

# 22

## Mobility Control and Its Applications in Mobile Ad Hoc Networks

---

22.1	Introduction.....	22-499
22.2	Background and Motivation .....	22-500
22.3	Existing Work .....	22-501
	CDS Formation • Topology Control • Mobility Control	
22.4	Link Availability.....	22-504
22.5	View Consistency .....	22-507
	View Consistency Issue • Consistent Local View • Conservative Local View • Smart Neighbor Discovery	
22.6	Simulation Results .....	22-512
22.7	Conclusion .....	22-514
	Acknowledgement .....	22-514
	References .....	22-514

Jie Wu and Fei Dai

### 22.1 Introduction

---

Most existing localized protocols in mobile ad hoc networks (MANETs) use local information for decentralized decision among nodes to achieve certain global objectives. These objectives include determining a small connected dominating set (CDS) for virtual backbone and topology control by adjusting transmission ranges of nodes. Because of asynchronous sampling of local information at each node, delays at different stages of protocol handshake, and movement of mobile nodes, local view captured at different nodes may be *outdated* and *inconsistent*, and may not reflect the actual network situation. The former may incur “broken” links that fail to keep the given global constraint such as global domination and connectivity; the latter may cause “bad” decisions, which in turn will ultimately cause the failure of the constraint. This chapter reviews several *mobility control* techniques that handle outdated and inconsistent local views. These techniques are illustrated using several well-known protocols in data communication and topology control.

The remainder of this chapter is organized as follows. Section 22.2 introduces the topology control problem and our motivation. Section 22.3 reviews related work in CDS formation, topology control, and

22-499

mobility control. Section 22.4 discusses the broken link issue and introduces a solution using large actual transmission ranges to create a buffer zone. Section 22.5 illustrates bad decisions caused by inconsistent local views, and introduces three view consistency mechanisms. Simulation results showing the effectiveness of the proposed topology control methods are presented in Section 22.6. Section 22.7 concludes this chapter and discusses some open issues.

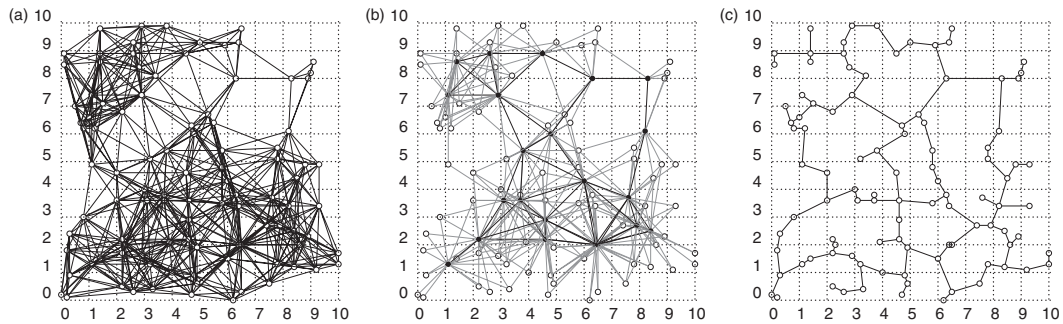
## 22.2 Background and Motivation

In MANETs, all nodes cooperate to achieve a global task, such as data gathering, communication, and area monitoring. A MANET is usually modeled as a *unit disk graph* where two nodes are connected if and only if their geographical distance is smaller than a given transmission range (as shown in Figure 22.1(a)). To design protocols that are simple and quick to converge, many protocols in MANETs rely on localized algorithms. A deterministic distributed algorithm is *localized* if its execution on each node relies on local information (e.g., 2-hop topology information or 1-hop location information) only, without any sequential propagation of partial computation results; otherwise, it is *nonlocalized*. Collectively, nodes running the localized algorithm achieve some desirable global objectives. Two popular applications of the localized algorithm are (1) determining a CDS for efficient routing [10,26,35,49] and (2) selecting an appropriate transmission range of each node for topology control [22,29,42,43].

A CDS is a subset such that each node in the system is either in the set or the neighbor of a node in the set. In addition, the induced graph of the CDS is connected. Nodes in the CDS are called *dominators* and those outside *dominatees*. As dominators form a backbone of the MANET, any dominatee can switch to the sleep mode for energy saving without causing network partition. Other applications of CDS include area coverage and efficient broadcasting. Finding a minimum CDS is known as NP-complete. Most practical approaches in MANETs use localized algorithms to find a small CDS. In a typical localized CDS protocol, each node  $v$  uses local information to determine its status, dominator, or dominatee, based on its 2-hop topology information, including  $v$ 's neighbors and neighbors' neighbors. Figure 22.1(b) shows a CDS constructed via a localized algorithm [10]. In this diagram the connections between dominatees are not shown.

In MANETs, in order to reduce energy consumption and signal interference, it is important to select an appropriate transmission power for each node, while still satisfying certain global constraints such as connectivity. This task, called topology control, is usually implemented via localized algorithms. In such an algorithm, each node first selects only a few *logical neighbors* among its *physical neighbors* (i.e., nodes within its normal transmission range), and then sets its actual transmission range to the distance of the farthest logical neighbor. Localized topology control algorithms usually rely on 1-hop location information of the current node  $v$ , including ids and  $(x, y)$  coordinations of  $v$ 's physical neighbors. Some algorithms require less information where the distance or angle of arrival (AoA) information is used instead of location information. Figure 22.1(c) shows the result of a localized topology control algorithm [22], where both the average number of neighbors and transmission range are reduced significantly, while the network is still connected.

Compared with their nonlocalized counterparts, localized algorithms are lightweight, fast to converge, and resilient to node movement. However, without a mobility control mechanism, global domination and connectivity may still be compromised by node movement. In most existing localized algorithms, each node in a MANET emits a periodic “Hello” message to advertise its presence and its position (if needed) at a fixed interval  $\Delta$ . “Hello” intervals at different nodes are asynchronous to reduce message collision. Each node extracts neighborhood information from received “Hello” messages to construct a local view of its vicinity (e.g., 2-hop topology or 1-hop location information). In a MANET with mobile nodes, the limited sample frequency, asynchronous “Hello” intervals, and delays at different stages of protocol handshake will cause a mismatch between the *virtual network* constructed from the collection of local views sampled at different nodes and the *actual network*. This mismatch will cause the link availability issue, where a neighbor in a virtual network is no longer a neighbor in the actual network, because the



**FIGURE 22.1** Virtual networks constructed via localized algorithms. The original MANET has 100 nodes and a transmission range of 2.5. (a) The original network. (b) Connected dominating set. (c) Topology control.

virtual network is constructed from outdated information. Therefore, special mechanisms are needed to address the following issue:

- *Delay and mobility management:* How protocols deal with imprecise neighborhood information caused by node mobility and various delays introduced at different stages of protocol handshake.

One solution in Reference 47, as will be discussed later in detail, uses two transmission ranges to address the link availability issue. First, a transmission range  $r$  is determined based on the selected protocol. This transmission range is either the same as the “Hello” message range  $r'$  as in the CDS protocol or shorter than  $r'$  as in the topology control protocol. The actual transmission uses a long transmission range set to  $r + l$ . The difference,  $l$ , between these two ranges is based on the update frequency and the speed of node movement.

The mismatch between the virtual- and actual-network will cause a more serious problem: inconsistent local views. Inconsistent local views may cause “bad” decisions that fail to keep the global constraint such as global domination and connectivity. Again, special mechanisms are needed to address the following issue:

- *Node synchronization and consistent local view:* How each node knows when to sample its local view. How each node collects and uses the local information in a consistent way.

We will examine three different approaches that address the consistency issue: enforcing consistent views, making conservative decisions, and smart neighbor discovery. The first approach was initially proposed in Reference 48 to construct consistent 1-hop location information for topology control. This approach can also be extended to support the construction of 2-hop information. The second approach was originally proposed in Reference 47 for CDS formation. The same principle can be used in topology control. The third approach was first proposed in Reference 11 and extended in Reference 47 for mobility control in CDS-based efficient broadcasting.

The main objective of this chapter is to expose to the reader the challenging issue related to mobility control. Through discussion on the effect of mobile nodes on several important protocols, we present some problems, provide possible solutions, and discuss several open issues.

## 22.3 Existing Work

Before introducing our mobility control schemes, we shall first have a brief look at existing CDS formation and topology control protocols. Here we focus on localized algorithms based on accurate neighborhood information, which have no or limited topology control. This section also examines some preliminary work on the mobility control issue.

### 22.3.1 CDS Formation

Both *centralized* and *decentralized* algorithms have been proposed to form a CDS in MANETs. Centralized algorithms [12,41] usually yield a smaller CDS than decentralized algorithms, but their application is limited due to the high maintenance cost. Decentralized algorithms can be further divided into cluster-based algorithms and localized algorithms. Cluster-based algorithms [2,5,14,23] have a constant approximation ratio in unit disk graphs and relatively slow convergence. In localized algorithms [1,9,26,28,32,33,46,49], the status of each node depends on its 2- or 3-hop information only. A localized algorithm takes constant steps to converge, produces a small CDS on average, but has no constant approximation ratio.

Wu and Li [49] devised the marking process and two pruning rules (called Rule 1 and Rule 2) that select a few nodes to form a CDS in an undirected graph. Wu [46] extended this scheme for CDS formation in directed graphs. This scheme is further extended by Dai and Wu [10] via a more efficient pruning rule. Rieck et al. [28] extended Wu and Li's marking process to form a  $d$ -hop CDS, and suggested to use Rule 1 only to preserve shortest paths among any pair of source and destination nodes. Since Rule 1 alone is not so efficient in reducing the number of dominators, Rieck's scheme usually yields a larger CDS.

Chen et al. [9] presented an approach similar to the marking process (called Span) to select a few *coordinators* for energy efficient routing. Ideally coordinators form a CDS such that other nodes can switch to the energy saving mode without compromising the routing capability of the network. A node  $v$  becomes a coordinator if it has two neighbors that are not directly connected, indirectly connected via one intermediate coordinator, or indirectly connected via two intermediate coordinators. Before a node changes its status from noncoordinator to coordinator, it waits for a backoff delay which is computed from its energy level and 2-hop neighborhood topology. The backoff delay can be viewed as a priority value, such that nodes with shorter backoff delay have a higher chance of becoming coordinators. Span cannot ensure a CDS when two coordinators simultaneously change back to noncoordinators.

Qayyum et al. [26] put forward an efficient broadcast scheme called multipoint relaying (MPR). In MPR, each node designates a few 1-hop neighbors (MPRs) to cover its 2-hop neighbor set. All MPRs form a CDS, which was used as a backbone in the optimized link state routing (OLSR) protocol. Adjih et al. [1] suggested a method to reduce the backbone size. Based on the new rule, not all MPRs are dominators; a MPR becomes a dominator only when it happens to be designated by a node with the lowest id in its 1-hop neighbors. Meanwhile, to maintain the connectivity, a any node with the lowest id among its neighbors becomes a dominator automatically.

Sinha et al. [32] proposed the *core*-based scheme. Each node designates a node with the highest priority among its neighbors (including itself) as its dominator (called core). The set of cores form a dominating set, but not necessarily a CDS. To get a CDS, each core designates several gateway nodes to connect neighboring cores within 3 hops [33]. This scheme was extended by Amis et al. [3] to form a  $d$ -hop dominating set.

### 22.3.2 Topology Control

Many topology control schemes have been proposed to use a small actual transmission range to conserve energy and bandwidth consumption while maintaining network connectivity. Centralized algorithms [25,27,44] construct optimized solutions based on global information and, therefore, are not suitable in MANETs. Probabilistic algorithms [7,24,27] 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. Most localized topology control algorithms use nonuniform actual transmission ranges computed from 1-hop location information. Here the 1-hop neighborhood refers to physical neighbors under the normal transmission range.

The relative neighborhood graph (RNG) [38] 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 [8,31], each node determines its logical neighbor set based on positions 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 [16] is a special case of RNG, where the third node  $w$  is restricted to the disk with diameter  $uv$ .

Rodoplu and Meng [29] gave 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 the 2-hop path  $(u, w, v)$  consumes less energy than direct transmission. Li and Halpern [20] 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 [20] and [29], 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; that is, each position outside of the search region can be reached via a  $k$ -hop path through a selected logical neighbor, and the  $k$ -hop path is more energy-efficient than direct transmission. If the search region is the entire 1-hop neighborhood, the Li and Halpern's algorithm is equivalent to constructing a local shortest path tree (SPT) where only neighbors of the root in SPT become logical neighbors.

In cone-based topology control (CBTC) [21,43], the logical neighbor set  $\{w_1, w_2, \dots, w_k\}$  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  $\alpha$ . It was proved in Reference 21 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 Reference 21 to further reduce the number of logical neighbors and transmission range. Bahramgiri et al. [4] 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.

A similar but different scheme is based on Yao graph [42], where a disk centered at node  $v$  is evenly divided into  $k$  cones, and a logical neighbor is selected from each cone. It was proved that Yao graph is connected with  $k \geq 6$ . Yao graph with  $k = 6$  can be viewed as a special case of CBTC with  $\alpha = 2\pi/3$ , but not vice versa.

Li et al. [22] 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 (6) on the number of logical neighbors of each node.

### 22.3.3 Mobility Control

We use the name mobility control to denote mechanisms in localized algorithms that compensate inaccurate neighborhood information and maintain the global constraints. Previous work on topology control is very limited. This section addresses those preliminary attempts to handle the mobility issue.

Locally formed CDS are frequently used in efficient broadcast protocols, where dominators become forward nodes to rely the broadcast packet. In their simulation study, Williams and Camp [45] showed that localized broadcast protocols suffer low delivery ratio in mobile and congested networks. In a later simulation study [11], Dai and Wu concluded that the major cause of the low delivery ratio is contention and mobility. In order to reduce contention, fewer forward nodes should be used to reduce redundancy. However, low redundancy makes a broadcast protocol more vulnerable to packet losses caused by mobility. Therefore, an adaptive approach is desirable that has low redundancy under low mobility, and increases redundancy only when the mobility level rises. Although several probabilistic broadcast protocols [18,39] have been proposed by trading between efficiency with delivery ratio, it is difficult to establish a direct connection between forwarding probability and node mobility.

It was shown in Reference 24 that connectivity in probabilistic topology control algorithms is barely affected by mobility. Blough et al. [7] showed that connectivity is preserved with high probability (95%) if every node keeps nine logical 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 logical neighbors on average. Although the node degree in Reference 7 can be further reduced, it is not clear whether the resultant topology is still resilient to mobility after optimization.

Very little work has been done in maintaining an accurate neighbor set in ad hoc networks. One exception is [19], where a *stable zone* and a *caution zone* of each node has been defined based on a node's position, speed, and direction information obtained from GPS. Specifically, stable zone is the area in which a mobile node can maintain a relatively stable link with its neighbor nodes since they are located close to each other. Caution zone is the area in which a node can maintain an unstable link with its neighbor nodes since they are relatively far from each other. The drawback of this approach is that it is GPS-based, which comes with a cost, as will be discussed in the next section. In addition, there is no rigorous analysis on the impact of mobility on the selection of these two zones.

Several papers [6] addressed the issue of how long two nodes will remain in close enough proximity for a link between them to remain active. Several routing protocols, associativity-based routing (ABR) [37] and signal stability-based adaptive routing (SSA) [13], have been proposed that select *stable links* to construct a route. In Reference 36, GPS information is used to estimate the expiration time of the link between two adjacent hosts. Recently, several studies have been done on the effect of mobility on routing path [30]. However, no broadcast protocol uses the notion of stable link to evaluate the stability of neighbor set in order to better decide the forwarding status of each node.

In general, the capacity of ad hoc networks is constrained by the mutual interference of concurrent transmissions between nodes. The mobility of nodes adds another dimension of complexity in the mutual interference. Epidemic routing [40] 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 [15,17,34], 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.

## 22.4 Link Availability

The broken link problem caused by outdated local views is relatively easy to understand and deal with. In this section, we discuss the link availability issue and our solution to this problem. The view consistency issue is more complex and will be discussed in the next section.

In MANETs, because of asynchronous "Hello" messages and various protocol handshake delays, neighborhood information and position used in decision making may be outdated. For example, a previously sampled neighbor can move out of transmission range during the actual transmission. In order to apply existing protocols without having to redesign them, the notion of *buffer zone* is used in Reference 47, where two circles with radii  $r$  and  $r + l$  are used (see Figure 22.2),  $r$  corresponds to the transmission range determined by a selected protocol, whereas  $r + l$  corresponds to the actual transmission range used and  $l = d \times 2t$  is defined as a buffered range depending on the moving speed  $t$  of mobile nodes and the maximum delay  $d$ . To simplify the discussion, both "Hello" intervals and moving patterns/speeds are homogeneous, and hence,  $l$  is uniform for each node.

The above requirement of buffered range guarantees link availability in the worst case situation. However, probabilistic study in Reference 47 reveals that the worst case rarely happens. In MANETs

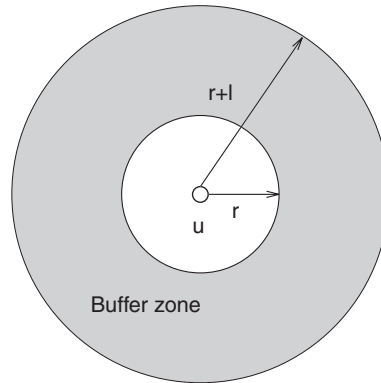


FIGURE 22.2 The notion of buffer zone with different transmission ranges.

with very high moving speed ( $t$ ), it is either impossible or too expensive to use such a large  $l$ . Both probabilistic analysis and simulation results in Reference 47 show that link availability is preserved in most cases with a buffered range much smaller than  $d \times 2t$ . There is a wide range of potential trade-offs between efficiency and connectivity.

Specifically, suppose  $r'$  is the normal “Hello” message range. A typical CDS protocol works as follows:

1. Select  $r = r'$  for neighborhood information exchange.
2. Apply the selected localized CDS protocol to determine the status of each node.
3. Use  $r + l$  for each dominator in the actual transmission.

Step 2 of the above process varies from protocol to protocol. Here we use Wu and Li’s marking process and Rules 1 and 2 [49] to illustrate:

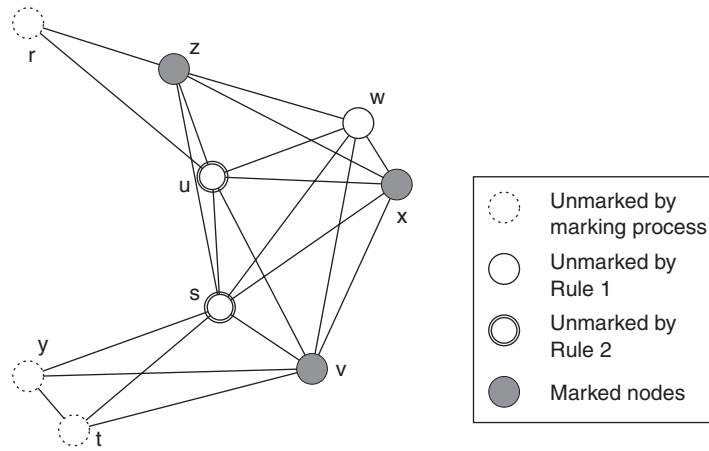
At each node  $u$ :

- Marking Process:  $u$  is marked true (i.e., becomes a dominator) if there are two unconnected neighbors.
- Rule 1:  $u$  is unmarked (i.e., becomes a dominatee) if its neighbor set is covered by another node with a higher id.
- Rule 2:  $u$  is unmarked if its neighbor set is covered jointly by two connected nodes with higher id’s.

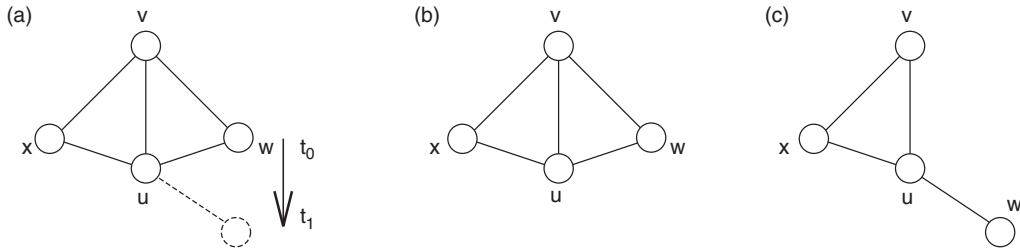
In Rules 1 and 2, we say  $u$ ’s neighbor set,  $N(u)$ , is *covered* by one or two *covering* nodes, if every node  $w$  in  $N(u)$  is either a covering node or a neighbor of a covering node. Using the above marking process with Rules 1 and 2, marked nodes form a CDS. Figure 22.3 shows a sample ad hoc network with 9 nodes. Node  $r$  is unmarked by the marking process because its neighbors  $u$  and  $z$  are directly connected. Node  $w$  is unmarked by Rule 1 because its neighbor set is covered by node  $x$ . Here node id  $x$  is higher than  $w$  according to the alphabetical order. Node  $u$  is unmarked by Rule 2 because its neighbor set is covered by two connected nodes  $x$  and  $z$ . Clearly, the marked nodes  $v$ ,  $x$ , and  $z$  form a CDS of the sample network.

Originally, Rules 1 and 2 use only marked nodes as covering nodes, and involve overhead in communicating dominating set status. Stojmenovic et al. [35] showed that unmarked nodes can also be covering nodes, and there is no need to exchange dominating set status. Dai and Wu’s [10] proposed a generalized rule (called Rule  $k$ ) to construct a smaller CDS. Based on the generalized rule,  $u$  is unmarked if its neighbor set is covered by several connected nodes with higher id’s. The number of the covering nodes is allowed to be more than two. The generalized rule can unmark more nodes, and is easier to check than Rules 1 and 2.

When the network is static or local views are consistent (say, all nodes see only solid line) in Figure 22.4, both nodes  $u$  and  $v$  are marked after the marking process.  $u$  will be unmarked using Rule 1.



**FIGURE 22.3** Wu and Li's CDS algorithm. Black nodes are marked (i.e., in the CDS). Nodes unmarked by the marking process have dotted lines; those unmarked by Rule 1 have single solid lines; those by Rule 2 have double solid lines.



**FIGURE 22.4** Both nodes  $u$  and  $v$  are unmarked due to inconsistent local views sampled at nodes  $u$  and  $v$ . (a)  $w$ 's position at  $t_0$  and  $t_1$ . (b)  $u$ 's decision as an unmarked node at  $t_1$ . (c)  $v$ 's decision as unmarked at  $t_1$ .

A typical topology control protocol works as follows:

1. Select  $r'$  to collect neighborhood information.
2. Apply a selected localized topology control protocol to select  $r(u)$ ,  $r(u) \leq r'$ , for node  $u$  to cover its farthest logical neighbor.
3. Use  $r(u) + l$  for the actual transmission.

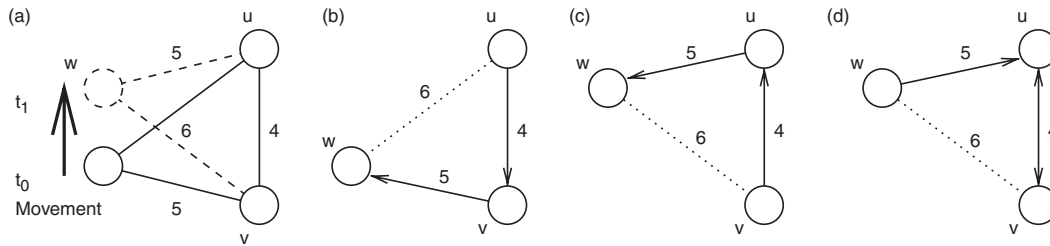
We use Li, Hou, and Sha's topology control algorithm based on local MST [22] to illustrate Step 2 of the above process.

At each node  $u$ :

1. Build a local MST using Prim's algorithm based on 1-hop location information. The resultant MST covers all 1-hop neighbors of  $u$ .
2. Select neighbors in MST as logical neighbors of  $u$ .
3. Set the transmission range of  $u$  to its distance to the farthest logical neighbor.

When the network is static or local views are consistent in Figure 22.5(a), all nodes build an identical MST that includes two links  $(u, v)$  and  $(w, v)$ . Node  $u$  has one logical neighbor  $v$  and sets its range to 4. Node  $w$  has one logical neighbor  $v$  and sets its range to 5. Node  $v$  has two logical neighbors  $u$  and  $w$  and sets its range to 5 to reach the farthest node  $w$ .





**FIGURE 22.5** Node  $w$  becomes unreachable from nodes  $u$  and  $v$  due to inconsistent local views sampled at nodes  $u$  and  $v$ . (a)  $w$ 's position at  $t_0$  and  $t_1$ . (b)  $LMST(u)$  built at time  $t_0$ . (c)  $LMST(v)$  built at time  $t_1$ . (d) Network topology at time  $t_1$ .

When the network contains mobile nodes, such as node  $w$  in Figure 22.5, the transmission range of each node is increased to maintain the link availability. For example, if it is known that the maximum relative movement between two nodes during one “Hello” interval is  $l = 2$ , then the actual transmission range of nodes  $u$ ,  $v$ , and  $w$  are adjusted to 8, 7, and 8, respectively. Therefore, link  $(v, w)$  is still available even if node  $w$  moves upward and the distance between  $v$  and  $w$  becomes 6. It is also observed in Reference 47 that the buffered range  $l = 2$  is conservative and not always necessary. The probability is high that all links can be maintained with a smaller  $l$ .

## 22.5 View Consistency

In this section, we first explain what are inconsistent local views and consequences of using inconsistent views in localized algorithms. Then we discuss three possible solutions to overcome the inconsistent view problem. In the first solution, a certain synchronization mechanism is used to ensure consistent views during a given time period. In the second solution, each node makes conservative solutions based on recent history information. The third solution relies on an aggressive fault detection mechanism, which precautions neighbors about unreliable links. Due the early warning, most “bad” decisions of neighbors can be avoided.

### 22.5.1 View Consistency Issue

We use two localized algorithms as examples to demonstrate how inconsistent local views cause “bad” decisions in MANETs. Both algorithms, Wu and Li’s marking process [49] for CDS construction, and Li, Hou, and Sha’s topology control algorithm based on local MST [22], have been introduced in the last section.

In the CDS construction example (as shown in Figure 22.4), we assume that node  $w$  moves southward. Link  $(v, w)$  exists at time  $t_0$  and is broken at time  $t_1$ . We also assume  $t_0$  and  $t_1$  belong to two intervals. Since link  $(v, w)$  is two hops away from node  $u$ , when node  $u$  decides its status, it uses the outdated information (lagging by one interval) that link  $(v, w)$  still exists. The local view  $u$  is shown in Figure 22.4(b). Based on Rule 1, node  $u$  is unmarked because its neighbor set is covered by node  $v$ . However, when node  $v$  decides its status, it has the fresh information that link  $(w, v)$  is broken since it is adjacent to the link (as shown in Figure 22.4(c)). Based on the marking process, the only two neighbors of  $v$ ,  $x$ , and  $u$ , are connected, so node  $v$  is also marked false. As a consequence, none of the nodes in the network are marked.

In the topology control example (as shown in Figure 22.5), assume node  $u$ 's view reflects the topology at  $t_0$  (as shown in Figure 22.5(b)) whereas node  $v$ 's view corresponds to the topology at  $t_1$  (as shown in Figure 22.5(c)). This happens when the recent “Hello” message from  $w$  is sent at  $t$ , where  $t_0 < t < t_1$ . In this case,  $u$  has only one logical neighbor  $v$ , and  $v$  has only one logical neighbor  $u$ . Based on the protocol, a link is selected only if both end nodes select each other. As a result, only one link  $(u, v)$  exists after the topology control (as shown in Figure 22.5(d)). A network partition occurs.

In the earlier examples, individual nodes make “bad” decisions based on inconsistent local views. Two views are inconsistent if their common parts do not match. In the CDS example, link  $(v, w)$  exists in node  $u$ 's view but not in node  $v$ 's view. In the topology control example,  $w$  is closer to  $v$  in  $u$ 's view but closer to  $u$  in  $v$ 's view.

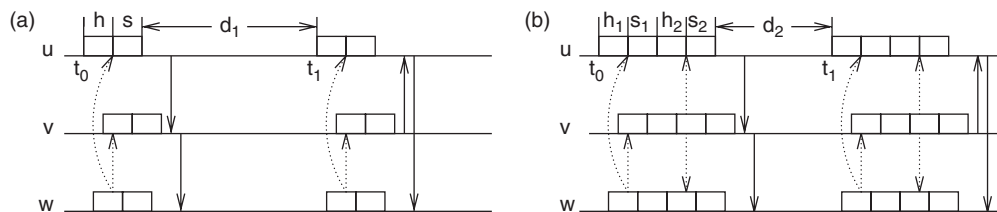
## 22.5.2 Consistent Local View

One solution to the above problem is to enforce consistent views such that the same information “version” is used by all nodes in making their decisions. First we consider the problem of collecting consistent 1-hop information via loosely synchronous “Hello” messages. Originally, 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 the topology control example, if both nodes  $u$  and  $v$  make their decisions 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.

The traditional solution for this problem is to build local views only once at the beginning of each “Hello” interval. As shown in Figure 22.6(a), each “Hello” interval is divided into three time periods  $\Delta = h + s + d_1$ . 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 during period  $h$ , waits for a period  $s$ , and conducts normal activities (e.g., sending data packets) in period  $d_1$ . 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 packets in period  $d_1$  based on these local views.

This scheme can be extended to build 2-hop information. As shown in Figure 22.6(b), each “Hello” interval is divided into five periods  $\Delta = h_1 + s_1 + h_2 + s_2 + d_2$ . Normally  $h_1 = h_2$  and  $s_1 = s_2$ . Here we assume the clock difference is bounded by both  $s_1$  and  $s_2$ . Each node first advertises its 0-hop information (i.e., its id and location) in period  $h_1$ , builds 1-hop information at the end of period  $s_1$ , and then advertises the newly constructed 1-hop information in period  $h_2$ . At the end of period  $s_2$ , every node constructs its consistent local view, which is ready for use in period  $d_2$ . The drawback of this scheme is that two “Hello” messages are sent during each interval  $\Delta$ , and the effective communication period  $d_2$  is further reduced.

Then we consider another situation with asynchronous “Hello” messages. The traditional solution relies on the assumption 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



**FIGURE 22.6** Build consistent local views at the beginning of each “Hello” interval. Dotted lines represent “Hello” messages. Solid lines represent data packets. (a) Consistent 1-hop information. (b) Consistent 2-hop information.

applied. Although various solutions 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 attach  $i_v$  to each “Hello” message from  $v$ . 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. 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 coexist in the same network. Each logic topology corresponds to a timestamp and is connected. The logic time (i.e., the timestamp of the latest local view) of the originator of the data packet is used as a selector. It indicates in which logical topology this data packet is traveling.

In Figure 22.5, 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 forwards  $p$  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.

The above approach can tolerate a larger “time skew” among different local views and, therefore, involves less synchronization overhead. Let  $\Delta'$  be the maximal time difference between any two “Hello” messages with the same timestamp. This scheme still works when  $\Delta' > \Delta$ . However, the buffer zone should be adjusted accordingly to compensate for the “older” local views. If the local clock speed is different in each node,  $\Delta'$  may accumulate during a long time period, and the buffer zone may be very large. A certain degree of synchronization is still required to maintain a relatively small  $\Delta'$ . For example, randomly selected nodes can initiate periodical broadcast processes that help each node adjust its timestamp. This synchronization process need not be frequent when each node has a relatively accurate clock. If all local clocks are accurate, only one broadcasting is sufficient.

### 22.5.3 Conservative Local View

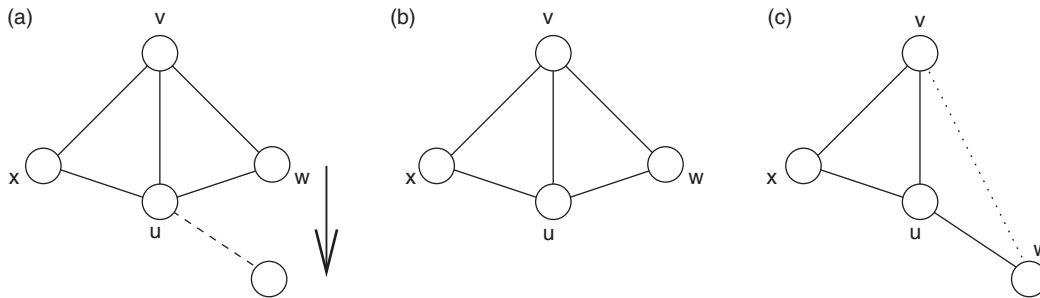
In the previous section, we discussed two methods of enforcing consistent local views. Both methods require a certain degree of internode synchronization, which introduces extra overhead. When maintaining consistent local views becomes too expensive or impossible, another approach called *conservative local view* [48] can be applied, which makes conservative decisions based on inconsistent views. No synchronization is necessary. A conservative decision is one that maintains the global property with the penalty of lower efficiency. That means selecting more logical neighbors in a topology control algorithm and marks more nodes as dominators in a CDS formation process. We use Wu and Li's marking process as an example to illustrate the conservative approach.

In Wu and Li's marking process, a node  $v$  may be incorrectly unmarked if (1)  $v$  no longer views a node  $w$  as its neighbor, and (2) another node  $u$  still views  $w$  as  $v$ 's neighbor and unmarks itself based on this view. As the broken link  $(v, w)$  is first detected by  $v$  and then propagated to  $u$  via periodical “Hello” messages, local views of nodes  $u$  and  $v$  are inconsistent for a short period. During that period,  $u$  and  $v$  may be unmarked simultaneously, and the CDS is temporarily compromised. In order to prevent conditions (1) and (2) from happening together, each node must use a conservative local view, instead of its most recent local view, to make conservative decisions. In this case, the conservative local view  $\text{View}_c(v)$  of node  $v$  is constructed from  $k$  most recent local view  $\text{View}_1(v), \text{View}_2(v), \dots, \text{View}_k(v)$  based on the following rule: a link  $(u, w)$  exists in  $\text{View}_c(v)$  if and only if (1)  $(u, w)$  exists in the most recent local views  $\text{View}_1(v)$ ,

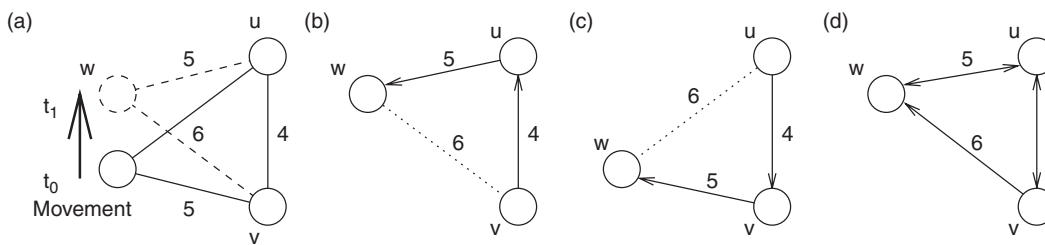
or (2)  $u = v$  and  $(u, v)$  exists in at least one recent local view  $View_i(v)$  for  $1 \leq i \leq k$ . That is, a broken link is preserved longer in the conservative views of its two end nodes than in those of all other nodes.

As shown in Figure 22.7, after node  $v$  detects a broken link  $(v, w)$ , it will keep a virtual link corresponding to the broken link in its local view for a short time period. Based on this conservative view (as shown in Figure 22.7(c)),  $v$  is still a dominator. Note that the virtual link  $(v, w)$  is still available during this time period, if  $v$  uses a large actual transmission range to create a buffer zone, as discussed in Section 22.2. The virtual link stays in  $v$ 's view until all other nodes have removed this link from their views. When 2-hop information is used, link  $(v, w)$  exists in local views of  $v$ 's 1-hop neighbors and  $w$ 's 1-hop neighbors, which will remove link  $(v, w)$  from their local views after receiving a "Hello" message from  $v$  or  $w$ . Node  $v$  will send its next "Hello" message within a "Hello" interval  $(\Delta)$ . Node  $w$  may detect the broken link and send its "Hello" message later than  $v$ , but the difference is bounded by  $\Delta$ . Therefore, it is safe to remove the virtual link  $(v, w)$  for  $v$ 's local view after  $2\Delta$ .

This approach can also be applied to other localized CDS and topology control algorithms. However, the conservative decisions are different from algorithm to algorithm, and the construction of conservative views depends on the specific localized algorithm. For example, in Li, Hou, and Sha's local MST algorithm, a conservative view of node  $v$  can be defined as follows: given  $k$  most recent local views  $View_1(v), View_2(v), \dots, View_k(v)$ , which contain distance values  $d_i(u, w)$  ( $1 \leq i \leq k$ ) between any two nodes  $u$  and  $w$  within  $v$ 's transmission range (including  $v$ ), their distance in the conservative view is (1)  $\max_i d_i(u, w)$ , if  $u \neq v$  and  $w \neq v$ , and (2)  $\min_i d_i(u, w)$  otherwise. That is, the virtual distance between  $v$  and a neighbor  $w$  in its conservative view may be smaller than the actual distance, and the virtual distance between two neighbors may be larger than the actual distance. As shown in Figure 22.8, when conservative local views are used, both nodes  $u$  and  $v$  select  $w$  as a logical neighbor, and the network after topology control is connected.



**FIGURE 22.7** Based on the conservative view, node  $v$  is still marked for a little while after it detected the broken link  $(v, w)$ . The dotted line represents a virtual link in  $v$ 's view. (a)  $w$ 's position at  $t_0$  and  $t_1$ . (b)  $u$ 's decision as an unmarked node at  $t_0$ . (c)  $v$ 's decision as marked at  $t_1$ .



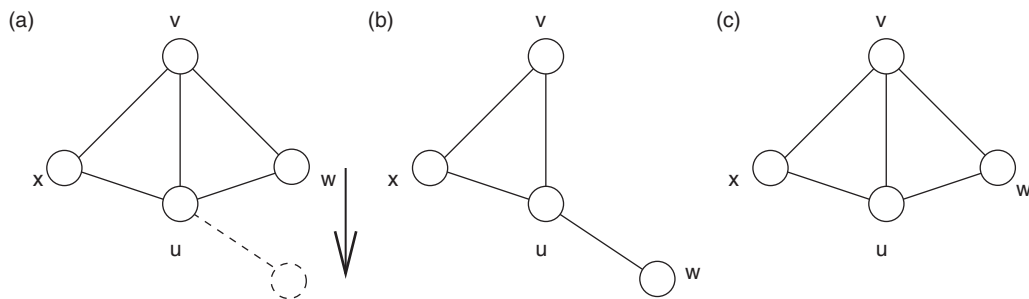
**FIGURE 22.8** Network connectivity is preserved using conservative views. (a)  $w$ 's position at  $t_0$  and  $t_1$ . (b)  $u$ 's conservative view. (c)  $v$ 's conservative view. (d) Network topology.

### 22.5.4 Smart Neighbor Discovery

A simple mobility control mechanism was proposed in Reference 11 to make conservative local views by maintaining two, instead of one, neighbor sets at each node. Most localized CDS formation and CDS-based broadcast algorithms use a neighbor discovery mechanism that monitors periodical “Hello” messages from neighbors. When a node receives a “Hello” message, the sender’s id is added to its current neighbor set. This id stays in the neighbor set as long as “Hello” messages from the same sender are received regularly. Otherwise, a link failure is detected and the corresponding node id is removed from the current neighbor set. In order to reduce the number of false alerts caused by collisions of “Hello” messages, the link failure detection mechanism is usually designed to tolerate two consecutive losses of “Hello” messages. A node  $v$  will consider another node  $w$  as a neighbor as long as it has received at least one “Hello” message from  $w$  during the last three “Hello” intervals. Therefore, the 1-hop neighbor information is usually outdated, and the protocol adapts poorly to node movement. On the other hand, an aggressive failure detector is not safer. For example, if node  $v$  in Figure 22.9(c) removes node  $w$  from its neighbor set by mistake, it will be unmarked based on the false information, leaving  $w$  uncovered.

In the dual neighbor set solution [11], two neighbor sets, one private and another public, are maintained using different link failure detection standards. After node  $v$  misses the first “Hello” message from node  $w$ ,  $w$  is removed from node  $v$ ’s public neighbor set. The public neighbor set is promptly advertised to other nodes so that a graceful handover can be arranged in advance. Meanwhile,  $w$  remains in  $v$ ’s private neighbor set until losses of three consecutive “Hello” messages. The status of  $v$  is determined based on its private neighbor set, which remains a dominator during the handover period. For example, when link  $(v, w)$  fails in Figure 22.9(a) due to the movement of  $w$ , it is first removed from the public neighbor set of  $v$ , which is advertised to node  $u$ . Therefore, node  $u$  is marked shortly after the node movement, as shown in Figure 22.9(b). On the other hand, as node  $w$  remains in  $v$ ’s private neighbor set, its local view is as in Figure 22.9(c), based on which  $v$  is still marked for a smooth handover. Simulation results in Reference 11 show that this enhancement is very effective in improving reliability, where the size of CDS increases automatically to compensate for increased mobility level.

The above “smart” neighbor discovery scheme is extended in Reference 47, where signal strength is used for early detection of unstable links. Here the “Hello” message is transmitted with the actual transmission range  $r + l$ . When a node  $v$  receives a “Hello” message, its distance to the sender  $w$  is estimated based on the incoming signal strength. The id  $w$  is always put into  $v$ ’s private neighbor set, but will be removed from  $v$ ’s public neighbor set if the distance between  $v$  and  $w$  is larger than  $r$ . This link failure detection scheme is more aggressive and eliminates the possibility of simultaneous withdrawal in Figure 22.4. In this case, node  $u$  is warned about the failure of link  $(v, w)$  and becomes a dominator before the failure actually happens. In the same time, node  $v$  remains a dominator until the link fails. Therefore, node  $w$  is always covered by either node  $u$  or node  $v$  or both. This scheme is also useful in sparse networks, where using a large “Hello” transmission range is necessary to avoid network partition.



**FIGURE 22.9** Because node  $w$  is removed from node  $v$ ’s public neighbor set while it is still in  $v$ ’s private neighbor set, both nodes  $u$  and  $v$  are marked shortly after  $t_1$ . (a)  $w$ ’s position at  $t_0$  and  $t_1$ . (b)  $u$ ’s position as marked at  $t_1$ . (c)  $v$ ’s position as marked at  $t_1$ .

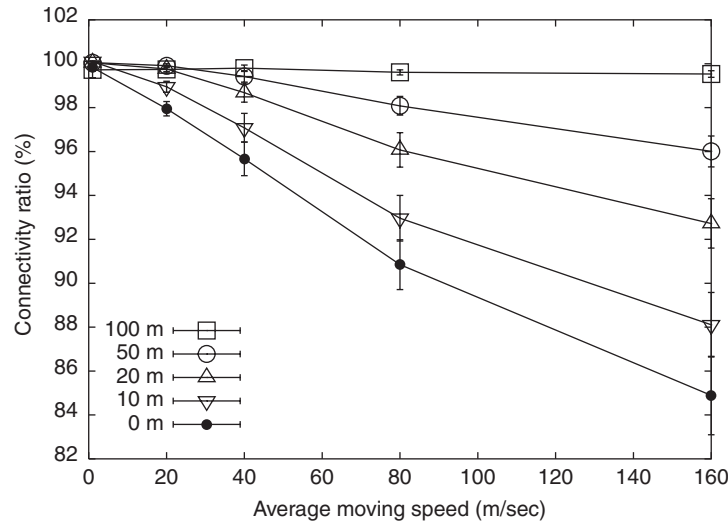


FIGURE 22.10 Connectivity ratio of a CDS algorithm under different buffered ranges.

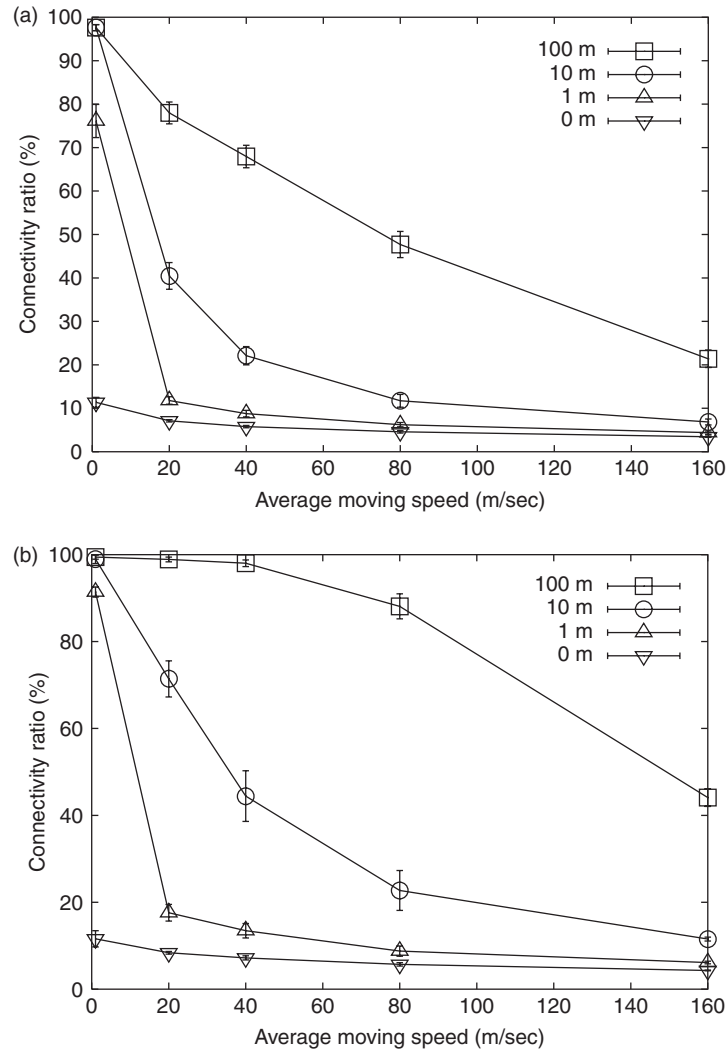
## 22.6 Simulation Results

The mobility control mechanisms for localized CDS and topology control algorithms can be evaluated using a popular network simulator *ns-2*. Simulation results show that, under moderate mobility, most existing CDS and topology control algorithms can no longer maintain a connected dominating set or connected topology in a MANET. On the other hand, global domination and connectivity can be preserved with high probability, when the mobility control mechanisms are used to enhance link availability and view consistency.

Figure 22.10 and Figure 22.11 show sample results from simulations of a CDS algorithm and a topology control algorithm. For results of more algorithms, the readers can refer to [48,47]. In the simulation, 100 nodes are randomly placed in a  $900 \times 900 \text{ m}^2$  area. The normal transmission range  $r$  is 250 m. The “Hello” interval is 1 sec. The mobility pattern follows the random waypoint model, where each node selects a random destination and moves to this destination with a random speed. After a node arrives at a destination, it pauses for a little while, and then selects a new destination and continues moving. This simulation uses a pause time of zero and an average moving speed varying from 1 to 160 m/sec. An ideal medium access control (MAC) layer without collision and contention is used to isolate the effect of mobility from other factors. The network connectivity is measured in terms of the connectivity ratio, which is defined as the ratio of pairs of connected nodes to the total number of pairs. In the simulation of a CDS algorithm, links between dominatees are removed before computing the connectivity ratio. That is, dominatees must be connected via dominators.

Figure 22.10 shows the connectivity ratio of Dai and Wu’s general rule [10], which is an extension of Wu and Li’s marking process and Rules 1 and 2 [49]. Different buffer zones are used to solve the link availability problem. In the original algorithm (0 m), the connectivity ratio drops rapidly as the average moving speed increases. When a small (20 m) buffer zone is used to tolerate broken links, the delivery ratio improves significantly under low (1 m/sec) to moderate (40 m/sec) mobility. With a 100 m buffer zone, the algorithm has almost 100% connectivity ratio under very high (160 m/sec) mobility. The effect of inconsistent views is not obvious in this case, because the buffer zone also increases redundancy in the CDS. A CDS with higher redundancy can tolerate more “bad” decisions caused by inconsistent views.

Figure 22.11(a) shows the connectivity ratio of Li, Hou, and Sha’s topology control algorithm based on local MST [22]. Various buffer zones are used to tolerate the inaccurate node location information caused by mobility. When there is no buffer zone (0 m), the connectivity ratio is very low (10%) under an average



**FIGURE 22.11** Connectivity ratio of a topology control algorithm under different buffered ranges. (a) With inconsistent local views. (b) With consistent local views.

moving speed of 1 m/sec. The connectivity ratio increases to 75% after a 1 m buffer zone is used, and 98% for a 10 m buffer zone. Obviously, the topology connectivity can be enhanced via a buffer zone with an appropriate width. On the other hand, using a buffer zone alone does not solve the entire problem. Hundred percent connectivity ratio is not achieved even with very low mobility level. Moderate and high mobility causes low connectivity ratio.

Figure 22.11(b) shows the effect of combining the buffer zone technique with consistent views. After individual nodes construct consistent local views, the connectivity ratio is improved significantly. When using a 20 m buffer zone in MANETs with a 10 m/sec average moving speed, the connectivity ratio is 40% without consistent views, and 70% with consistent views. When using a 100 m buffer zone under a 40 m/sec average moving speed, the connectivity ratio reaches 98% with consistent views, while the original connectivity ratio without consistent views is only 70%.

Overall, simulation results confirm that the global connectivity can be compromised by both link availability and view consistency issues. Both issues can be overcome with mobility control mechanisms, and the global property can be preserved with high probability and relatively small overhead.

## 22.7 Conclusion

We have addressed issues related to mobility control in mobile ad hoc networks. To illustrate the importance of the negative impact of mobile nodes on various protocols, we focus on two localized protocols, one for CDS construction and the other for topology control. It has been shown that most existing protocols on CDS construction and topology control will generate incorrect results in the presence of mobile nodes. We discuss two major problems caused by mobility control: link availability and view consistency, and provide several solutions. Mobility control in MANETs is still in its infancy. Many open issues exist:

- How does mobility affect protocols at other layers?
- Can approaches for view consistency in distributed systems be applied in mobile ad hoc networks?
- How should various kinds of cost and efficiency trade-off be done?

More efforts are needed to address these issues before various protocols can be applied in MANETs with mobile nodes.

## Acknowledgement

This work was supported in part by NSF grants CCR 0329741, ANI 0073736, and EIA 0130806.

## References

- [1] C. Adjih, P. Jacquet, and L. Viennot. Computing connected dominated sets with multipoint relays. Technical Report 4597, INRIA-Rapport de recherche, Oct. 2002.
- [2] K.M. Alzoubi, P.-J. Wan, and O. Frieder. Distributed heuristics for connected dominating sets in wireless ad hoc networks. *Journal of Communications and Networks*, 4: 22–29, 2002.
- [3] A.D. Amis, R. Prakash, T.H.P. Vuong, and D.T. Huynh. Max–min d-cluster formation in wireless ad hoc networks. In *Proceedings of the IEEE INFOCOM 2000*, pp. 32–41, 2000.
- [4] M. Bahrangiri, M. Hajiaghayi, and V.S. Mirrokni. Fault-tolerant and 3-dimensional distributed topology control algorithm in wireless multi-hop networks. In *Proceedings of the ICCCN*, pp. 392–397, 2002.
- [5] S. Basagni. Distributed clustering for ad hoc networks. In *Proceedings of the 1999 International Symposium on Parallel Architectures, Algorithms, and Networks (I-SPAN'99)*, pp. 310–315, June 1999.
- [6] S. Basagni, I. Chlamtac, A. Farago, V.R. Syroitiuk, and R. Talebi. Route selection in mobile multimedia ad hoc networks. *Proceeding of the IEEE MOMUC*, Nov. 1999.
- [7] D. Blough, M. Leoncini, G. Resta, and P. Santi. The K-Neigh protocol for symmetric topology control in ad hoc networks. In *Proceedings of the MobiHoc*, pp. 141–152, June 2003.
- [8] J. Cartigny, D. Simplot, and I. Stojmenovic. Localized minimum-energy broadcasting in ad hoc networks. In *Proceedings of the IEEE INFOCOM*, pp. 2210–2217, 2003.
- [9] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris. Span: an energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks. *ACM Wireless Networks Journal*, 8: 481–494, 2002.
- [10] F. Dai and J. Wu. Distributed dominant pruning in ad hoc wireless networks. In *Proceeding of the ICC*, p. 353, May 2003.
- [11] F. Dai and J. Wu. Performance comparison of broadcast protocols for ad hoc networks based on self-pruning. In *Proceedings of the IEEE WCNC*, Mar. 2004.
- [12] B. Das, R. Sivakumar, and V. Bharagavan. Routing in ad hoc networks using a spine. In *Proceedings of the 6th IEEE International Conference on Computers Communications and Networks (IC3N)*, pp. 1–20, Sep. 1997.
- [13] R. Dube, C. Rais, K. Wang, and S. Tripathi. Signal stability based adaptive routing (ssa) for ad hoc mobile networks. *IEEE Personal Communications* 4: 36–45, 1997.



- [14] A. Ephremides, J.E. Wjeselthier, and D.J. Baker. A design concept for reliable mobile radio networks with frequently hopping signaling. *Proceedings of the IEEE*, 71: 56–73, 1987.
- [15] R.H. Frenkiel, B.R. Badrinath, J. Borras, and R.D. Yates. The infostations challenge: balancing cost and ubiquity in delivering wireless data. *IEEE Personal Communication*, 7: 66–71, 2000.
- [16] K.R. Gabriel and R.R. Sokal. A new statistical approach to geographic variation analysis. *Systematic Zoology*, 18: 259–278, 1969.
- [17] M. Grossglauser and D.N.C. Tse. Mobility increases the capacity of ad-hoc wireless networks. In *Proceedings of the INFOCOM*, pp. 1360–1369, 2001.
- [18] Z.J. Haas, J.Y. Halpern, and L. Li. Gossip-based ad hoc routing. In *Proceedings of the IEEE INFOCOM*, vol. 3, pp. 1707–1716, June 2002.
- [19] W.-I. Kim, D.H. Kwon, and Y.-J. Suh. A reliable route selection algorithm using global positioning systems in mobile ad-hoc networks. In *Proceedings the ICC*, June 2001.
- [20] L. Li and J.Y. Halpern. Minimum energy mobile wireless networks revisited. In *Proceeding of the ICC*, pp. 278–283, June 2001.
- [21] 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 *Proceedings of the PODC*, pp. 264–273, Aug. 2001.
- [22] N. Li, J.C. Hou, and L. Sha. Design and analysis of and MST-based topology control algorithm. In *Proceedings of the INFOCOM*, vol. 3, pp. 1702–1712, Mar./Apr. 2003.
- [23] C.R. Lin and M. Gerla. Adapting clustering for mobile wireless networks. *IEEE Journal on Selected Areas in Communications*, 15: 1265–1275, 1996.
- [24] J. Liu and B. Li. MobileGrid: capacity-aware topology control in mobile ad hoc networks. In *Proceedings of the ICCCN*, pp. 570–574, Oct. 2002.
- [25] 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 *Proceedings of the MobiHoc*, pp. 123–134, June 2002.
- [26] A. Qayyum, L. Viennot, and A. Laouiti. Multipoint relaying for flooding broadcast message in mobile wireless networks. In *Proceedings of the HICSS-35*, Jan. 2002.
- [27] R. Ramanathan and R. Rosales-Hain. Topology control of multihop wireless networks using transmit power adjustment. In *Proceedings of the INFOCOM*, pp. 404–413, Mar. 2000.
- [28] M.Q. Rieck, S. Pai, and S. Dhar. Distributed routing algorithm for wireless ad hoc networks using d-hop connected dominating sets. In *Proceedings of the 6th International Conference on High Performance Computing in Asia Pacific Region*, Dec. 2002.
- [29] V. Rodoplu and T.H. Meng. Minimum energy mobile wireless networks. *IEEE Journal of Selected Areas in Communications*, 17: 1333–1344, 1999.
- [30] N. Sadagopan, F. Bai, B. Krishnamachari, and A. Helmy. Paths: analysis of path duration statistics and their impact on reactive manet routing protocols. In *Proceedings of the MobiHoc*, June 2003.
- [31] M. Seddigh, J. Solano, and I. Stojmenović. RNG and internal node based broadcasting in one-to-one wireless networks. *ACM Mobile Computing and Communications Review*, 5: 37–44, Apr. 2001.
- [32] P. Shinha, R. Sivakumar, and V. Bharghavan. CEDAR: a core-extraction distributed ad hoc routing algorithm. *IEEE Journal on Selected Areas in Communications, Special Issue on Ad Hoc Networks*, 17: 1454–1465, Aug. 1999.
- [33] P. Shinha, R. Sivakumar, and V. Bharghavan. Enhancing ad hoc routing with dynamic virtual infrastructures. In *Proceedings of the IEEE INFOCOM 2001*, pp. 1763–1772, 2001.
- [34] 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 *Proceedings of the MOBIHOC*, pp. 233–244, June 2003.
- [35] I. Stojmenovic, M. Seddigh, and J. Zunic. Dominating sets and neighbor elimination based broadcasting algorithms in wireless networks. *IEEE Transactions on Parallel and Distributed Systems*, 13: 14–25, 2002.
- [36] W. Su, S.-J. See, and M. Gerla. Mobility prediction and routing in ad hoc wireless networks. *International Journal of Network Management*, 11: 3–30, 2001.

- [37] C. Toh. Associativity-based routing for ad hoc mobile networks. *IEEE Personal Communications*, 4: 36–45, 1997.
- [38] G. Toussaint. The relative neighborhood graph of finite planar set. *Pattern Recognition*, 12: 261–268, 1980.
- [39] Y.-C. Tseng, S.-Y. Ni, Y.-S. Chen, and J.-P. Sheu. The broadcast storm problem in a mobile ad hoc network. *Wireless Networks*, 8: 153–167, 2002.
- [40] A. Vahdat and D. Becker. Epidemic routing for partially connected ad hoc networks. Technical Report CS-200006, Duke University, Apr. 2000.
- [41] P.J. Wan, K. Alzoubi, and O. Frieder. Distributed construction of connected dominating set in wireless ad hoc networks. In *Proceedings of the IEEE INFOCOM*, vol. 3, pp. 1597–1604, June 2002.
- [42] Y. Wang, X. Li, P. Wan, and O. Frider. Distributed spanners with bounded degree for wireless ad hoc networks. *International Journal of Foundations of Computer Science*, 14: 183–200, 2003.
- [43] R. Wattenhofer, L. Li, V. Bahl, and Y.M. Wang. Distributed topology control for power efficient operation in multihop wireless ad hoc networks. In *Proceedings of the INFOCOM*, pp. 1388–1397, Apr. 2001.
- [44] J.E. Wieselthier, G.D. Nguyen, and A. Ephremides. On the construction of energy-efficient broadcast and multicast trees in wireless networks. In *Proceedings of the INFOCOM*, pp. 585–594, Mar. 2000.
- [45] B. Williams and T. Camp. Comparison of broadcasting techniques for mobile ad hoc networks. In *Proceedings of the MobiHoc*, pp. 194–205, June 2002.
- [46] J. Wu. Extended dominating-set-based routing in ad hoc wireless networks with unidirectional links. *IEEE Transactions on Parallel and Distributed Systems*, 13: 866–881, 2002.
- [47] J. Wu and F. Dai. Mobility management and its applications in efficient broadcasting in mobile ad hoc networks. In *Proceedings of the INFOCOM*, Mar. 2004.
- [48] J. Wu and F. Dai. Mobility-sensitive topology control in mobile ad hoc networks. In *Proceedings of the IPDPS*, Apr. 2004. accepted to appear.
- [49] J. Wu and H. Li. On calculating connected dominating set for efficient routing in ad hoc wireless networks. In *Proceedings of the DialM*, pp. 7–14, 1999.

AQ: Please  
update  
Reference 48.