Mobility-Sensitive Topology Control in Mobile Ad Hoc Networks [*]

Jie Wu and Fei Dai
Department of Computer Science and Engineering
Florida Atlantic University
Boca Raton, FL 33431

Abstract

In most existing localized topology control protocols for mobile ad hoc networks (MANETs), each node selects a few logical neighbors *based on location information, and uses a small transmission range to cover those logical neighbors. Transmission range reduction conserves energy and bandwidth consumption, while still maintaining the network connectivity. However, the majority of these approaches assume a static network without mobility. In a mobile environment, however, network connectivity can be compromised by two types of "bad" location information:* inconsistent information, *which makes a node select too few logical neighbors, and* outdated information, *which makes a node use a too small transmission range. In this paper, we first demonstrate the existing problems, and then propose a mobility-sensitive topology control method that extends many existing mobility-insensitive protocols. Two mechanisms are introduced: consistent local views that avoid inconsistent information, and delay and mobility management that tolerate outdated information. The effectiveness of the proposed approach is confirmed through an extensive simulation study.*

1 Introduction

In mobile ad hoc networks (MANETs), all nodes cooperate to achieve a global task, such as area monitoring and data gathering/communication. To reduce energy consumption and signal interference, it is important to select an appropriate transmission power for each node, also called *topology control*, while still satisfying certain global constraints. Most existing topology control protocols in MANETs use the localized approach to find a small transmission range subject to some global constraints, including connectivity and other reliability and throughput related measures. However, the majority of these approaches assume a static network without mobility. In a typical localized approach, each node collects neighborhood information through periodic, asynchronous "Hello" messages. We refer to neighborhood information collected at each node as *local view* at a particular time.

Consider the example in Figure 1. Assume u's local view is sampled at t while v's local view is done at $t + \Delta$. At t (Figure 1 (a)), mobile node w is 4 and 6 away from nodes u and v, respectively, and at $t + \Delta$ (Figure 1 (b)), w is 6 and 4 away from nodes u and v. The *global view* (Figure 1 (c)) derived by a simple collection of u and v's local views does not correspond to the actual network at any moment. Most existing localized topology control protocols will assign a transmission range of 4 to u and v, resulting in a disconnected network!

In most localized topology control protocols, it is assumed that the network is connected at all times under a (large) normal transmission range. Then each node uses 1-hop information (i.e. location information of all nodes within the transmission range) or partial 1-hop information (e.g., direction or signal strength information of certain nodes within the transmission range) to select a set of *logical neighbors*, which is a subset of the 1-hop neighbor set under the normal transmission range. Then the (short) actual transmission range of each node is set to be the distance to its farthest logical neighbor. The union of logical neighbor sets of all nodes forms a *logical topology*. The logical topology is required to be connected. A topology is *connected* if for any two nodes u and v, v is reachable from u. It is also called *strongly connected* in directed graphs. Such connectivity is ensured under all localized topology control protocols when the network is static. However, since the location information of logical neighbors is collected at different times and nodes move around, there is no guarantee that a logical neighbor is within the actual transmission range at a particular time. In this case, some logical neighbors are no longer reachable while others are still reachable (and reachable neighbors are called *effective neighbors*). The union of effective neighbor sets of all nodes form an *effective topol-*

[*] This work was supported in part by NSF grants CCR 9900646, CCR 0329741, ANI 0073736, and EIA 0130806. Contact address: jie@cse.fau.edu

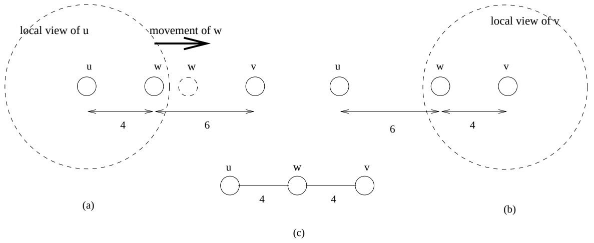


Figure 1. (a) local view of u at t, (b) local view of v at $t + \Delta$, and (c) global view.

ogy.

In the example of Figure 1, the logical topology is connected (assuming u and v are selected by w as logical neighbors) whereas the effective topology is disconnected. In fact, all 1-hop neighbors under the normal transmission range are selected at each node as logic neighbors. It is assumed that logical neighbors of each node are periodically refreshed. The above example shows that the effective topology is disconnected right after v refreshes its view at time $t + \Delta$. Due to inconsistent views of a particular node in terms of its location (u's view of w and v's view of w in the above example), a more serious problem might occur – disconnected logical topology as a result of inconsistent local views as will be shown later. The above discussion leads to two related issues in topology control:

Connected logical topology: Given that the original network is connected (under the normal transmission range), how to ensure that logical topology generated from a topology control protocol is connected.

Connected effective topology: Given that the corresponding logical topology is connected, how to ensure the effective topology is connected.

This paper attempts to address the above issues with a focus on mechanisms used to relax strict conditions used in many literatures on topology control, rather than proposing a new topology control protocol. The advantage of this approach is obvious – our approach can be applied to a large group of protocols by relaxing their assumptions. The proposed approach, called *mobility-sensitive topology control*, extends many existing mobility-insensitive protocols. Specifically, two mechanisms are proposed to address the above issues:

Consistent local views for connected logical topology: Consistent local views are enforced using either synchronous or asynchronous "Hello" messages. If all nodes use the same version of location information to select their logical neighbors, the resultant logical topology is guaranteed to be connected. It does not matter whether the location information corresponds to the actual location of each node.

Delay and mobility management for connected effective topology: Each node uses an extended transmission range that is larger than the actual transmission range in existing protocols. The extended transmission range is computed from the maximal moving speed and "Hello" interval. It creates a buffer zone that tolerates the outdated information caused by node mobility and various delays. When the logical topology is connected, a large buffer zone guarantees a connected effective topology. Even a small buffer zone can improve the connectivity significantly.

The effectiveness of the proposed approach is also confirmed through an extensive simulation study. To our best knowledge, although topology control has been studied extensively in MANETs, our approach is the first attempt ever to systematically extend a large body of localized topology control protocols to mobile environment without changing the original protocols.

2 Related Work

2.1 Topology Control

Most existing topology control algorithms select a less-than-normal transmission range (also called the actual transmission range) while maintaining network connectivity. Centralized algorithms [12, 14, 20] construct optimized solutions based on global information and, therefore, are not suitable in MANETs. Probabilistic algorithms [2, 11, 14] adjust transmission range to maintain an optimal number of neighbors, which balances energy consumption, contention level, and connectivity. However, they do not provide hard guarantees on network connectivity. In a few special cases [13], topology control is integrated into routing protocols to provide a minimal uniform actual transmission range. Most localized topology control algorithms use non-uniform actual transmission ranges computed from 1-hop information (under the normal transmission range). The following is a list of sample localized topology control algorithms that can be enhanced by the mobility management scheme proposed in this paper.

RNG-based algorithms. The relative neighborhood graph (RNG) [17] is a geometrical graph used to remove edges (i.e., reduce the number of neighbors) while maintaining network connectivity. An edge (u, v) is removed if there exists a third node w such that $d(u, v) > d(u, w)$ and $d(u, v) > d(v, w)$, where $d(u, v)$ is the Euclidean distance between u and v. In localized topology control protocols [4], each node determines its logical neighbor set based on location information of 1-hop neighbors. Two nodes u and v are logical neighbors if and only if edge (u, v) exists in RNG. The Gabriel graph [6] is a special case of RNG, where the third node w is restricted to the disk with diameter uv.

Minimum-energy algorithms. Rodoplu and Meng [15] proposed another method of reducing the number of edges while maintaining network connectivity and, in addition, preserving all minimum-energy paths. A minimum-energy path between two nodes u and v is defined as the shortest path between u and v, using transmission power as edge cost. An edge (u, v) can be removed if there exists another node w, such that 2-hop path (u, w, v) consumes less energy than direct transmission. Li and Halpern [8] extended this scheme by using k-hop ($k \geq 2$) paths to remove more edges, and at the same time reduce the computation overhead.

In both protocols [8] and [15], instead of selecting logical neighbors from the normal 1-hop neighbor set, each node collects location information of nodes within a small *search region* to conserve control message overhead. The radius of the search region is iteratively enlarged until logical neighbors in the search region can cover the entire normal 1-hop neighborhood.

Cone-based algorithms. In cone-based topology control (CBTC) [9, 19], the logical neighbor set $\{w_1, w_2, \ldots, w_k\}$ of node v is selected to satisfy the following condition: if a disk centered at v is divided into k cones by lines vw_i ($1 \leq i \leq k$), the angle of the maximal cone is no more than α. It was proved in [9] that, when $\alpha \leq 5\pi/6$, CBTC preserves connectivity, and, when $\alpha \leq 2\pi/3$, the corresponding symmetric subgraph (a subgraph after removing all unidirectional edges) is connected. Several optimizations are also proposed in [9] to further reduce the number of logical neighbors and transmission range. Bahramgiri et al [1] extended CBTC to provide k-connectivity with $\alpha \leq 2\pi/3k$. Similar to the minimum-energy algorithms, CBTC uses dynamic search regions to reduce control overhead. Furthermore, CBTC requires only direction information instead of accurate location information.

MST-based algorithm. Li et al [10] proposed to build a local minimal spanning tree (MST) at each node based on 1-hop location information and select neighbors in MST as logical neighbors. This scheme guarantees connectivity, is easy to implement, and has a constant upper bound (six) on the number of logical neighbors of each node.

2.2 Mobility Management

It was shown in [11] that connectivity in probabilistic topology control algorithms is barely affected by mobility. Blough et al [2] showed that connectivity is preserved with high probability (95%) if every node keeps nine neighbors. In our approach, the logical neighbor set and transmission range are first computed from the neighborhood information of each individual node, and then adjusted to balance the mobility. Compared with the uniform optimal node degree in probabilistic algorithms, our approach requires fewer neighbors on average. Although the node degree in [2] can be further reduced, it is not clear whether the resultant topology is still resilient to mobility after optimization.

Epidemic routing [18] is a scheme to provide eventual delivery in occasionally partitioned networks. It exploits rather than overcomes mobility to achieve a certain degree of connectivity. Basically, when two nodes are moving close to each other within their mutual transmission range, these two nodes will exchange data (based on a probabilistic model) that does not appear in the other node. This scheme extends the Infostation model [16], where one or several neighbors are selected as intermediate nodes to store the data received from the source and to relay the data when they eventually roam into the transmission range of the destination. The difference is that Infostation corresponds to a 2-hop routing process (from source to destination via one intermediate node) whereas epidemic routing is a general multiple-hop routing process.

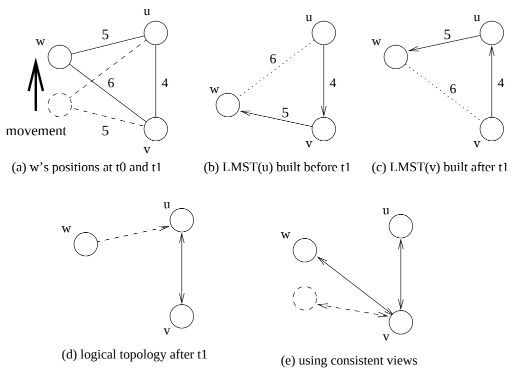
We recently proposed a mobility management scheme [22] to guarantee a connected dominating set (CDS) in a MANET. In order to guarantee link availability in the CDS, only links with relatively small distance values are considered in the formation of CDS. Asynchronous local views of each node are also considered in this scheme. However, local views in CDS formation consist of connection information only. The technique used to overcome view inconsistency in [22] does not apply in topology control, where accurate location information is needed.

All localized topology control algorithms discussed in Section 2.1 depend on accurate location or direction information to guarantee connectivity. In MANETs, neighborhood information is updated via periodical exchanges of control messages. However, as shown in Section 3.1, no matter how small the exchange period, connectivity can always be compromised by inconsistent views at different nodes.

3 Proposed Method

3.1 Preliminaries

For the sake of clarity, we consider only topology control protocols using normal 1-hop information for logical neighbors selection, and leave the integration of dynamic search region to the future work. In those protocols, each node advertises its id and location through periodic “Hello” messages with the normal transmission range. We assume a fixed “Hello” interval; that is, the period between two “Hello” messages from the same node is a constant Δ. However, due to the inaccuracy of local clocks in individual nodes, “Hello” messages from different nodes may be asynchronous.



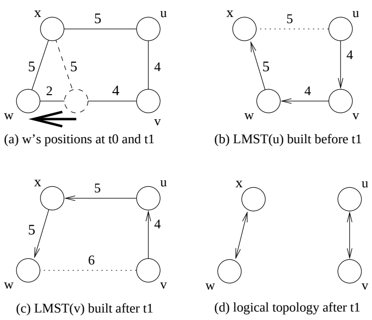
(a) w's positions at t0 and t1 (b) LMST(u) built before t1 (c) LMST(v) built after t1

(d) logical topology after t1 (e) using consistent views

Figure 2. Partition in a 3-node network.

The latest "Hello" messages received from 1-hop neighbors form the local view of each node, which includes id's and locations of most nodes within its normal transmission range. Due to the node mobility, some new 1-hop neighbors may come in undetected. Because topology control technique is usually applied to dense networks, we assume the network is always connected with the normal transmission range after removing those undetected links. The local view of a node also includes the id and location of itself. Two local views are *inconsistent* if they contain conflicting information about the same node; otherwise, they are consistent. As shown in Figure 2 (a), when node w moves upwards and sends two "Hello" messages from different locations, node u's local view based on the former "Hello" message from w (Figure 2 (b)) and node v's local view based on the latter "Hello" message (Figure 2 (c)) are inconsistent.

In most existing protocols, each node selects its logical neighbor set periodically based on its local view. To simplify the discussion, we assume the interval between two selections on the same node is also Δ, and selections on different nodes are asynchronous. If we add a logical link among every two logical neighbors, the set of all logical links from a logical topology. Usually, a topology control protocol guarantees a connected logical topology if all local views are consistent and the network is dense enough. However, the logical topology may be disconnected as a result of inconsistent local views. We use the MST-based algorithm [10] as an example to illustrate the problem of disconnected logical topology. In this approach, each node v builds an MST spanning its 1-hop neighbors. Then only neighbors that are directly connected to v in MST are selected as v's logical neighbors. Each node v calculates its local MST, $LMST(v)$, periodically based on its local view. In static networks, connectivity of the logical topology is guaranteed. In MANETs, however, different nodes may update their local MSTs based on inconsistent local views, which may yield a disconnected logical topology!



(a) w's positions at t0 and t1 (b) LMST(u) built before t1

(c) LMST(v) built after t1 (d) logical topology after t1

Figure 3. Partition in a 4-node network.

Suppose node w in Figure 2 (a) moves upward, and advertises its locations twice at time t_0 and $t_1 = t_0 + \Delta$, respectively. If node u updates $LMST(u)$ at $t_1 - \delta$ and node v updates $LMST(v)$ at $t_1 + \delta$, neither u nor v considers w as a logical neighbor (Figures 2 (b) and (c)). Based on u's local view, v should select w as its logical neighbor, because w is close to v. In v's local view, however, w is close to u and should be u's logical neighbor. The corresponding logical topology is disconnected (Figure 2 (d)), no matter how small δ is. Here it is assumed that node w selects node u as its logical neighbor based on its local view. Link (u, v) is bidirectional (the solid line) and link (w, v) is unidirectional (the dash line). Note that connectivity can be maintained in this case, if we convert every unidirectional link in LE_t into a bidirectional link. However, this does not solve the problem for the general case. As shown in Figure 3, node w moves from right to left and advertises its locations at time t_0 and $t_1 = t_0 + \Delta$. If $LMST(u)$ is built at $t_1 - \delta$ and $LMST(v)$ at $t_1 + \delta$, neither link (u, x) nor (v, w) exists in $LE_{t_1+\delta}$. Similarly, if $LMST(x)$ is built at $t_1 - \delta$ and $LMST(w)$ at $t_1 + \delta$, links (x, u) and (w, v) are not in $LE_{t_1+\delta}$. That is, the logical topology is disconnected.

Once a set of logical neighbors is determined, each node adjusts its actual transmission range to the distance to the farthest logical neighbor. All nodes within the actual transmission range are called physical neighbors. In some topology control protocols, physical neighbors are *functional*; that is, a message from a physical neighbors is always passed to the upper level protocol. In other protocols, the physical neighbors are not functional; messages from a non-logical physical neighbor is simply discarded. If a neighbor is both a logical neighbor and a physical neighbor, it is an effective neighbor. An effective link exists between every two effective neighbors. All effective links form the effective topology. We assume all data packets are transmitted along effective links. Therefore, a connected network after topology control means a connected effective topology. Since the effective topology is a subgraph of the logical topology, a connected network also implies a connected

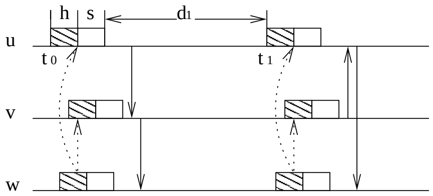


Figure 4. Build consistent local views

logical topology.

Our mobility-sensitive topology control scheme preserves connectivity in two steps. First the connectivity of the logical topology is guaranteed by building and using consistent local views. Two methods are proposed. One method uses synchronized "Hello" messages to enhance connectivity during each "Hello" interval. The other method uses asynchronous and timestamped "Hello" messages to achieve connectivity in the routing of each packet. Unlike Infostation variations and epidemic routing, our method does not cause significant increase of end-to-end delay or memory consumption. The second step is to ensure the connectivity of the effective topology. Each node uses a larger-than-actual transmission range (called an *extended transmission range*) to create a "buffer zone" that preserves all logical links in the effective topology. The size of the buffer depends on the maximal moving speed and "Hello" interval.

3.2 Consistent views under synchronous "Hello" intervals

In existing topology control algorithms, each node receives "Hello" messages from its 1-hop neighbors, and updates its local view upon the arrival of every "Hello" message. If all nodes have synchronized clocks, this scheme actually works. In Figure 2, if both nodes u and v selects their logical neighbors at t_0, they will agree that v is closer to w; at t_1, they will agree that u is closer. Here we omit the propagation delay and assume that a "Hello" message is received by all neighbors at the same time. However, it is impossible to have totally synchronized clocks in a MANET without centralized control. If u makes its decision slightly earlier than v, and w's "Hello" message arrives after u's decision and before v's decision, then the two nodes have inconsistent views. This inconsistency cannot be avoided no matter how small the asynchrony is.

One solution is to build local views only once at the beginning of each "Hello" interval. As shown in Figure 4 (a), each "Hello" interval is divided into three time periods $\Delta = h + s + d$. Because of asynchronous clocks, different nodes may start their "Hello" intervals at different times. That is, some nodes have "faster" clocks than other nodes. However, we assume the difference between two clocks is bounded by s. In the construction of consistent views, each node sends its "Hello" message (represented by dotted lines) during period h, waits for a period s, and conducts normal activities (e.g., sending data packets, represented as solid lines) in period d. As the h period of the "slowest" node ends before the s period of the "fastest" node, every node receives all "Hello" messages before the end of its s period. Local views built in the end of s are consistent. It is safe to route data messages in period d_1 based on these local views.

When this method is applied to MST-based algorithm [10] to address the problem of disconnected logical topology, each node calculates its local MST based on "Hello" messages of the same version, i.e., the one transmitted during the current "Hello" interval. For example, since both $LMST(u)$ and $LMST(v)$ in Figure 2 (e) are calculated based on the same location of w, either node u or v will consider node w as its logical neighbor. When all nodes use the location information advertised at time t_0 in their local views, $G_{t_0+\delta}$ contains a bidirectional link (vw) and is connected.

3.3 Consistent views under asynchronous "Hello" intervals

In the last subsection, it is assumed that the maximal difference among local clocks, s, is predictable and $s \leq \Delta$. In a totally asynchronous system, $s = \Delta$ and the above simple approach cannot be applied. Note that even if $s < \Delta$ at a particular network, delays accumulate unless some clock synchronization protocol is applied. Although various solutions [19] exist to adjust clock values, frequent clock synchronization is costly. When maintaining (partially) synchronous "Hello" interval becomes too expensive or impossible, we propose using timestamped asynchronous "Hello" messages to enforce application specific consistent local views.

The basic idea is to maintain a sequence number i_v at each node v, and attache the sequence number to each "Hello" messages from this node. The sequence number is initialized to zero and increased by one before sending each "Hello" message. The sequence number serves as a timestamp. Consistent local views are obtained from "Hello" messages with the same timestamps. This can be done by carrying a timestamp in each data packet (including control packets from a higher level protocol). The timestamp is chosen by the originator of the data packet, and all nodes relaying this packet must determine their logical neighbors based on information of the same version (i.e., with the same timestamp). In this scheme, each node keeps several local views, each local view corresponding to a recently used timestamp. Similarly, several logic topologies co-exist in the same network. Each logic topology corresponds to a timestamp and is connected by itself. The logic time (i.e., the timestamp of the latest local view) of the originator of the data packet is used as an selector. It indicates in which logical topology this data packet is travelling in.

In Figure 2, suppose the first "Hello" message from node w has timestamp 0, and the second one has timestamp 1. When the above method is applied, two parallel logic topologies exist. The logical topology corresponding to timestamp 0 includes two bidirectional links (u, v) and (v, w). The logic topology corresponding to timestamp 1 includes (u, v) and (u, w). When a data packet p is sent from u to w, the source node u selects a recent timestamp and forward p on the corresponding logical topology. If p has timestamp 0, it is first forwarded to v. Based on v's local view with timestamp 0, w is a logical neighbor of v, and p is forwarded along the logical link (v, w). If p has timestamp 1, it is sent to w directly via logical link (u, w). In both cases, p arrives safely at its destination.

Note that a local view cannot be used until it includes "Hello" messages with the same timestamp from all 1-hop neighbors. Since in MANETs, the 1-hop neighbor set changes over time, it is unknown whether a node has collected all "Hello" messages or not. Assume the maximal time interval between two "Hello" messages with the same timestamp, called synchronization delay, is $\Delta^{'}$. For the sake of safety, after a node receives a "Hello" message with timestamp i, it must wait a time period of $\Delta^{'}$ before the formation of its local view with timestamp i. Compared with the maximal clock difference s in the previous scheme, $\Delta^{'}$ is allowed to be larger than Δ, which means a lower level of node synchronization and lower maintenance overhead.

We use a simple scheme to achieve a certain degree of node synchronization. Periodically, a random node u increases its current sequence number i_u by a large integer number inc, such the new sequence number $i_u^{'} = i_u + inc$ is larger than other sequence numbers in the same network with a high probability. Then u broadcasts $i_u^{'}$ to all other nodes through flooding or an efficient broadcasting process [21]. Once a node v receives $i_u^{'}$ and its current sequence number $i_v \leq i_u^{'}$, it sets $i_v \leq i_u^{'}$; otherwise, it initiates a synchronization process using $i_v^{'} = i_v + inc$. When the process completes, the synchronization delay is the "Hello" interval plus broadcast delay. Normally broadcast delay is very small compared to the "Hello" interval and can be neglected. Note that this synchronization process needs not be frequent in MANETs with relatively accurate clocks. If the clocks are accurate, only one flooding is sufficient.

3.4 Delay and mobility management

Although under the above model each node obtains a consistent local view, views of different nodes are taken from different physical times. In other words, the node information shows node positions at different times. In order to apply existing topology control protocols without having to re-design them, we use the notion of *buffer zone*, where two circles with radii r and $r + l$ are used (see Figure 5). r corresponds to the actual transmission range determined by a topology control protocol. $r + l$ corresponds to the extended transmission range used, where $l = d \times t$ is defined as a buffered range depending on the moving speed $t/2$ of mobile nodes and the maximum time delay d. In the proactive approach, a local view taken at time t may depend on the "Hello" message sent at $t - \Delta^{'}$ and may be used until $t + \Delta^{'}$. Therefore, d is $2\Delta^{'}$. In the reactive approach, all "Hello" messages are sent at the beginning of the current "Hello" interval. Therefore, d is bounded by Δ plus the propagation delay (including the short backoff delays at intermediate nodes) of the flooding process. Several optimization methods can be used to have a good estimate of l at each node. For example, the "timeliness" of each "Hello" message can be measured by latency between the (physical) time it is received and the time it is used in a local decision.

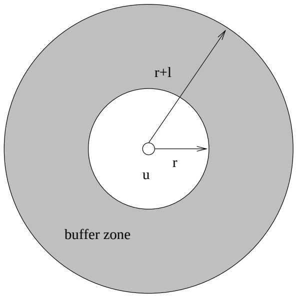


Figure 5. The notion of buffer zone with different transmission ranges.

When this approach is applied to address the problem of disconnected effective topology (such as the one in Figure 1), the extended transmission range is properly set based on the "Hello" interval and node moving pattern and its speed. In the example of Figure 2 (e), the transmission power of each node is enlarged to create a buffer zone that guarantees the existence of a physical link even if the distance between v and w has been changed due to the movement.

4 Simulation

4.1 Implementation

We evaluate topology control algorithms under $ns2$ [5] and its CMU wireless and mobility extension [7]. 100 nodes are randomly placed in a $900 \times 900m^2$ area. The normal transmission range is $250m$, which yields an average node degree of 18 without topology control. The mobility pattern is generated based on the random waypoint model [3] with zero pause time and the average moving speed varying from 1 to $160m/s$. In order to isolate the effects of mobility from other factors such as collision, all simulations use an ideal MAC layer without collision and contention. Each simulation lasts $100s$ and is repeated 20 times. Each result is associated with the 95% confidence interval.

In our implementations of baseline algorithms, each node advertises its location via asynchronous "Hello" messages. Although MAC layer collision is not simulated, the "Hello" interval of each node is randomly selected from $1 \pm 0.25s$ to avoid the collision in the real world. "Hello" messages are transmitted with the normal transmission power. Each node selects its logical neighbors based on the complete 1-hop information. Three baseline algorithms are implemented: RNG-based algorithm, MST-based algorithm, and minimum-energy (SPT-based) algorithm. The minimum-energy algorithm builds local SPTs based on the energy function $E = d^{\alpha}$, where E is the required transmission power, and d is the length of a link. We use two choices of α: (1) $\alpha = 2$ as in the free space model, and (2) $\alpha = 4$ as in the two-way ground reflection model. In all algorithms, each node updates its logical neighbor set whenever it sends a "Hello" message, and adjusts its transmission power to the minimal power that reaches the farthest logical neighbor. The logical neighbor set is attached in the header of every outgoing packet. The receiver will drop the packet if it is not in the sender's logical neighbor set. Since the logical neighbor set is usually very small, single round convergence is achieved without excessive overhead. Unidirectional links are neither removed nor converted into bidirectional edges. We have not simulated the cone-based algorithm, as we are still in search of an implementation of CBTC with all its optimizations in [9] and single round convergency.

Algorithm	Trans. range (m)	Node degree
MST	65.09 ± 1.61	2.09 ± 0.01
RNG	78.95 ± 2.65	2.44 ± 0.03
SPT ($\alpha = 4$)	75.04 ± 2.00	2.51 ± 0.05
SPT ($\alpha = 2$)	100.10 ± 2.75	3.46 ± 0.10

Table 1. Average transmission range and node degree of baseline algorithms.

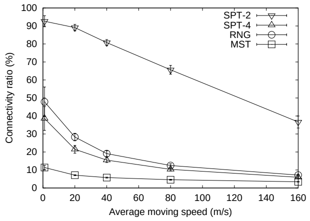


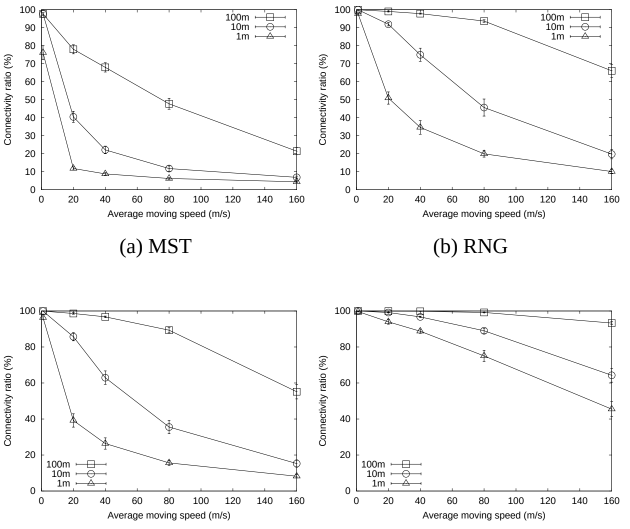
Figure 6. Connectivity ratio of baseline algorithms.

Two connectivity models can be defined in MANETs: *strict connectivity* and *weak connectivity*. A MANET is strictly connected if its snapshot (i.e., the effective topology at a particular time) taken at every moment is connected. However, in a MANET with mobile nodes, it is difficult to capture network topology under a snapshot (although we can do so in simulations via assuming an omniscient "god"). Weak connectivity is more appropriate which is application dependant. In this model, the connectivity is defined in terms of capability of completing a connectivity-related task, such as global flooding, measured in terms of the percentage of nodes that receive the message. Note that a weakly connected network may not be strictly connected under a particular snapshot (or even any snapshot). In Figure 1, a broadcast initiated at u at time t and forwarded by w at time $t + \Delta$ ensures a full coverage. However, the network is not connected under any snapshot. Note that weak connectivity is exploited only in special routing schemes such as Infostation variations [16] and epidemic routing [18], where end-to-end delay is traded for eventual delivery. In a flooding that completes in a small ($< 0.01s$) time period, weak connectivity is a rather accurate approximation of the strict connectivity.

Against the baseline algorithms, we evaluate three mechanisms that enhance the connectivity in MANETs.

Buffer zone: If the logical topology is connected, then using a buffer zone can tolerate the inaccurate location information caused by mobility. In the worst case, the age of the location information is twice the maximal "Hello" interval, i.e., $2.5s$, and the relative speed between two neighbors is two times the maximal moving speed and four times the average moving speed. Therefore, to tolerate an average moving speed of $10m/s$, the width of the buffer zone shall be $100m$. However, as shown in [22], the same level of mobility can be tolerated by a much thinner buffer zone with high probability.

View synchronization: The connectivity of the logical topology cannot be guaranteed based on inconsistent local views. We use a simplified mechanism to provide almost consistent views on-the-fly. Whenever a node sends a packet, it updates its logical neighbor set based on the current view, i.e., the location information advertised in latest "Hello" messages from 1-hop neighbors. If the packet travels fast enough, nodes visited by the same packet will probably have consistent local views. Note that each node must use its previous location advertised in the last "Hello" message, instead of its current location, in its calculation.



(a) MST

(b) RNG

(c) SPT ($\alpha = 4$)

(d) SPT ($\alpha = 2$)

Figure 7. Connectivity ratio with different buffer zone widths.

Physical neighbor: The network connectivity can be enhanced by allowing non-logical neighbors to relay packets instead of dropping them. This mechanism works better with a large buffer zone, where more physical neighbors form multiple paths that tolerate higher mobility levels.

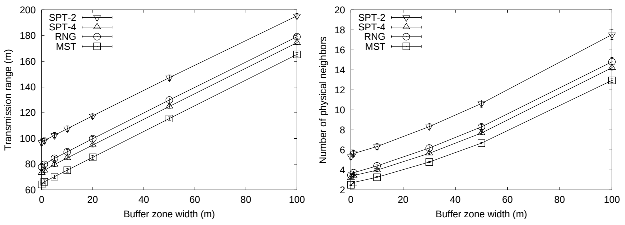
The baseline algorithms and different enhancements are compared in terms of the following metrics.

Connectivity ratio: i.e., the ratio of connected node pairs to the total number of node pairs. We compute the connectivity ratio as the average delivery ratio of broadcast packets originated from random sources. The broadcast frequency is 10 packets per second and 1000 packets per simulation.

Transmission range: The average transmission range serves as an indicator of the average transmission power. We avoid using transmission power directly, because the diversity of the energy models may cause unnecessary ambiguity. The transmission range is also a good indicator of the channel reuse ratio.

Node degree: One common goal of topology control algorithms is to reduce the network density, which can be represented by the average node degree. Here we consider only the number of logical neighbors, except in the third enhancement, where physical neighbors also count.

4.2 Results



(a)

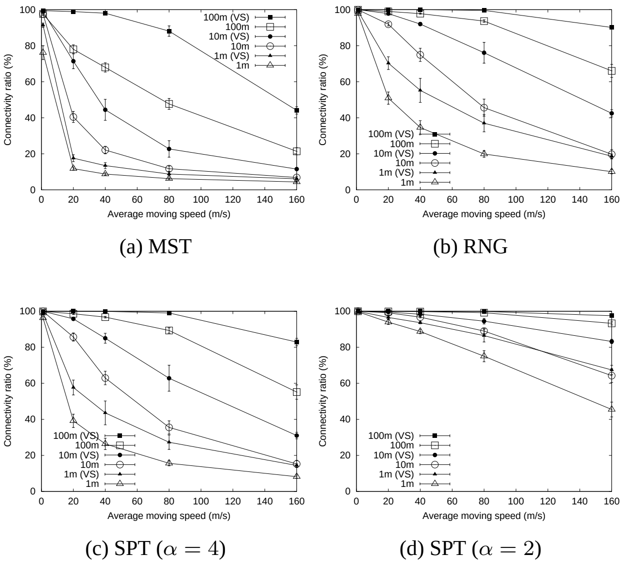
(b)

Figure 8. Average transmission range (a) and number of physical neighbors (b) versus buffer zone width.

Baseline algorithms. Table 1 shows the effectiveness of each baseline algorithm in reducing the transmission range and number of logical neighbors. The MST-based algorithm (MST) has the smallest transmission range and node degree. The average node degree of 2.09 implies that the logical topology is close to tree, which has the average node degree of $2(n-1)/n = 1.98$. A tree is the most efficient way to maintain a connected logical topology. However, it is also the most vulnerable. The SPT-based algorithm with $\alpha = 2$ (SPT-2) has the largest transmission range ($100m$) and node degree (3.46). Compared with the normal transmission range ($250m$) and original node degree (18), SPT-2 still saves significantly in energy and bandwidth consumption. The RNG-based algorithm (RNG) and SPT-based algorithm with $\alpha = 4$ (SPT-4) have similar transmission range and node degrees, which lie between MST and SPT-2. RNG has slightly larger transmission range and smaller node degree than SPT-2, suggesting that RNG have more physical neighbors than SPT-2.

Figure 6 shows the connectivity ratio of baseline algorithms in MANETs. The mobility level varies from very low ($1m/s$) to moderate (20-40m/s) and extremely high (80-160m/s). Our objective is to find methods that maintain high connectivity ratio ($\geq 90\%$) under low and moderate mobility. Extremely high mobility is unlikely in MANETs and is used to benchmark the resilience of each algorithm to mobility. As shown in Figure 6, all baseline algorithms are vulnerable to mobility. The best algorithm, SPT-2, can tolerate only very slow mobility. Other algorithms have only 50% (RNG), 40% (SPT-4) and 10% (MST) connectivity ratio under very low mobility. MST is the most vulnerable, because in a tree-like topology, the probability of partition is very high. In most scenarios, a single link failure is enough to disconnect the entire network.

Buffer zone. We first handle link failures caused by logical neighbors moving out of the actual transmission range. The goal is to find the minimal buffer zone width that tolerates moderate mobility, that is, maintains 90% delivery ratio when the average moving speed is below or equal to



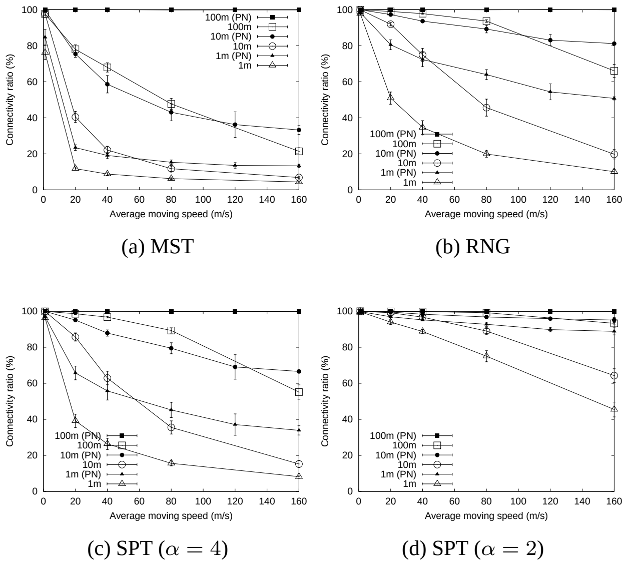
(a) MST

(b) RNG

(c) SPT ($\alpha = 4$)

(d) SPT ($\alpha = 2$)

Figure 9. Connectivity ratio with and without view synchronization.



(a) MST

(b) RNG

(c) SPT ($\alpha = 4$)

(d) SPT ($\alpha = 2$)

Figure 10. Connectivity ratio before and after using physical neighbors.

$40m/s$. Our finding is that using buffer zone alone does not eliminate the problem in most algorithms. As shown in Figure 7, MST tolerates $1m/s$ mobility with a $10m$ buffer zone. However, it cannot tolerate $20m/s$ or higher mobility. Both RNG and SPT-4 can tolerate moderate mobility with a $100m$ buffer zone, but cannot do so with a $10m$ buffer zone. The only exception is SPT-2, which tolerates moderate mobility with a $10m$ buffer zone.

Algorithms using a buffer zone have the same average node degree in their logical topologies. They do, however, have larger transmission ranges. Figure 8 (a) shows that, when a $100m$ buffer zone is used to tolerate moderate mobility, the average transmission ranges of RNG and SPT-4 are above $160m$. On the other hand, the same job is done in SPT-2 with a $10m$ buffer zone and $120m$ average transmission range. The suggestion is that a certain level of redundancy may be necessary for saving energy in MANETs.

View synchronization. We consider the partitioned logical topology caused by inconsistent local views. When the simple view synchronization mechanism is used together with buffer zones, all algorithms show solid improvement in connectivity ratios. Figure 9 compares different connectivity ratios achieved with and without view synchronization. With view synchronization (VS), MST can tolerate moderate mobility with a $100m$ buffer zone. RNG can do so with a $10m$ buffer zone. SPT-4 can tolerate $20m/s$ mobility with a $10m$ buffer zone, but still needs a $100m$ buffer zone to tolerate $40m/s$ mobility. SPT-2 can tolerate $40m/s$ mobility with a $1m$ buffer zone, $80m/s$ mobility with a $10m$ buffer zone, and $160m/s$ with a $100m$ buffer zone.

Algorithms using view synchronization have the same average transmission range and node degree as algorithms not using this mechanism. RNG is our favorite in this case: it tolerates moderate mobility with $10m$ buffer zone, which corresponds to an average transmission range of $88m$, as shown in Figure 8. Meanwhile, the $1m$ buffer zone width used in SPT-2 corresponds to an average transmission range of $98m$.

Physical neighbor. The connectivity ratio can be improved via relatively high redundancy, i.e., a large neighbor set. An effective method that increases redundancy is to treat all physical neighbors as logical neighbors. That is, the topology control protocol will pass to the upper layer every packet it receives, instead of dropping packets from non-logical neighbors. Asynchronous views are now tolerable, because the resultant logical neighbor sets are only references in computing a small transmission range that maintains connectivity with a high probability. The idea is similar to that in the K-Neigh [2] protocol. The difference is that, in K-Neigh, a uniform optimal number of neighbors is used to decide the transmission power at each node.

Figure 10 shows the effect of using physical neighbors (PN). The result is similar to the effect of view synchronization. SPT-2 can tolerate moderate mobility with a $1m$ buffer zone, RNG and SPT-4 can with a $10m$ buffer zone, and MST with a $100m$ buffer zone. Note that, when $100m$ buffer zones are used, every algorithm has a perfect connectivity ratio (100%) under extremely high mobility

($160m/s$). Actually, MST achieves 93% connectivity ratio with a $30m$ buffer zone in our simulation. Figure 8 (b) illustrates the increased redundancy. The average node degree that tolerates moderate mobility is 4.7 for MST ($30m$), 4.2 for RNG ($10m$), 3.8 for SPT-4 ($10ms$), and 5.4 for SPT-2 ($1m$). These results are smaller than the optimal node degree in K-Neigh.

Simulation results can be summarized as follows: (1) All baseline algorithms suffer from low connectivity ratio in MANETs. (2) The low connectivity ratio is caused by both disconnected physical link and disconnected logical topology due to inconsistent views. (3) When a simple view synchronization mechanism is used, RNG and SPT can tolerate moderate mobility ($\leq 40m/s$) with small buffer zones ($\leq 10m$). (4) If all physical neighbors are allowed to forward packets, all algorithms can tolerate moderate mobility with average node degrees from 3.8 to 5.4.

5 Conclusion

We have proposed a mobility-sensitive topology control method that extends many mobility-insensitive protocols. This method is based on two mechanisms: local view synchronization based on synchronous or asynchronous "Hello" messages, and buffer zone created by slightly increasing the actual transmission range. These two mechanisms ensure the connectivity of both logical topology and effective topology, two notions proposed in this paper for topology control in dynamic networks. Extensive simulation confirmed the effectiveness of these two mechanisms in maintaining network connectivity under slow and moderate mobility. Our future work includes exploring other means of handling mobility and verifying the effectiveness of these mechanisms measured by other metrics such as network bi-connectivity.

References

[1] M. Bahramgiri, M. Hajiaghayi, and V. S. Mirrokni. Fault-tolerant and 3-dimensional distributed topology control algorithms in wireless multi-hop networks. In *Proc. of ICCCN*, pages 392–397, 2002.

[2] D. Blough, M. Leoncini, G. Resta, and P. Santi. The K-Neigh protocol for symmetric topology control in ad hoc networks. In *Proc. of MobiHoc*, pages 141–152, June 2003.

[3] T. Camp, J. Boleng, and V. Davies. A survey of mobility models for ad hoc network research. *Wireless Communication & Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications*, 2(5):483–502, 2002.

[4] J. Cartigny, D. Simplot, and I. Stojmenović. Localized minimum-energy broadcasting in ad-hoc networks. In *Proc. of INFOCOM*, volume 3, pages 2210 –2217, Mar./Apr. 2003.

[5] K. Fall and K. Varadhan. The *ns* manual. The VINT Project, UCB, LBL, USC/ISI and Xerox PARC, http://www.isi.edu/nsnam/ns/doc/, Apr. 2002.

[6] K. R. Gabriel and R. R. Sokal. A new statistical approach to geographic variation analysis. *Systematic Zoology*, 18:259–278, 1969.

[7] D. B. Johnson, J. Broch, Y.-C. Hu, J. Jetcheva, and D. A. Maltz. The CMU Monarch projects wireless and mobility extensions to ns. In *Proc. of 42nd Internet Engineering Task Force*, Aug. 1998.

[8] L. Li and J. Y. Halpern. Minimum energy mobile wireless networks revisited. In *Proc. of ICC*, pages 278–283, June 2001.

[9] L. Li, J. Y. Halpern, V. Bahl, Y. M. Wang, and R. Wattenhofer. Analysis of a cone-based distributed topology control algorithm for wireless multi-hop networks. In *Proc. of PODC*, pages 264–273, Aug. 2001.

[10] N. Li, J. C. Hou, and L. Sha. Design and analysis of an mst-based topology control algorithm. In *Proc. of INFOCOM*, volume 3, pages 1702–1712, Mar./Apr. 2003.

[11] J. Liu and B. Li. MobileGrid: Capacity-aware topology control in mobile ad hoc networks. In *Proc. of ICCCN*, pages 570–574, Oct. 2002.

[12] E. L. Lloyd, R. Liu, M. V. Marathe, R. Ramanathan, and S. S. Ravi. Algorithmic aspects of topology control problems for ad hoc networks. In *Proc. of MobiHoc*, pages 123–134, June 2002.

[13] S. Narayanaswamy, V. Kawadia, R. S. Sreenivas, and P. R. Kumar. Power control in ad-hoc networks: Theory, architecture, algorithm and implementation of the COMPOW protocol. In *Proc. of European Wireless*, pages 156–162, Feb. 2002.

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

[15] V. Rodoplu and T. H. Meng. Minimum energy mobile wireless networks. *IEEE Journal of Selected Areas in Communications*, 17(8):1333–1344, Aug. 1999.

[16] T. Small and Z. J. Haas. The shared wireless infostation model: a new ad hoc networking paradigm (or where there is a whale, there is a way). In *Proc. of MobiHoc*, pages 233–244, June 2003.

[17] G. Toussaint. The relative neighborhood graph of finite planar set. *Pattern Recognition*, 12(4):261–268, 1980.

[18] A. Vahdat and D. Becker. Epidemic routing for partially connected ad hoc networks. Technical Report CS-200006, Duke University, Apr. 2000.

[19] R. Wattenhofer, L. Li, V. Bahl, and Y. M. Wang. Distributed topology control for power efficient operation in multihop wireless ad hoc networks. In *Proc. INFOCOM*, pages 1388–1397, Apr. 2001.

[20] J. E. Wieselthier, G. D. Nguyen, and A. Ephremides. On the construction of energy-efficient broadcast and multicast trees in wireless networks. In *Proc. of INFOCOM*, pages 585–594, Mar. 2000.

[21] J. Wu and F. Dai. A generic distributed broadcast scheme in ad hoc wireless networks. In *Proc. of ICDCS 2003*, pages 460–468, May 2003.

[22] J. Wu and F. Dai. Mobility management and its applications in efficient broadcasting in mobile ad hoc networks. In *Proc. of INFOCOM*, 2004. Accepted.