Efficient On-Demand Topology Control for Wireless Ad Hoc Networks

Yu Wang
Department of Computer Science
University of North Carolina at Charlotte
Charlotte, North Carolina 28223, USA
Email: ywang32@uncc.edu

Xinghua Shi
Department of Computer Science
University of Chicago
Chicago, Illinois 60637, USA
Email: shi@cs.uchicago.edu

***Abstract*— Topology control in wireless ad hoc networks has been heavily studied recently. Different geometric topologies were proposed to be used as the underlying network topologies, in order to achieve the sparseness of the communication network or to guarantee the package delivery of specific routing methods. However, most of the proposed topology control algorithms were applied for all nodes in the networks at the initial network startup stage, and the constructed topologies were kept being maintained thereafter. The overhead of topology control at each node at any time is notable, and this affects the performance of the network and wastes energy for each wireless node. This paper seeks to investigate the practices of efficient on-demand topology control protocol for wireless ad hoc networks. In our new protocol, we only apply the specific topology control technique where and when the wireless node needs it. Our simulation confirms that our scheme has better performance than several existing methods.**

I. Introduction

Wireless *ad hoc* networks have been undergoing a revolution that promises to have a significant impact throughout society. Unlike traditional fixed infrastructure networks, there is no centralized control over ad hoc networks, which consist of an arbitrary distribution of mobile devices in a certain geographical area. In ad hoc networks, mobile devices can communicate via multi-hop wireless channels: a device can reach all devices in its transmission range, while two far-away devices communicate through the messages relaying by intermediate devices. Ad hoc network intrigues many challenging research problems, as it intrinsically has many special characteristics and some unavoidable limitations, compared with other wired or wireless networks. An important requirement of an ad hoc network is that it should be self-organizing, i.e., transmission ranges and data paths are dynamically restructured with changing topology. Energy conservation and network performance are probably the most critical issues in ad hoc networks, because mobile devices are usually powered by batteries only and have limited computing capability and memory.

The *topology control* technique is to let each wireless device *locally* adjust its transmission range and select certain neighbors for communication, while maintaining a structure that can support energy efficient routing and improve the overall network performance. Unlike traditional wired networks and cellular wireless networks, mobile devices are often moving during the communication, which could change the network topology in some extent. Hence it is more challenging to design a topology control algorithm for ad hoc networks: the topology should be locally and self-adaptively maintained without affecting the whole network, and the communication cost during maintenance should not be too high. In the past several years, topology control algorithms [1]–[7] have drawn a significant amount of research interests. Comparing with centralized algorithms, distributed (or localized) algorithms are more suitable for wireless networks since the environment is inherently dynamic and they are adaptive to topology changes at the cost of possible less optimality. The primary distributed or localized topology control algorithms for ad hoc networks aim to maintain network connectivity, optimize network throughput with power-efficient routing, conserve energy and increase the fault tolerance.

Different topologies have different properties, however, most of the proposed topologies can not achieve all three preferred properties for unicast applications on ad hoc networks: power spanner, planar, degree-bounded. Until recently, Wang and Li [8] proposed a localized algorithm to build a degree-bounded planar spanner, which is based on the combination of *localized Delaunay triangulations* [9] and Yao structure [10]. It is the first localized algorithm that can achieve all the three desirable features, however, the theoretical node degree of their structure and the communication cost of their method can be large. Thus, Song *et al.* [11] further proposed two more communication efficient methods to construct small degree-bounded planar power spanners. Nonetheless, all of the proposed topology control algorithms were applied for all nodes in the network at the initial startup stage, and the constructed topologies were kept being maintained thereafter at each node. The overhead of topology control is notable, and thus affects the performance and power efficiency of the network. In this paper, we investigate the practices of efficient topology control protocol for ad hoc networks by introducing on-demand concepts. In our new protocol, the specific topology control technique is only applied where and when the wireless node needs it, in other words, the topology is maintained in an on-demand style.

The remainder of this paper is organized as follows. In Section II, we introduce the network model and summarize the preferred properties of the network topology. Section III provides an overview of the prior literature related to topology control and geographic routing. Section IV describes our new on-demand topology control protocol. Some simulation results are shown in Section V. Finally, the brief conclusion of our research work is highlighted in Section VI.

II. Preliminaries

A. Network Model

A wireless ad hoc network consists of a set V of n wireless nodes distributed in a two-dimensional plane. Each node has the same *maximum* transmission range. By a proper scaling, we assume that all nodes have the maximum transmission range of one unit. These wireless nodes define a *unit disk graph* UDG in which there is an edge between two nodes if and only if their Euclidean distance is at most one. In other words, two nodes can always receive the signal from each other directly if the Euclidean distance between them is no more than the maximum transmission range. Hereafter, UDG is always assumed to be connected. We also assume that all wireless nodes have distinctive identities and each node knows its position information either through a low-power GPS receiver or some other ways. By one-hop broadcasting, each node u can gather the location information of all nodes within its transmission range. As in the most common power-attenuation model, the power to support a link uv is assumed to be $\|uv\|^{\beta}$, where $\|uv\|$ is the Euclidean distance between u and v, β is a real constant between 2 and 5 depending on the wireless transmission environment.

B. Preferred Properties

Ad hoc network topology control protocols are to maintain a structure that can be used for efficient routing [12], [13] or improve the overall networking performance [1]–[3], by selecting a subset of links or nodes used for communication. In the literature, the following desirable features are well-regarded and preferred in wireless ad hoc networks:

Power Spanner: A good network topology should be energy efficient, that is to say, the total power consumption of the shortest path (most power efficient path) between any two nodes in final topology should not exceed a constant factor of the power consumption of the shortest path in original network. Given a path $v_1 v_2 \cdots v_h$ connecting two nodes v_1 and v_h, the energy cost of this path is $\sum_{j=1}^{h-1} \|v_j v_{j+1}\|^{\beta}$. The path with the least energy cost is called the shortest path in a graph. Formally speaking, a subgraph H is called a *power spanner* of a graph G if there is a positive real constant ρ such that for any two nodes, the power consumption of the shortest path in H is at most ρ times of the power consumption of the shortest path in G. The constant ρ is called the *power stretch factor*. A power spanner is usually energy efficient for routing.

Bounded-Degree: It is also desirable that node degree in the constructed topology is small and upbounded by a constant. A small node degree reduces the MAC-level contention and interference, and also may help to mitigate the well-known hidden and exposed terminal problems. Especially in Bluetooth-based ad hoc networks, the *master* node degree is preferred to be less than 7, according to Bluetooth specifications, to maximize the efficiency. In addition, a structure with small degree will improve the overall network throughout [17].

Planar: Many routing algorithms require the planar topology (in which no links intersect each other) to guarantee the message delivery, such as right hand routing, greedy face routing [12], greedy perimeter stateless routing [13], adaptive face routing [18], and greedy other adaptive face routing [14].

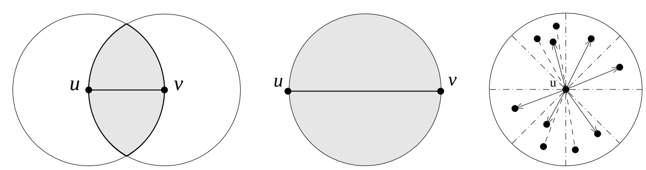
Efficient Localized Construction: Due to limited resources and high mobility of the wireless nodes, it is preferred that the underlying network topology can be constructed and maintained in a localized manner (each node only needs the information of nodes within a constant hops).

III. Priori Arts

A. Topology Control

Several geometrical structures have been proposed for topology control in ad hoc networks.

The *relative neighborhood graph* (RNG) consists of all edges uv such that the intersection of two circles centered at u and v and with radius $\|uv\|$ does not contain any node from V. The *Gabriel graph* (GG) contains edge uv if and only if $disk(u, v)$ contains no other node of V, where $disk(u, v)$ is the disk with edge uv as a diameter. See Figure 1(a) and (b) for illustration. Both GG and RNG are connected, planar if UDG is connected. In [19], [20], the RNG is used as the underlying topology for broadcasting and information dissemination. In [12], [13], RNG and GG are used as underlying routing topologies for their routing algorithm. However, in [5], Li *et al.* showed that the power stretch factor of RNG is $n-1$ while the power stretch factor of GG is 1, i.e., GG is power efficient for routing while RNG is not.



(a) RNG (b) GG (c) YG_8

Fig. 1. Definitions of RNG, GG, and YG. Shaded area is empty of nodes.

Recently, some researchers [3]–[5] proposed to construct the network topology based on Yao graph. The *Yao graph* [10] with an integer parameter $k > 6$, denoted by YG_k, is defined as follows. At each node u, any k equally-separated rays originating at u define k cones. In each cone, node u selects and adds the shortest edge $uv \in UDG$ among all edges emanated from u, if there is any. Ties are broken arbitrarily or by ID. See Figure 1(c) for an illustration of a Yao structure centered at u when $k = 8$. The resulting graph is called the Yao graph. It is known that the power stretch factor and the node out-degree of Yao graph are bounded by some positive constants. In [5], [6], Li *et al.* further proposed to use another sparse topology, *Yao and Sink*, which has both a constant bounded node degree and a constant bounded power stretch factor. However, all of these Yao-based graphs are not guaranteed to be planar.

Wang and Li [8] proposed the first efficient localized algorithm to build a degree-bounded planar spanner for ad hoc networks. In [11], Song *et al.* further proposed two more communication efficient methods to construct small degree-bounded planar power efficient structures.

Besides these geometrical structures, dynamic cluster techniques are also applied in topology control for ad hoc networks. Many methods [21]–[24] are proposed to build a virtual backbone for routing in ad hoc networks. However, most of these cluster-based methods do not use position information and the formatted virtual backbone is not a planar structure, so they are not suitable for the geometric routing methods.

There are also some research [25]–[27] on topology control for sensor networks. Most of them are based on the observation that sufficient density of sensor nodes requires only small number of nodes to be on at any time to forward traffic for active connections. These methods turn off as many nodes as possible without significantly diminishing the capacity or connectivity of the network to save power. However, they have a strong assumption that the density is high enough so that putting some nodes to sleep will not hurt the connectivity. This may be true in sensor networks due to large number of sensors, but in ad hoc networks, the number of wireless devices is usually not very large. Shutting down nodes in a sparse ad hoc network may hurt the performance of the network or even disconnect the network. In addition, in some ad hoc applications, every node needs to participate the communications, so that no node can be put to sleep. Thus, these topology management methods do not work for general ad hoc networks.

B. Geographic Routing

The geometric nature of the multi-hop wireless networks provides a promising idea: localized geometric routing. A routing protocol is *localized* if the decision to which node to forward a packet is based only on: 1) the information in the header of the packet (including the source and the destination of the packet); 2) the local information gathered by the node from a small neighborhood (i.e. 1-hop neighbors of the node). In order to make the localized geometric routing work, the source node has to learn the current location of the destination node. For sensor networks, the destination node (or called the sink node) is often fixed, thus, all nodes know its location. However, for most ad hoc networks applications, the help of location services is needed. Recently, algorithms for distributed location services are studied in [28], [29].

There are many localized geometric routing protocols [12], [15], [30], [31] proposed in the networking and computational geometry literature. The most common and efficient localized routing is *greedy routing*. It works as follows. Let t be the destination node. Current node u finds the next relay node v such that the distance $\|vt\|$ is the smallest among all neighbors of u in a given topology. Though greedy routing was widely used in early routing protocols for wireless networks, it is easy to construct an example to show that greedy algorithm will not succeed to reach the destination but fall into a local minimum (a node without any "better" neighbors). To guarantee the packet delivery after simple greedy heuristic fails, many methods [12]–[16] applied face routing as a backup.

Right hand rule is a long-known method for traversing one graph (in analogy to following the right hand wall in a maze). Applying the right hand rule in planar graphs, a routing protocol called *face routing* is proposed by [30] (they called it *Compass Routing II*). The idea of face routing is to walk along the faces which are intersected by the line segment st between the source s and the destination t. In each face, it uses the right hand rule to explore the boundaries. The authors of [18], [30] proved that the face routing algorithm guarantees to reach the destination after traversing at most $O(n)$ edges where n is the number of nodes when the underlying network topology is a planar graph.

Greedy routing is a simple and efficient method but can not guarantee the packet delivery, while face routing can guarantee the delivery but may take a very long exploration (O(n) steps). One natural approach to improve the performance of localized routing is to combine greedy routing and face routing by using face routing to recover the routing after simple greedy method fails in local minimum. Many wireless protocols used this approach [12]–[16]. The first combined method is *greedy face routing* (GFG) [12], where the authors used RNG or GG as the planar routing topology. It works as follows. When a node v receives a packet, it searches its neighbor table for a neighbor who is closer to the destination. If there is one, node v will forward the packet to that neighbor. When no neighbor is closer, the node v forwards the packet using a simple planar graph traversal (face routing) until the packet reaches another node u that is strictly closer to the destination than v, at which point the greedy method is resumed. GFG can guarantee the delivery of the packets when the underlying network topology is a planar graph. Recently, in [32], Qing, Gao and Guibas also studied how to find the stuck nodes in a sensor network where the greedy forwarding gets stuck in the local minimum. They gave a local rule (called *TENT* rule) for each node to test whether a packet can get stuck at that node. To help the packets get out of stuck nodes, they also developed an algorithm to build routes around *holes*, which are connected regions of the network with boundaries consisting of all the stuck nodes. The idea of their method is similar to the face routing.

IV. Our Scheme: On-Demand Topology Control

Although [8], [11] can efficiently construct degree-bounded planar power spanners for ad hoc networks, their algorithms were applied for all wireless nodes in the network at the initial network startup stage, and the constructed topologies were kept being maintained thereafter like most proposed topology control algorithms. Even though their algorithms were localized and used only linear number of communications, the overhead of topology control at each node is notable and non-ignorable. As people did in routing [33], one natural idea is applying on-demand concept in topology control: only applying the specific topology control technique where and when it is needed. This should improve the performance of the network and save energy for each wireless node.

On-demand concept is first applied to ad hoc routing in [33]. This paper is not the first attempt to introduce the on-demand concept to topology control. In [34], Zheng and Kravets applied the on-demand concept in power management. The nodes maintain soft-state timers that determine power management transitions. By monitoring routing control messages and data transmission, these timers are set and refreshed on-demand. Nodes that are not involved in data delivery may go to sleep, and nodes that are actively forwarding packets stay in active mode. However, in our paper, as we mentioned previously, we assume all nodes must be actively awake due to network connectivity or high-level applications. Thus, topology control protocol is focused on selecting certain neighbors for communications.

A. Basic Scheme

Remember that we would like to achieve bounded-degree planar spanner for topology control protocols. The easiest way to achieve bounded-degree spanner is applying Yao structure, while Gabriel graph is a sparse planar power spanner which can be constructed efficiently. Therefore, our on-demand topology protocol uses these two simple structures in an on-demand fashion.

First, to bound node degree, we apply Yao structure to a wireless node only when it has more than k neighbors. For those nodes who have less than k neighbors, we keep all of their neighbors as active neighbors. This not only saves the power of the node (only a few nodes need to perform Yao structure), but also increases the number of possible routes for future routing (more links in the constructed topology than Yao graph). We call the constructed topology *on-demand Yao graph* (ODYao).

Second, notice that the reason for requiring planar property is that we need to use face routing on planar topology to help the greedy routing get out from the local minimum. Therefore, there is no need to keep the whole network planar, we can apply the Gabriel graph only where the greedy routing fails or could fail. Assume a node v receives a packet and it can not find a neighbor who is closer to the destination t. It then applies Gabriel graph and forwards the packet using face routing. When a node u receives the face routing packet, it checks whether it is strictly closer to the destination than v. If yes, it resumes the greedy routing and forwards the packet to the neighbor who is closer to the destination t. If not, it will apply GG and continue the face routing until the packet reaches another node w that is strictly closer to the destination than v, at which point the greedy method is resumed. Comparing with using GG as the network topology, our method can still guarantee the packet delivery while dramatically reducing the number of nodes applying GG and increasing the number of possible routes for greedy routing.

Here, the on-demand concept covers two aspects: spatial on-demand and temporal on-demand. Spatially, we only apply the topology control techniques (Yao structure or Gabriel graph) on those nodes (with large node degree or where greedy routing fails) where the techniques are needed. Temporally, we only apply the topology control techniques when the techniques are needed. For example, a node u may initially have less than k neighbors so that it won't apply Yao structure, but later some new nodes join the networks which causes u with more than k neighbors. Then the Yao structure will be applied at u when the number of neighbor becomes larger than k. For Gabriel graph, the time of execution is dependent on the routing traffic. We apply the Gabriel graph only when the greedy routing fails and the face routing applies.

B. Nice Properties

Our on-demand topology control protocol not only saves energy for individual wireless devices and increases the possible routing routes for routing but also keeps all the preferred properties for topology control: connectivity, power spanner, bounded-degree, localized construction, and guaranteed packet delivery (with greedy face routing).

The connectivity and localized construction are straightforward. For power spanner, in [6], Li *et al.* proved the structure constructed by applying the Gabriel graph on Yao graph (denoted by GYao) is still a power spanner with the same power stretch factor as Yao graph. The proof is based on that the power stretch factor of Gabriel graph is one. Using the same technique, we can prove the structure constructed by applying the Gabriel graph on ODYao (denote by GODYao) is still a power spanner with the same power stretch factor of Yao graph. Notice that YG is a subgraph of ODYao. Then, since the structure constructed by our on-demand topology control is obviously a super-graph of GODYao, in other words, the GODYao is a subgraph of our structure. Therefore, our structure is still a power spanner, i.e., it is power efficient for routing.

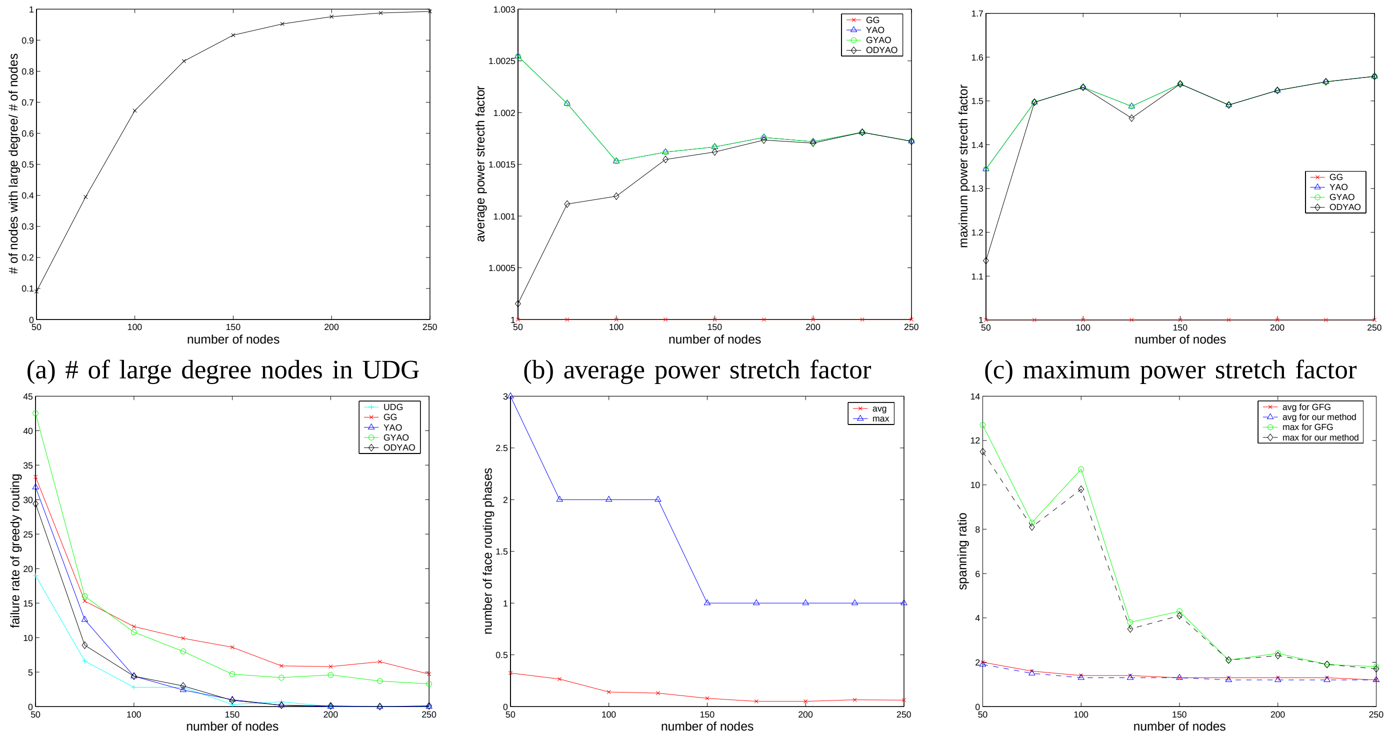
Notice that by applying Yao structure, the out-degree of each node is bounded by k for sure. If you want to bound both in-degree and out-degree, you can use the *Yao and Sink* structure [5], [6] instead of Yao structure. Here, we ignore it and only use Yao structure as a simple example to show the on-demand concept.

When greedy routing fails at a local minimum we apply Gabriel graph and face routing to recover it. Therefore, the packet will be guaranteed to deliver successfully. The proof is the same with that of greedy face routing (GFG) in [12].

Consequently, our on-demand topology control can achieve all properties that [8], [11] achieves, while using much fewer resources.

C. Variations

In this paper, we only use the Yao structure and Gabriel graph as two simple topology control techniques to demonstrate the on-demand concept in ad hoc networks. We definitely can apply more complex or advanced techniques in on-demand fashion to achieve more nice results. For example, we can change Yao structure to Yao and Sink structure to bound the in-degree.



(a) # of large degree nodes in UDG (b) average power stretch factor (c) maximum power stretch factor

(d) failure rate of greedy routing (e) # of face routing phases in GFG (f) spanning ratios of routings

Fig. 2. Simulation results: topology properties and routing performance.

Notice that in our scheme, the execution of Gabriel graph is dependent on routing traffic. Another possible variation of our method is using the *TENT* rule in [32] to identify the nodes where the greedy routing could fail without knowing the routing traffic. Thus we only apply Gabriel graph on those nodes identified as stuck nodes by TENT rule.

V. Simulations

We evaluate the performance of our on-demand topology control protocol by conducting simulations with random networks. In our experiments, we randomly generate a set V of n wireless nodes and its UDG, then test the connectivity of UDG. If it is connected, we construct different localized topologies on UDG, and measure the power efficiency of these topologies. Then we randomly generate routing traffics, and test the greedy routing and greedy face routing on these localized topologies. In the experimental results presented here, we generate n random wireless nodes in a 20×20 square; the parameter k, i.e., the number of cones, is set to 8 when we construct Yao, $ODYao$ and $GYao$; the transmission range is set to 4. We vary the number of nodes in the network n from 50 to 250, where 100 vertex sets are generated for each case to smooth the possible peak effects caused by some exception examples. The average and the maximum are computed over all these 100 vertex sets.

First, we count how many nodes in the networks that have large node degree (larger than $k = 8$). Figure 2 (a) shows the simulation results. When number of nodes is 50, less than 10% nodes have large degree. Thus, when the network is so sparse, by applying our on-demand method, over 90% nodes do not need to apply Yao structures. When the number of nodes increases, the percentage of nodes with large degree also increases. When the network has 250 nodes in the 20×20 square, almost all nodes have more than 8 neighbors.

Figure 2 (b) and (c) illustrate the average and maximum power stretch factors of GG, YG, $GYao$ and $ODYao$. Notice that the power stretch factor of GG is always one. YG and $GYao$ have almost the same average and maximum power stretch factors, while $ODYao$ has smaller power stretch factor than YG and $GYao$. This is much clearer when the network is sparse. For a sparse network, $ODYao$ keeps most of the links for those nodes with small degree (90% for $n = 50$), which will benefit the power efficiency. When the network is very dense ($n > 200$), almost all nodes have large degree and apply the Yao structure, therefore, there is no difference between YG and $ODYao$.

Then, we simulate routing traffics on the network. We randomly select 10% of nodes as the source, and for each source we randomly select 10% of nodes as the destination. The statistics are computed over 20 different node sets.

Figure 2 (d) gives the failure rate of greedy routing on all these topologies. It is obvious that more failures occur with sparser topologies. Notice that YG and $ODYao$ have fewer failures than GG and $GYao$. Therefore, in our on-demand method, we avoid using GG as the underlying topology for GFG routing. Figure 2 (e) shows how many face routing phases the GFG routing executes when GG is the underlying topology. Notice that the average number of face routing phases is less than 0.5 for each routing. Thus, the places where face routing is needed is very limited in the network. Our on-demand method apply GG only at those places.

Finally, we compare our on-demand method with GFG routing (on GG). Since both methods can guarantee the delivery of packets, we only compare the average and maximum spanning ratios of the resulting routes. Here the spanning ratio of a route is defined by $\|\Pi(s,t)\|/\|st\|$, where $\Pi(s,t)$ is the route traversed by the packet using localized routing protocols from source s to destination t. Figure 2 (f) illustrates the results. The travelled distance by our method is slightly better than that by GFG since the structure in our on-demand method is denser than the GG in GFG. The differences between them are not significant, and they are still at the same level. However, remember that our method costs much less resources than GFG does in the topology control phases, which is the main motivation of our on-demand topology control protocol.

VI. Conclusion

Topology control for ad hoc networks has been heavily studied recently and different geometric topologies were proposed to achieve the sparseness and connectivity of the network or to guarantee the package delivery of specific geometric routing methods. In this paper, we introduced the on-demand concept into topology control: applying the specific topology control technique where and when the wireless node needs it. The overhead of topology control at each node is reduced while the nice properties of the topology are kept. Our simulation also confirmed that the performance of routing is also slightly improved by the proposed method. Notice that for topology control techniques we used here (Yao and GG), the overhead we save is mainly the computation cost. The communication cost of overhead messages is almost the same, since the communication cost of building Yao or GG is small comparing with the cost for collecting neighbor information or the cost of "hello" messages. But for some more complex topology control techniques (as in [8]), applying on-demand concept can remarkably save both computation cost and communication cost. In this paper, we only used two simple topology control techniques as the examples to introduce the on-demand concept. We hope it could lead to the full investigation on the practices of on-demand topology control for ad hoc networks. Also we did not consider a certain aspects here, such as throughput and mobility, we leave them as our future work.

References

[1] R. Ramanathan and R. Hain, "Topology control of multihop wireless networks using transmit power adjustment," in *IEEE INFOCOM*, 2000.

[2] M. Grünewald, T. Lukovszki, C. Schindelhauer, and K. Volbert, "Distributed maintenance of resource efficient wireless network topologies," in *8th European Conf. on Parallel Computing*, 2002.

[3] L. Li, J. Y. Halpern, P. Bahl, Y.-M. Wang, and R. Wattenhofer, "Analysis of a cone-based distributed topology control algorithms for wireless multi-hop networks," in *ACM PODC*, 2001.

[4] R. Wattenhofer, L. Li, P. Bahl, and Y.-M. Wang, "Distributed topology control for wireless multihop ad-hoc networks," in *IEEE INFOCOM*, 2001.

[5] X.-Y. Li, P.-J. Wan, and Y. Wang, "Power efficient and sparse spanner for wireless ad hoc networks," in *IEEE ICCCN*, 2001.

[6] X.-Y. Li, P.-J. Wan, Y. Wang, and O. Frieder, "Sparse power efficient topology for wireless networks," in *IEEE HICSS*, 2002.

[7] R. Rajaraman, "Topology control and routing in ad hoc networks: A survey," *SIGACT News*, 33:60–73, 2002.

[8] Y. Wang and X.-Y. Li, "Localized construction of bounded degree planar spanner for wireless networks," in *ACM DIALM-POMC*, 2003.

[9] X.-Y. Li, G. Calinescu, P.-J. Wan, and Y. Wang "Localized Delaunay triangulation with applications in wireless ad hoc networks," *IEEE Trans. on Parallel and Distributed Systems*, 14(10):1035–1047, 2003.

[10] A. C.-C. Yao, "On constructing minimum spanning trees in k-dimensional spaces and related problems," *SIAM J. Computing*, 11:721–736, 1982.

[11] W.-Z. Song, Y. Wang, X.-Y. Li, and O. Frieder, "Localized algorithms for energy efficient topology in wireless ad hoc networks," in *ACM MobiHoc*, 2004.

[12] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia, "Routing with guaranteed delivery in ad hoc wireless networks," *Wireless Networks*, 7(6):609–616, 2001.

[13] B. Karp and H.T. Kung, "GPSR: Greedy perimeter stateless routing for wireless networks," in *ACM MobiCom*, 2000.

[14] F. Kuhn, R. Wattenhofer, and A. Zollinger, "Worst-case optimal and average-case efficient geometric ad-hoc routing," in *ACM MobiHoc*, 2003.

[15] I. Stojmenovic and X. Lin, "Loop-free hybrid single-path/flooding routing algorithms with guaranteed delivery for wireless networks," *IEEE Trans. on Parallel and Distributed Systems*, 12(10):1–10, 2001.

[16] F. Kuhn, R. Wattenhofer, Y. Zhang, and A. Zollinger, "Geometric ad-hoc routing: of theory and practice"," in *ACM PODC*, 2003.

[17] L. Kleinrock and J. Silvester, "Optimum transmission radii for packet radio networks or why six is a magic number," in *IEEE National Telecommunications Conf.*, 1978.

[18] F. Kuhn, R. Wattenhofer, and A. Zollinger, "Asymptotically optimal geometric mobile ad-hoc routing," in *ACM DialM*, 2002.

[19] E. Jennings and C. Okino, "Topology control for efficient information dissemination in ad-hoc networks," in *Int'l Symp. on Performance Evaluation of Computer and Telecommunication Systems*, 2002.

[20] M. Seddigh, J. S. Gonzalez, and I. Stojmenovic, "RNG and internal node based broadcasting algorithms in wireless one-to-one networks," *ACM Mobile Comput. & Commun. Review*, 5(2):37–44, 2001.

[21] N. Nikaein and C. Bonnet, "Topology management for improving routing and network performances in mobile ad hoc networks," *Mob. Netw. Appl.*, 9(6):583–594, 2004.

[22] L. Bao and J. J. Garcia-Luna-Aceves, "Topology management in ad hoc networks," in *ACM MobiHoc*. 2003.

[23] B. Liang and Z. J. Haas, "Virtual backbone generation and maintenance in ad hoc network mobility management," in *IEEE INFOCOM*, 2000.

[24] K. Alzoubi, X.-Y. Li, Y. Wang, P.-J. Wan, and O. Frieder, "Geometric spanners for wireless ad hoc networks," *IEEE Trans. on Parallel and Distributed Processing*, 14(4):408–421, 2003.

[25] C. Schurgers, V. Tsiatsis, and M. B. Srivastava, "STEM: Topology management for energy efficient sensor networks," in *IEEE Aerospace Conf.*, 2002.

[26] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, "Span: an energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks," *Wireless Network*, 8(5):481–494, 2002.

[27] A. Cerpa and D. Estrin, "Ascent: Adaptive self-configuring sensor network topologies," *SIGCOMM Comput. Commun. Rev.*, 32(1):62–62, 2002.

[28] Z. Haas and B. Liang, "Ad-hoc mobility management with uniform quorum systems," *IEEE/ACM Trans. on Netw.*, 7(2):228–240, 1999.

[29] J. Li, J. Jannotti, D. De Couto, D. Karger, R. Morris, "A scalable location service for geographic ad-hoc routing," in *ACM MobiCom*, 2000.

[30] E. Kranakis, H. Singh, and J. Urrutia, "Compass routing on geometric networks," in *Canadian Conf. on Computational Geometry*, 1999.

[31] P. Bose and P. Morin, "Online routing in triangulations," in *Annual Int. Symp. on Algorithms and Computation (ISAAC)*, 1999.

[32] Q. Fang, J. Gao, and L. Guibas, "Locating and bypassing routing holes in sensor networks," in *IEEE INFOCOM*, 2004.

[33] C. Perkins, "Ad hoc on demand distance vector (AODV) routing," in *Internet draft, draft-ietf-manet-aodv-00.txt*, November 1997.

[34] R. Zheng and R. Kravets, "On-demand power management for ad hoc networks," in *IEEE INFOCOM*, 2003.