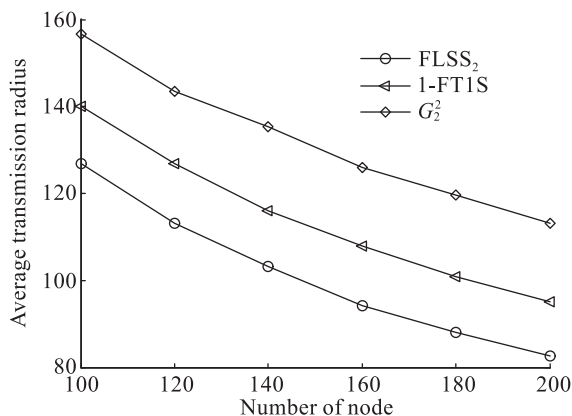


Fig. 3 Average transmission radii for 2-connected networks

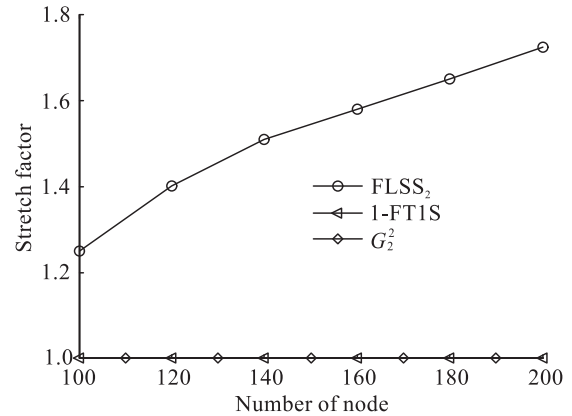


Fig. 4 Stretch factors for 2-connected networks

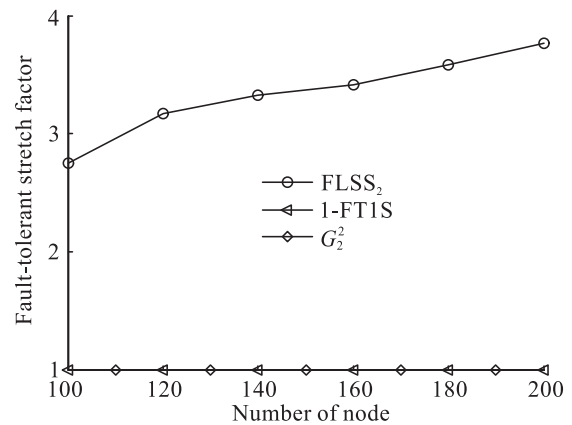


Fig. 5 Fault-tolerant stretch factors for 2-connected networks

which is consistent with Theorem 2. The fault-tolerant stretch factor with G_2^2 is also 1, but there is no discussion of this metric in the literature^[7]. The simulations show that G_2^2 constructs a 1-fault-tolerant 1-spanner, but there is no theoretical proof. Additionally, the algorithm only builds 2-connected networks, while k -FT1S works for an arbitrary k -connectivity, and the logical degree, physical degree, and transmission radius with k -FT1S are lower than these of G_2^2 .

Thus, the simulations demonstrate that in terms of the logical degree, physical degree, and transmission radius, $FLSS_k$ performs the best, k -FT1S is second, and G_2^2 is the worst. k -FT1S and G_2^2 give the same stretch factors and fault-tolerant stretch factors, while $FLSS_k$ is the worst.

3 Conclusions

This paper introduces a geometric structure fault-tolerant spanner for topology control in wireless ad hoc networks. A localized topology control algorithm,

k -FT1S is developed to improve the energy efficiency of the remaining network after some nodes fail and leave the network. In k -FT1S, each node constructs shortest path trees rooted at itself in the local remaining graphs obtained by removing any k neighbor nodes and their incident edges, and sets the one-hop neighbor in the shortest path tree as its neighbor in the resulting topology. The topology constructed under k -FT1S is proven to be a k -fault-tolerant 1-spanner. Simulations demonstrate the effectiveness of k -FT1S with the remaining network preserving all the minimum energy paths after nodes fail. Other network performances such as logical degree, physical degree, and transmission radius are slightly worse than with FLSS $_k$, the algorithm which performs the best in terms of the logical degree, physical degree, and transmission radius among algorithms based on k -connectivity.

The k -FT1S algorithm constructs a k -fault-tolerant 1-spanner. In the future, since the parameter t is relative to the sparseness of the network, a k -fault-tolerant t -spanner will be developed for arbitrary parameter $t > 1$ to allow a tradeoff between sparseness and energy efficiency.

References

- [1] Santi P. Topology control wireless ad hoc and sensor networks. *ACM Computing Survey*, 2005, **37**: 164-194.
- [2] Bahramgiri M, Hajiaghayi M, Mirrokni V S. Fault-tolerant and 3-dimensional distributed topology control algorithms in wireless multihop networks. In: Proceedings of 11th Int'l Conf. Computer Comm. and Networks. Miami, USA, 2002: 392-397.
- [3] Li N, Hou J C. Localized fault-tolerant topology control in wireless ad hoc networks. *IEEE Transactions on Parallel and Distributed Systems*, 2006, **17**(4): 307-320.
- [4] Hajiaghayi M, Immorlica N, Mirrokni V S. Power optimization in fault-tolerant topology control algorithms for wireless multi-hop networks. In: Proceedings of Ninth ACM Int'l Conf. Mobile Computing and Networking. San Diego, USA, 2003: 300-312.
- [5] Miyao K, Nakayama H, Ansari N, et al. LTRT: An efficient and reliable topology control algorithm for ad-hoc networks. *IEEE Transactions on Wireless Communications*, 2009, **8**(12): 6050-6058.
- [6] Roy H, Kumar De S, Maniruzzaman M D, et al. Fault-tolerant power-aware topology control for ad-hoc wireless networks. In: Proceedings of Networking 2010, 9th International IFIP TC 6 Networking Conference. Chemrai, India, 2010: 303-314.
- [7] Shen Z, Chang Y, Cui C, et al. A fault-tolerant and minimum-energy path-preserving topology control algorithm for wireless multi-hop networks. In: Proceedings of International Conference on Computational Intelligence and Security. Xi'an, China, 2005: 864-869.
- [8] Levkopoulos C, Narasimhan G, Smid M. Efficient algorithms for constructing fault-tolerant geometric spanners. In: Proceedings of Thirtieth Annual ACM Symposium on Theory of Computing. Dallas, USA, 1998: 186-195.
- [9] Shen Y, Cai Y, Xu X. A shortest-path-based topology control algorithm in wireless multihop networks. *ACM SIGCOMM Computer Communication Review*, 2007, **37**(5): 29-38.
- [10] Narasimhan G, Smid M. Geometric Spanner Networks. Cambridge, UK: Cambridge University Press, 2007.
- [11] Chechik S, Langberg M. Fault-tolerant spanners for general graphs. In: Proceedings of the 41st Annual ACM Symposium on Theory of Computing. Bethesda, USA, 2009: 435-444.
- [12] Lukovszki T. New results on fault tolerant geometric spanners. In: Proceedings of the 6th International Workshop on Algorithms and Data Structures. Vancouver, Canada, 1999: 193-204.
- [13] Fredman M L, Tarjan R E. Fibonacci heaps and their uses in improved network optimization algorithms. *Journal of the ACM*, 1987, **34**(3): 596-615.