Wireless Personal Communications (2007) 43:605–621
DOI 10.1007/s11277-007-9252-9

MANET Routing with Provably Low Complexity Through Constant Density Clustering and Route Request Broadcast

HYO-SIK YANG[1], LUKE RITCHIE[2], ANDRÉA W. RICHA[3]
and MARTIN REISSLEIN[2]

[1] *Department of Computer Engineering, Sejong University, Seoul, Korea*
E-mail: hsyang@sejong.ac.kr
[2] *Department of Electrical Engineering, Arizona State University,Goldwater Center, MC 5706,TempeAZ 85287–5706,USA*
E-mail: {Luke.Ritchie, reisslein}@asu.edu
[3]*Department of Computer Science and Engineering, Arizona State University, Tempe, AZ, USA*
E-mail: aricha@asu.edu

Abstract. As mobile ad hoc networks (MANETs) are emerging as important components in critical and large-scale applications, it is crucial to develop MANET routing mechanisms with provably low complexity. In this paper, we give a tutorial overview of the efficient use of elementary node clustering and route request broadcast mechanisms for low-complexity MANET routing. We explain these mechanisms with illustrative examples and discuss their theoretical performance characteristics. We demonstrate that node clustering with constant density and route request broadcasting with a doubling radius technique over the network of cluster leaders can be employed for MANET routing with theoretically proven low complexity. Moreover, we contrast these efficient elementary clustering and route request broadcast mechanisms with clustering and route information accumulation mechanisms in the widely studied AODV and DSR routing protocols and discuss the implications of these various mechanisms for scalable MANET routing.

Keywords: 1-hop clustering, algorithm/protocol design and analysis, message complexity, routing protocol, scalability, time complexity, wireless mobile ad hoc network

1. Introduction

As large-scale mobile ad hoc networks (MANETs) are envisioned for a variety of applications, including military, medical, and entertainment applications, it becomes important to understand the fundamental performance characteristics of the central MANET functionalities. Routing, i.e., the process of finding a route from a source node to a destination node, is one of the most central functionalities in a MANET. Indeed, a large number of studies have contributed to a better understanding of MANET routing. However, the formal analysis of the complexity of MANET routing has only recently attracted some attention, see e.g., [1–7]. It has been found that the complexity of the existing MANET routing algorithms is typically proportional to the overall network size, e.g., the total number of nodes in the network, unless restrictive assumptions about the node locations or their mobility patterns are made.

In this paper, we give a tutorial overview of a recent line of algorithm-theoretic work [8, 9] that has formally analyzed the elementary mechanisms of node clustering and route request broadcast for low-complexity MANET routing. More specifically, we explain how (i) node clustering with a specific constant density of cluster leaders (whereby density refers to the number of cluster leaders per unit area) that adapts optimally to node mobility can be combined with (ii) broadcasting of route requests over the network of cluster leaders with a doubling radius technique to achieve MANET route discovery with a complexity proportional to the distance between the source and destination node and independent of the overall network size. The Cluster Overlay Broadcast (COB) routing protocol [9] is build upon the combination of these two elementary mechanisms, which are broadly applicable in MANET routing.

We explain the basic mechanisms of node clustering and route request broadcast with illustrative examples that provide insight into their complexity properties. We discuss the underpinnings of the formal algorithm-theoretic results for the node clustering and route request broadcast mechanisms and outline their implications for efficient MANET routing. We further illustrate the performance of COB with selected simulation results. We contrast the examined efficient clustering and route request broadcast mechanisms with the clustering and route information accumulation mechanisms that have been proposed in the context of the well known ad hoc on demand distance vector (AODV) and the dynamic source routing (DSR) protocols. From these comparisons we identify lessons learned from the formal analysis of the basic clustering and route request broadcast mechanisms for the design of low-complexity MANET routing protocols.

1.1. Related Work

In general, MANET routing can be conducted over a *flat* network where all nodes participate equally in the route discovery. In a dense network area containing many nodes per unit area, the flat network approach becomes typically unscalable with many nodes exchanging route requests and responses [10, 11]. Grouping nodes in *clusters* and conducting the route discovery primarily through the cluster leaders can mitigate these scalability problems. Research on node clustering in MANETs over the past two decades has resulted in a wide variety of clustering algorithms, with the lowest-ID [12, 13] and Least Cluster Change (LCC) [14] algorithms having been widely considered and used as benchmarks for comparisons [15]. The developed algorithms have typically been evaluated through simulations which give valuable insights, but provide only a limited understanding of the underlying fundamental performance limitations. As part of our algorithm theoretic line of work we have analyzed the mobile piercing set problem [8], a computational geometry problem, which provides a formal framework for studying clustering in MANETs. Our formal analysis provides fundamental insights into the properties of common clustering algorithms, such as the LCC algorithm. These formal analysis results complement and underscore properties which have only been empirically observed before and which we exploit for developing a route discovery approach with unprecedented low complexity.

The elementary mechanism of broadcasting (i.e., flooding) route requests through parts of the network is used in essentially all MANET routing protocols for discovering routes [16, 17]. Naive forwarding of the broadcast message by every node in a flat network results typically in many redundant copies, the so-called broadcast storm problem [18]. This broadcast storm problem can be addressed by limiting the number of nodes that forward broadcast messages [19, 20]. The strategy of first clustering the network and then having only cluster leaders forward the route requests has been extensively examined through simulations, see for instance [14, 10, 21, 22]. With this strategy, the discovered routes traverse only cluster leaders, forming so-called backbones [23].

We complement these existing studies with a tutorial overview of a formal analysis of the basic mechanisms of node clustering and route request broadcast over the network of cluster leaders that have been previously only examined through simulations. We explain that it is possible to significantly reduce the complexity of MANET route discovery by judiciously exploiting the insights from our formal analyses of clustering and route request broadcast. Specifically, the complexity for route discovery can be proportional to the shortest source-to-destination distance, and independent of the overall network size. This complexity is significantly lower than those of previously formally analyzed routing protocols which had complexities directly proportional to the overall network size [1–7] (which can be much larger than the shortest source-to-destination distance).

2. Network Setup and Model

We follow the widely considered basic unit-disk model, which represents the nodes' communication range as a circular disk, and make no other assumptions about the density or distribution of the nodes within the network. Nodes may move in any pattern – we only assume they do not move so quickly that wireless links are broken during the short time period it takes to discover a route. Our architecture does not make any non-standard assumptions about the medium access layer. As long as there is some reliability built into the broadcast mechanism, then each transmission is received by all neighboring nodes with high probability within a period of time we call a "time step."

We employ commonly available capabilities to realize two transmission ranges (as also employed in [22]): a shorter range for communication within clusters and a longer range for communication between cluster leaders. Successful route discovery requires that the network is connected with short-range transmissions, i.e., that there exists a route from any node in the network to any other node with only short-range transmission. We consider long-range transmissions that reach three times as far as short-range transmissions. This ensures in a connected network that the leaders of adjacent clusters are always able to communicate in one long-range hop, as will be illustrated in the next section on clustering. The two transmission ranges do not mean that every node needs two transmitters: two power levels, or a "logical" long-range channel implemented as a series of shorter transmissions would work equally well. It is important that all nodes have this capability, since any of them may be required to function as a cluster leader at some point.

3. Node Clustering

3.1. Initial Clustering and Properties of Cluster Cover

Initially, all nodes begin with the same neutral status, see Figure 1(a). Then, each node broadcasts its ID number using the short transmission range. Any node which does not receive an ID lower than its own ID marks itself as a leader, and signals all nodes within the short transmission range to become member nodes. In our example, nodes 0,1,2,4,5,7,8, and 13 have the lowest IDs in their respective neighborhoods. This process is illustrated in Figure 1(b),

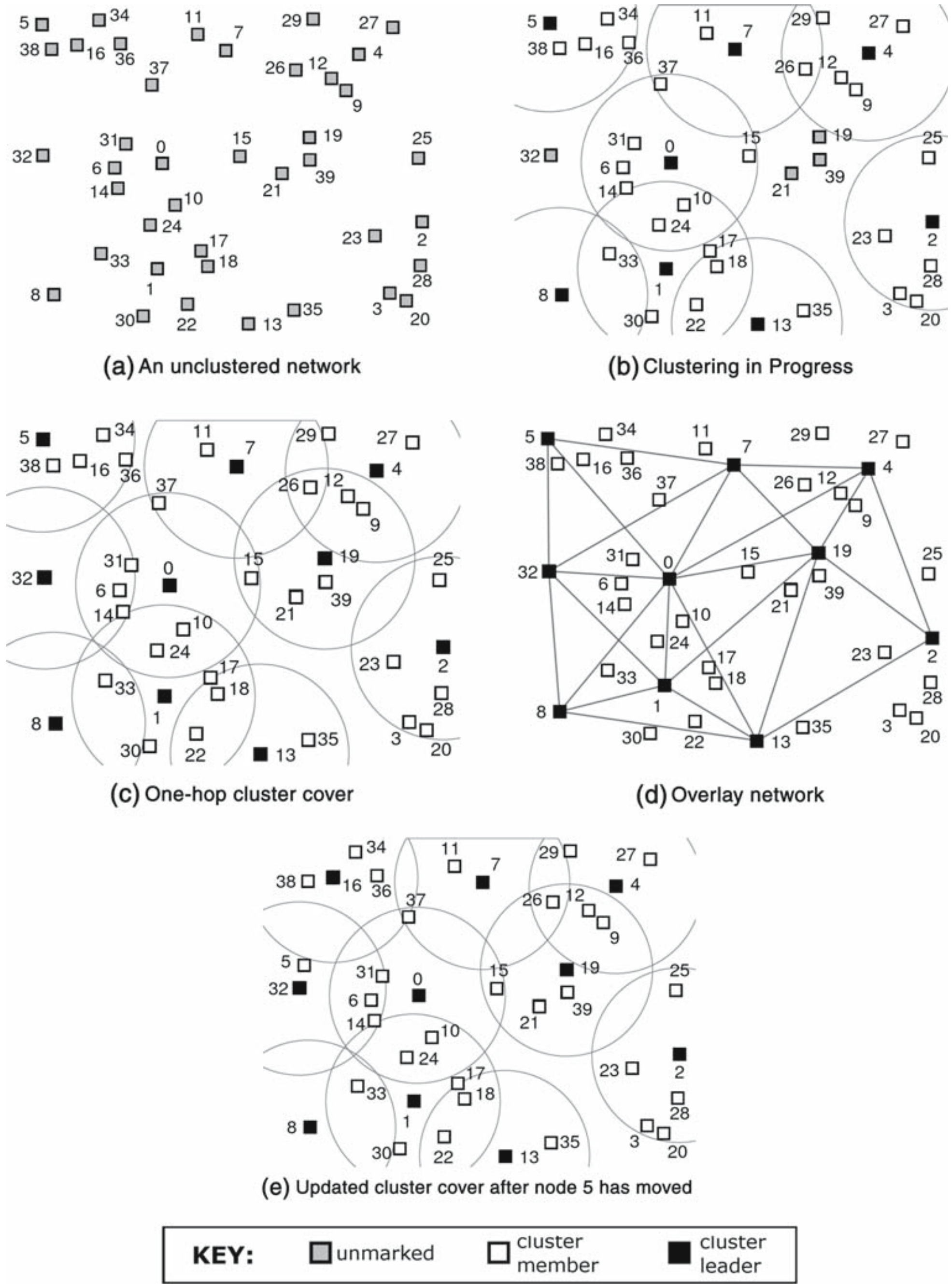


Figure 1. Illustration of clustering, routing over the overlay network, and update of cluster cover after node mobility.

where these first two steps are shown: leaders marked in black, members marked in white. At this point, a few nodes are still unmarked, shown in gray. Since 19 and 32 have the lowest *unmarked* IDs in their short-range neighborhoods, they then become cluster leaders as well, as illustrated in Figure 1(c). Once the entire network is marked, the cluster leaders can function as an overlay network, see Figure 1(d), capable of reaching any member node in one short-range hop – i.e., we have obtained a so-called one-hop cluster cover of the network.

The outlined algorithm corresponds to the LCC algorithm [14], which may use the lowest-ID clustering algorithm [12, 13] for the initial cluster formation. We have proven that for this algorithm, the density of the overlay network of cluster leaders – i.e., the number of cluster leaders per unit area – is no higher than a constant, which is independent of the number of nodes or their distribution; specifically the cluster leader density is no higher than $4/\pi$ [9]. This result in turn can be used to show that a given cluster leader has no more than 49 other cluster leaders within its long transmission range. These two results are important for two main reasons. First, the bounded density of cluster leaders and bounded number of neighboring cluster leaders limit the number of cluster leaders that receive and forward route requests, which allows us to bound the complexity of the routing, as discussed in Section 4.2. Second, the limited number of neighboring cluster heads aids in the medium access control for the long-range transmissions, as only a limited number of long-range transmission can collide. The clustering algorithm does not require cluster leaders to maintain lists of the regular nodes in their clusters. Instead, each regular member node only keeps track of which node it considers its leader, resulting in low-storage overhead.

We have also proven that the density of the network of cluster leaders (i.e., of the one-hop cluster cover) can not be much smaller than that achieved by the LCC algorithm. More specifically, if the LCC algorithm finds a one-hop cluster cover consisting of $\mathcal{C}$ cluster leaders, it is absolutely not possible to achieve the one-hop cluster cover with less than $\mathcal{C}/7$ cluster leaders [8]. This result is important as it implies that no other algorithm, no matter how sophisticated, could find a one-hop cluster cover with significantly fewer (less than a seventh) of the number of leaders declared by the LCC algorithm. Clearly, having as few cluster leaders as possible is preferable to ensure that all network functions that operate on the overlay network of cluster leaders, such as routing, have as simple a view of the network as possible. Our theoretical results prove that the overlay network can not be much simpler than that found with the LCC algorithm. We have also shown that the complexity of forming the initial cluster cover with the LCC algorithm is linearly proportional to the number of formed clusters [8].

In summary, we have shown that the LCC clustering algorithm (which may employ the lowest-ID technique for the initial clustering) forms an overlay network of cluster leaders, whereby each node is within the short-range distance of a cluster leader. The density of the network of cluster leaders (number of cluster leaders per unit area) is no higher than $4/\pi$ and the density of a valid one-hop cluster cover can not be less than seven times smaller than that achieved by the LCC algorithm. These two main results hinge critically upon the fact that no two cluster leaders are ever within the short transmission range of each other. Any arbitrary clustering technique that ensures that two cluster leaders are at least the short-range distance apart has the $4/\pi$ upper bound on the cluster leader density and no valid one-hop cluster cover can have less than one seventh of the cluster leaders declared by the considered clustering technique. Also, these results hold regardless of the node distribution in the network, as long as the entire network is connected by short-haul hops.

3.2. Update of Clustering Upon Node Mobility: Algorithm Outline and Theoretical Results

There are two events that require an update of the clustering, i.e., a change in the nodes designated as cluster leaders:

(i) A cluster leader moves and leaves at least one node uncovered (outside the short transmission range of any existing cluster leader), or a regular node moves and becomes uncovered. In either case, the uncovered node(s) run the initial clustering procedure outlined above. In the example illustrated in Figure 1(e), cluster leader 5 moves and leaves regular nodes 16, 34, 36, and 38 uncovered. These uncovered nodes then form a new cluster with node 16 as the leader.

(ii) A cluster leader A moves within the short-haul transmission range of another cluster leader B. Then the leader that moved (A) is demoted to a regular node. In Figure 1(e), cluster leader 5 moved within the short-haul transmission range of cluster leader 32. Node 5 is therefore demoted to a regular node within the cluster led by node 32.

If a regular node moves from within the short transmission range of one cluster leader to within the short range of another leader, then no update of the clustering is required.

The outlined update mechanisms correspond to the LCC algorithm [14] with the subtle modification that for update event (ii) the cluster leader that moved is demoted. (The LCC specification in [14] allows either leader A or B to be demoted according to the lowest-ID, highest-connectivity, or some other tie-breaking rule. Demoting the node that moved typically results in somewhat less overhead, but the following complexity results hold also for any arbitrary tie-breaking rule.) We have proven that the updates required by both events (i) and (ii) can be completed within a small constant amount of time, that is, independent of the number of nodes [8]. We have also shown that when k nodes are uncovered due to an event of type (i), then the number of messages required to update the clustering in linearly proportional to k. No other deterministic algorithm can update the clustering with a number of messages, that is, smaller than a constant times k [8]. Importantly, the updates according to the outlined LCC algorithm preserve the properties (such as bounded constant density) of the one-hop cluster cover.

4. Route Discovery Through Route Request Broadcast

4.1. Description and Illustration of Route Request Broadcast over Network of Cluster Leaders

Routing begins when a cluster leader receives a new message, either from itself or via a short range transmission from a cluster member. For an illustrative example, consider a message generated by node 16 and destined to node 25. Node 16 sends the message to its cluster leader, node 5. These transmissions from member to leader nodes always use the short-range. The leader (node 5) then begins flooding the network with a route request (RREQ) for the message using a version of an expanding ring search. First, it long-range broadcasts the RREQ with a time to live (TTL) equal to 1 that only reaches as far as the adjacent cluster leaders. If a reply does not come back in two time steps, the leader sends another RREQ with twice the previous TTL. This process continues with the doubling of TTL and waiting time until the RREQ reaches its destination and the destination node (25) returns a route acknowledgement (ACK). Note that the TTL limits the broadcast radius to no more than TTL long-range transmissions.

In the first round, the single RREQ broadcast by cluster leader node 5 reaches adjacent cluster leaders (in our example, nodes 0, 7, and 32) and the rest of the regular nodes in its cluster (nodes 34, 36, and 38). Other regular nodes (such as 11 or 31) are in range to hear this

Table 1. Route Requests (RREQs) stored in cluster leaders 7, 19, and 2 during the third broadcast round of the node 16-to-node 25 route discovery. The predecessor node entries are used to pass back the acknowledgement for the discovered route to the source. The incoming Time to Live (TTL) values help prevent routing loops

At node number	7	19	2
Source			
Destination	…	…	…
Predecessor			
Broadcast round			
Incoming TTL			
Source	16	16	16
Destination	25	25	25
Predecessor	15	7	19
Broadcast round	3	3	3
Incoming TTL	4	3	2
Source			
Destination	…	…	…
Predecessor			
Broadcast round			
Incoming TTL			

broadcast, but ignore any RREQ not from their respective cluster leader nodes. During the second round, the TTL is set to two, and node 5 waits for four time steps before beginning the next round. Leader nodes 1, 4, 8, 13, and 19 also receive the RREQ as well as the members of clusters 0, 7, and 32. In the third broadcast round, the TTL is set to four and a RREQ reaches node 2, the cluster leader for our considered destination node 25, by several paths. In our example, we assume that the RREQ traveling through nodes 7 and 19 is the first to arrive. Node 2 broadcasts the RREQ with a long-haul transmission reaching both its neighboring cluster leaders and the regular nodes in its cluster, including the destination node 25.

During flooding, each intermediate cluster leader stores a copy of the first RREQ it receives for a given route discovery during a given broadcast round. The RREQ stored at a given cluster leader contains the ID of the source node, the ID of the destination node, the ID of the predecessor node from whom the RREQ was received, the number of the current broadcast round, and the TTL value with which the RREQ was received. The RREQs stored at cluster leaders 7, 19, and 2 during the third broadcast round are illustrated in Table 1.

Once node 2 receives the ACK from node 25 (via a short-range transmission), the ACK is forwarded back along the path marked by the stored RREQs in nodes 2, 19, and 7. The total round trip is eight hops, so it reaches node 5 before the next round begins. As intermediate leaders forward the ACK, they mark the route as active, and this marked route is used to forward the actual message from nodes 5 to 25. Because this is a purely reactive protocol, very little information needs to be stored at any one time. At most, there will be one stored RREQ for each route discovery currently underway and for each active route. Routing loops are avoided by passing the ACK back to a predecessor node which was reached with a higher incoming TTL field, i.e., the predecessor node is at least one-hop closer to the source.

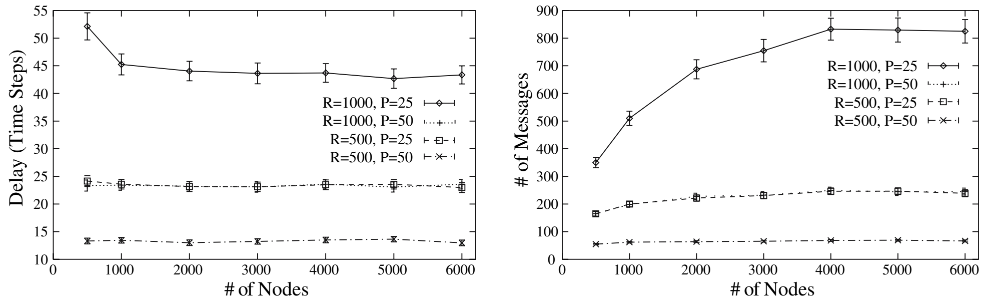
The COB routing protocol [9] incorporates the efficient node clustering mechanism outlined in Section 3 and the efficient route discovery mechanism outlined in this section. These mechanisms are broadly applicable to MANET routing and can be incorporated in a wide range of MANET routing protocols. To fix ideas, we consider in the following COB as an instantiation of a routing protocol based on the outlined efficient node clustering and route request broadcast mechanisms.

4.2. Theoretical Results for Route Request Broadcast

We let Δ denote the length of the shortest possible source-to-destination path in short-range hops. We have proven that the route discovery approach outlined in Section 4.1 finds a route with a hop distance that is linearly proportional to the shortest source-to-destination hop distance Δ [9]. Note that this results holds even though we are employing long-range hops between cluster leaders in the route discovery and the transmission of the actual message. To see this, observe that in a connected network, one long-range hop between cluster leaders generally corresponds to at most three short-range hops via at most two gateway nodes between cluster leaders, thus a hop distance measured in short-range hops is linearly proportional to a hop distance measured in long-range hops. The discovered path is also guaranteed to be loop-free.

The route discovery approach with RREQ broadcast over the constant-density network of cluster leaders limits the time and message complexity of route discovery. Since the broadcast radii are powers of two, the last round always accounts for more than half of the total overhead. And since the overlay network has a bounded constant density, the overhead is a function of the length Δ of the shortest source-to-destination path. More specifically, the number of broadcasts during route discovery is within a constant factor of Δ^2, and the time spent is within a constant of Δ, i.e., in standard asymptotic notation they are $O(\Delta^2)$ and $O(\Delta)$, respectively, [9]. These formally proven performance bounds make COB the first formally analyzed MANET routing protocol with time and message complexities that are linear, resp. quadratic, in the shortest source-to-destination distance, and independent of the overall network size, such as the total number of nodes in the network or the total hop-distance diameter of the network. Previous formally analyzed routing protocols have complexities that depend on the total number of nodes or the total hop distance diameter of the network [1, 2, 5], or require location aid, or special mobility patterns. Our results for the route discovery approach outlined in Section 4.1 hold for any arbitrary mobility pattern and node distribution, and do not require location aid.

A key theoretical result for the outlined route discovery approach is that it adapts asymptotically optimally to the mobility of the nodes. To understand this result, note that the outlined route discovery is purely on-demand, operating on top of a clustering cover of the network. Thus, any changes due to node mobility are absorbed by updates of the cluster cover. As noted above, we have shown that using the LCC algorithm, the total elapsed time of an update is a small constant, and thus asymptotically optimal. We have also shown that (i) the number of messages exchanged during an update of the clustering structure is linearly proportional to the number of nodes left uncovered due to the mobility of a node, and that (ii) no deterministic algorithm can update the cluster cover with a number of message exchanges, that is, less than linearly proportional to the number of uncovered nodes. Thus, the COB routing protocol, which routes on the network of cluster leaders, is asymptotically optimal in that no other routing algorithm could adapt to node movements in less than a constant amount of time (i.e., a time that is independent of the number of nodes involved) or with fewer message exchanges than a fixed constant times that number of involved nodes.



(a) Delay **(b)** Number of message transmissions

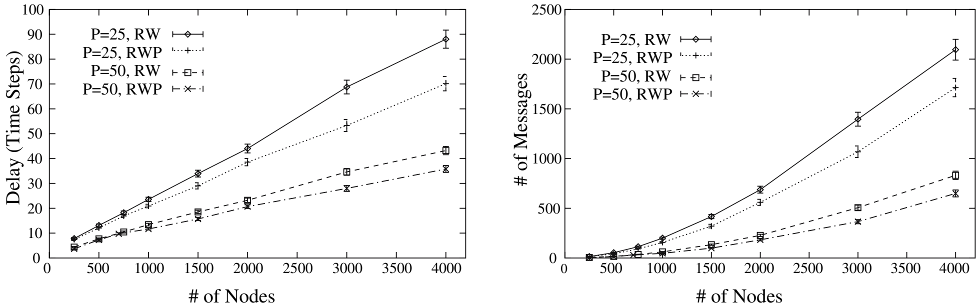
Figure 2. Node density scaling: overhead for a route discovery as a function of number of nodes N for different (fixed) network areas of $R \times R\,\mathrm{m}^2$ and short transmission ranges P m.

4.3. Simulation Results

In this section, we present selected simulation results that illustrate the key theoretical properties of the COB routing protocol, and refer to [9] for more details on the simulations. In the presented simulations, we evaluate COB with respect to the mobility process and the size of the network. We simulate the route discovery sequentially, randomly selecting a series of source/destination pairs. This ensures that we measure the network layer performance of COB, in isolation from any positive or negative effects of the MAC layer or cross-traffic. We include simulations for two scaling scenarios: (i) a node density scaling scenario, where the area of the network is a square of fixed size ($R \times R\,\mathrm{m}^2$) and the number of nodes (N) in the network is varied, and (ii) a network diameter scaling scenario, where we jointly scale up the number of nodes in the network and the diameter of the network area. The goal of the diameter scaling is to measure the performance as the shortest hop distance Δ between source and destination increases. Since scaling Δ directly is computationally prohibitive in a distributed simulation, we make the reasonable assumption that Δ scales approximately linearly with the diameter of the network, i.e., $\Delta \sim R$. It is important to note that the time and message complexity of COB depend only on the shortest source-destination distance and not on the overall network dimensions.

We conduct simulations for both the random walk (RW) and the random waypoint (RWP) mobility models with a mobile speed of 10 m/second. The pause time for the random waypoint mobility model is 10 seconds. In our simulation, a route discovery typically takes two orders of magnitude less time than changes in the cluster coverage due to node movement, so we can reasonably approximate the node positions as static during a given route discovery. The practical deployment of the COB routing protocol would of course require a means for recovering from node changes in the cluster coverage that affect an ongoing route discovery. Recovery mechanisms similar to those in existing routing protocols could be adapted to COB.

In Figures 2(a) and 2(b), we consider the node density scaling scenario and plot the delay for a route discovery (in time steps) and the number of message transmissions per route discovery. Generally, we observe that the delay for the route discovery does not change significantly as the node density increases. An exception is the sparsest network ($R = 1000$ m, $P = 25$ m, $N = 500$), where there are significant "uncovered" areas in the network not reachable by any node's long transmission range. Here, many routes require a path around the uncovered area –i.e., the routes tend to be more crooked and less straight in this scenario.



(a) Delay **(b)** Number of message transmissions

Figure 3. Diameter scaling: overhead for a route discovery as a function of number of nodes N with proportional ($R = N/2$) network area of $R \times R\,\mathrm{m}^2$ for different (fixed) short transmission ranges P m and mobility patterns (random walk (RW) and random waypoint (RWP)).

From Figure 2(b) we observe that the number of messages transmitted for a route discovery generally tends to initially increase and then level off as the node density increases. This effect is most pronounced for the network with the large area and the small transmission range. This is caused by an increasing number of clusters as the network becomes more populated, until they approach the maximum density of clusters. Once the entire network area–or at least the region involved in the doubling radius search–is covered by clusters, there is no further increase in the number of messages.

In Figures 3(a) and 3(b), we consider the diameter scaling scenario for both mobility models. We observe from Figure 3(a) that the route discovery delay increases linearly as we jointly scale up the diameter and number of nodes in the network (implicitly scaling up Δ). Also, we observe from Figure 3(b) that the number of messages transmitted for a route discovery appears to increase quadratically. We also observe that both the random walk and the random waypoint mobility models result in the same underlying trends in the hop distance, delay, and message complexity. This is exactly what we expect to see since the COB algorithm does not assume any specific mobility model. The difference in results for the random waypoint model seen in the plots is due to the slight tendency for the nodes to more densely populate the center of the network area using that model, resulting in somewhat lower constants in the asymptotic scaling behavior.

5. Discussion

In this section, we discuss the elementary node clustering and route request broadcast mechanisms examined in Sections 3 and 4 and contrast them with clustering and route information accumulation mechanisms that have been proposed in the context of extensively studied on-demand ad hoc routing protocols, such as AODV [21] and DSR [24].

5.1. Clustering

Clustering has been proposed as an addition to both AODV and DSR and a number of other routing protocols because it has some very useful properties that are quite generally applicable. The most commonly cited advantage is that hierarchical routes tend to be more stable: links

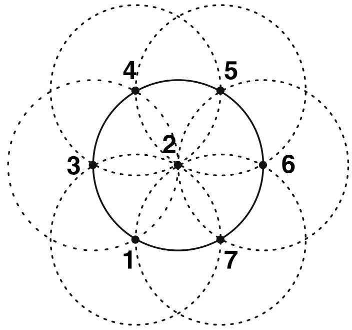


Figure 4. Worst case scenario of the ripple effect: if the central node gives up its leader status, up to six other nodes formerly belonging to its cluster may be far enough apart to become new cluster leaders.

between clusters do not break as often as links between individual nodes. And as we have seen, dividing a MANET into clusters can effectively limit the density of the overlay network. Lowering the effective density of a MANET is useful because high-density networks generally tend to experience more collisions and more congestion. Limiting density also limits the overhead of flooding the network with a route request by eliminating redundant broadcasts. It can also have the side effect of turning the cluster leaders into bottlenecks which must do more work and forward more traffic than the other nodes [25]. This is another reason for keeping the processing and storage load as low as possible in the node clustering approach outlined in Section 3.

It is important to note that a limited density of cluster leaders can only be guaranteed if the cluster leaders are always a minimum distance apart. In the LCC algorithm, they are never within the short communication range. Other clustering algorithms, such as the proposed addition to AODV, Adaptive Routing using Clusters (ARC) [10], do not have this property. ARC uses a subset rule to determine when cluster leaders change status. With the subset rule, a cluster leader that moves within the short range of another leader only gives up its leader status if all of its regular nodes also move within the short range of the other leader. This subset rule was developed in [10] in order to avoid the "ripple effect." This is the name for the situation when one change in leader status triggers additional changes in the clustering.

For an illustration of the worst-case ripple effect we consider the two update events of the LCC algorithm in the context of the scenario depicted in Figure 4. Suppose that node 2 in the illustration is a cluster leader and the nodes 1, 3, 4, 5, 6, and 7 at the edge of the short transmission range of node 2 are members of the cluster. Now suppose that node 2 moves outside of the illustrated area and leaves its member nodes uncovered. According to update event (i) from Section 3.2, the uncovered nodes organize themselves into clusters which results in the worst case (if the nodes 1, 3, 4, 5, 6, and 7 are just slightly outside of each others short range) in six new clusters. (In the best-case, when the nodes are within the short range only two new clusters are formed.) Next, consider the update event (ii) by considering again Figure 4. Suppose that node 1 is a cluster leader that has just moved from its old location outside the illustrated area to just within the short range of the cluster leader 2, which has nodes 3, 4, 5, 6, and 7 as regular member nodes. With our update rule of demoting the node that moved, node 1 demotes itself to a regular node joining the cluster led by node 2. With the lowest-ID tie breaking rule allowed for by LCC, node 1 stays cluster leader and node 2 is demoted. The demotion of node 2 leaves in the worst case the nodes 3, 4, 5, 6, and 7 uncovered (namely if

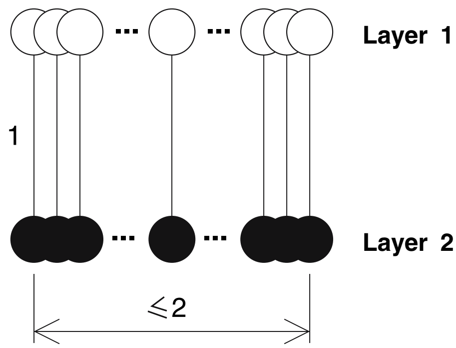


Figure 5. Each of the regular nodes (in white) in layer 1 is covered by exactly one leader (in black) one short-range away in layer 2. With the subset rule of ARC, none of the cluster leaders can give up its leader status. With LCC, between two and four leaders can cover all nodes.

nodes 3 and 7 are just slightly outside the short range of node 1). The uncovered nodes need to organize themselves into clusters, which in the worst case results in five new clusters. We see from these considerations, that the ripple effect in LCC is limited to a localized area and does not propagate across the network.

Realistically, several factors limit the ripple effect to significantly fewer cluster leader changes. If the node speed and clustering update interval take reasonable values, it is rare for an entire disk to be uncovered at once. For a relatively low-node density, the simulations in [10] show a higher, roughly proportional rate of cluster changes with LCC than with a subset rule. However, for higher densities, it is more likely that members nodes will be within range of two or more leaders, and even fewer nodes will be left uncovered by a leadership change. Changes in leader status are also a less important issue for COB than for AODV, because losing a leader does not mean also losing all of the route information it has accumulated (an aspect discussed in the next section).

The subset rule of ARC can lead to a very large number of leaders and possibly to an unbounded density of cluster leaders. To see this, consider the scenario depicted in Figure 5 where the cluster leaders are drawn in black and their members in white. More specifically, each of the leader nodes in layer 2 has one member node in layer 1. With the subset rule, none of the leaders can give up its leader status since each member node is only within the short range of exactly one leader node. As the number of nodes in layers 1 and 2 grows very large (while maintaining the depicted geometry), the number of cluster leaders becomes very large and so does the density of cluster leaders, which in turn can lead to a very large message complexity for route discoveries. Note that for the depicted geometry, all nodes could be covered with two clusters, one led by the middle node in layer 1, and one led by the middle node in layer 2. In the worst case, four leader nodes would be required with LCC, one at each end point of each layer.

5.2. Accumulation of Route Information

The route discovery outlined in Section 4, similar to the route discovery in DSR and AODV, is *reactive*, in that it finds routes as they are needed. The key advantage of this approach over *proactive* routing is the reduced routing load, which has a significant performance impact on wireless links [26]. However, AODV and DSR do behave, in another sense, proactively by caching and maintaining route information for later use once it has been discovered. This has the advantage of reducing the frequency of RREQ floods, at the price of increasing the overhead associated with storing and maintaining up-to-date route information. A comparison with the route discovery approach of Section 4 illustrates how this design decision may ultimately affect the scalability of a routing protocol.

In both DSR and AODV, each node maintains a route cache/route table. In AODV, this table contains a separate entry for each destination, which in a network with N nodes may eventually grow to $O(N)$ in size. By virtue of source routing, DSR has access to a larger amount of routing information than AODV. However, the size of an adaptive routing cache [27] may increase over time, in the limiting case, to include information about every possible route through a node (larger than $O(N)$). Each node in COB, on the other hand, stores just one entry for each active route through that node.

The price COB pays for provably limited storage is a RREQ flood for every new route. But as mentioned in Section 4, employing a doubling radius expanding ring search and limited broadcast through the constant-density overlay network limits the RREQ sending overhead to $O(\Delta^2)$. Starting fresh with each route also eliminates the additional complexity associated with stale routes such as resending data lost on invalid paths and avoiding count-to-infinity scenarios.

Even when a route is not stale or invalid, it still may not be the "best" route available. Most routing protocols, including AODV, are designed to select the shortest route in terms of the number of hops from source to destination. However, in a congested network, a route with more hops may actually be faster in terms of end-to-end delay. The "shortest" path may include critical links that are experiencing exceptional traffic loads. In this sense, the route discovery approach of Section 4.1 employs a type of load balancing, using a RREQ flood to select the route that replies first, regardless of hop count. (Although the doubling ring search does ensure that the discovered route will be no more than twice the shortest number of hops.)

Networks with highly mobile nodes can also diminish the effectiveness of routing tables/caches. If nodes move faster or more erratically, routes become stale even sooner. More route update messages must be sent, and nodes will have to work harder to update their information. While DSR outperforms AODV and COB in nearly-static networks, reliance on past information becomes a liability in higher-mobility scenarios. On a related note, consider the case of introducing additional nodes to the network or shutting down existing nodes. This may occur for a variety of reasons: power-saving "sleep" cycles; damage, repair, or expansion of network coverage; or a changing user population. All of these can be viewed as a special case of mobility, since they involve nodes changing location. Protocols such as DSR and AODV that rely on accumulation of route information over time will likewise be at a disadvantage in these types of situations.

6. Conclusions

In summary, we have provided tutorial guidelines for the efficient use of the elementary mechanisms of node clustering and route request broadcast in MANET routing. We have explained how these two basic mechanisms can form the basis for provably efficient route discovery with a delay (time complexity) linearly proportional to the shortest source-destination hop distance, i.e., $O(\Delta)$, and a number of message exchanges (message complexity) quadratically proportionally to the shortest source-destination distance, i.e., $O(\Delta^2)$. In contrast, the time and message complexities of previous formally analyzed routing protocols scaled with the total network size (e.g., total number of nodes, diameter of network), which is typically significantly larger than the shortest source-destination hop distance.

The examined basic node clustering and route request broadcast mechanisms are generally applicable to routing in a mobile ad hoc network. Comparison with clustering and route information accumulation mechanisms extensively studied in the context of existing routing protocols, such as DSR and AODV, has provided insights into the implications of these mechanisms for scalable MANET routing. As our comparison and formal analysis indicate, features such as density-limiting clustering, minimal storage, and purely-reactive route discovery all have provable benefits to scalability. And improved scalability is essential for expanding the potential applications for ad hoc networks by eliminating constraints on network mobility, density, and size.

References

1. M. Abolhasan, T. Wysocki, and E. Dutkiewicz, “A Review of Routing Protocols for Mobile Ad Hoc Networks”, *Ad Hoc Networks*, Vol. 2, No. 1, pp. 1–22, 2004.
2. X. Hong, K. Xu, and M. Gerla, “Scalable Routing Protocols for Mobile Ad Hoc Networks”, *IEEE Network*, Vol. 16, no. 4, pp. 11–21, 2002.
3. R. Rajamaran, “Topology Control and Routing in Ad Hoc Networks: A survey”, *SIGACT News*, Vol. 33, pp. 60–73, 2002.
4. A. Sankar and Z. Liu, “Maximum Lifetime Routing in Wireless Ad-hoc Networks”, in *Proc. of IEEE Infocom*, Hong Kong, 2004.
5. C.A. Santivanez, B. McDonald, I. Stavrakakis, and R. Ramanathan, “On the Scalability of Ad Hoc Routing Protocols”, in *Proc. of IEEE Infocom*, New York, NY pp. 1688–1697, 2002.
6. A. Srinivas and E. Modiano, “Minimum Energy Disjoint Path Routing in Wireless Ad-hoc Networks”, in *Proc. of ACM MobiCom*, San Diego, CA pp. 122–133, 2003.
7. J. Sucec and I. Marsic, “Hierarchical Routing Overhead in Mobile Ad Hoc Networks”, *IEEE Transactions on Mobile Computing*, Vol. 3, No. 1, pp. 46–56, 2004.
8. H. Huang, A.W. Richa, and M. Segal, “Approximation Algorithms for the Mobile Piercing Set Problem with Applications to Clustering in Ad Hoc Networks”, *Mobile Networks and Applications (MONET)*, Vol. 9, No. 2, pp. 151–161, 2004.
9. L. Ritchie, H.-S. Yang, A.W. Richa, and M. Reisslein, “Cluster Overlay Broadcast (COB): MANET Routing with Complexity Polynominal in Source-Destination Distance”, *IEEE Transactions on Mobile Computing*, Vol. 5, No. 6, pp. 653–667, 2006.
10. E.M. Belding-Royer, “Multi-level Hierarchies for Scalable Ad Hoc Routing” *Wireless Networks*, Vol. 9, No. 5, pp. 461–478, 2003.
11. E.M. Belding-Royer and C.E. Perkins, “Evolution and Future Directions of the Ad Hoc On-demand Distance-vector Routing Protocol” *Ad Hoc Networks*, Vol. 1, No. 1, pp. 125–150, 2003.
12. A. Ephremides, J.E. Wieselthier, and D.J. Baker, “A Design Concept for Reliable Mobile Radio Networks with Frequency Hopping Signalling”, *Proceedings of the IEEE*, Vol. 75, No. 1, pp. 56–73, 1987.
13. M. Gerla and J.T.C. Tsai, “Multicluster Mobile Multimedia Radio Networks”, *ACM-Baltzer Journal of Wireless Networks*, Vol. 1, No. 3, pp. 255–256, 1995.
14. C.-C. Chiang, H.-K. Wu, W. Liu, and M. Gerla, “Routing in Clustered Multihop, Mobile Wireless Networks with Fading Channel,” in *Proc. IEEE Singapore Int. Conf. on Networks*, Singapore pp. 197–211, 1997
15. J.Y. Yu and P.H.J. Chong, “A survey of Clustering Schemes for Mobile Ad Hoc Networks”, *IEEE Communications Surveys and Tutorials*, Vol. 7, No. 1, pp. 32–48, 2005.
16. O. Liang, A. Sekercioglu, and N. Mani, “A Survey of Multipoint Relay based Broadcast Schemes in Wireless Ad Hoc Networks”, *IEEE Communications Surveys and Tutorials*, Vol. 8, No. 4, pp. 30–46, 2006.
17. K. Viswanath, K. Obraczka, and G. Tsudik, “Exploring Mesh and Tree-based Multicast Routing Protocols for MANETs”, *IEEE Transactions on Mobile Computing*, Vol. 5, No. 1, pp. 28–42, 2006.

18. 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*, Vol. 8, No. 2–3, pp. 153–167, 2002.
19. B. Williams, D. Metha, T. Camp, and W. Navidi, "Predictive Models to Rebroadcast in Mobile Ad Hoc Networks", *IEEE Transactions on Mobile Computing*, Vol. 3, No. 3, pp. 295–303, 2004.
20. J. Wu and F. Dai, "Efficient Broadcasting with Guaranteed Coverage in Mobile Ad Hoc Networks", *IEEE Transactions on Mobile Computing*, Vol. 4, No. 3, pp. 259–270, 2005.
21. S.-J. Lee, E.M. Belding-Royer, and C.E. Perkins, "Scalability Study of the Ad Hoc On-demand Distance Vector Routing Protocol", *International Journal of Network Management*, Vol. 13, No. 2, pp. 97–114, 2003.
22. K. Xu and M. Gerla, "A Heterogeneous Routing Protocol based on a New Stable Clustering Scheme", in *Proc. of IEEE Milcom*, Anaheim, CA pp. 838–843, 2002.
23. J. Wu and F. Dai, "Virtual Backbone Construction in MANETs using Adjustable Transmission Ranges", *IEEE Transactions on Mobile Computing*, Vol. 5, No. 9, pp. 1188–1200, 2006.
24. D.B. Johnson, D.A. Malts, and J. Broch, "DSR: The Dynamic Source Routing Protocol for Multi-hop Wireless Ad Hoc Networks", in *Ad Hoc Networking*, Chapter 5, C.E. Perkins, Ed. Reading, MA: Addison–Wesley, pp. 139–172, 2001
25. Y. Qin and J. He, "The Impact on Throughput of Hierarchical Routing in Ad Hoc Wireless Networks", in *Proc. of IEEE ICC*, Seoul, Korea pp. 3010–3014, 2005
26. C.E. Perkins, E.M. Belding-Royer, S.R. Das, and M.K. Marina, "Performance Comparison of Two On-demand Routing Protocols for Ad Hoc Networks", *IEEE Personal Communications*, Vol. 8, No. 1, pp. 16–28, 2001.
27. X. Yu and Z. Kedem, "A Distributed Adaptive Cache Update Algorithm for the Dynamic Source Routing Protocol", in *Proc. of IEEE Infocom*, Miami, FL, 2005, pp. 730–739.

Martin Reisslein is an Associate Professor in the Department of Electrical Engineering at Arizona State University, Tempe. He received his Ph.D. in systems engineering from the University of Pennsylvania in 1998. From July 1998 through October 2000 he was a scientist with the German National Research Center for Information Technology (GMD FOKUS), Berlin and lecturer at the Technical University Berlin. He maintains an extensive library of video traces for network performance evaluation, including frame size traces of MPEG–4 and H.263 encoded video, at http://trace.eas.asu.edu.

Andréa W. Richa is an Associate Professor at the Department of Computer Science and Engineering at Arizona State University, Tempe, since August 2004. Prof. Richa received her M.S. and Ph.D. degrees from the School of Computer Science at Carnegie Mellon University, in 1995 and 1998, respectively. She also earned an M.S. degree in Computer Systems from the Graduate School in Engineering (COPPE), and a B.S. degree in Computer Science, both at the Federal University of Rio de Janeiro, Brazil, in 1992 and 1990, respectively. Prof. Richa's main area of research is in network algorithms. Prof. Richa's data tracking (or name lookup) algorithm has been widely recognized as the first benchmark algorithm for the development of distributed databases in peer-to-peer networking, having being references by over 130 academic journal or conference publications to date, and being implemented as part of two of the current leading projects in peer-to-peer networking. Dr. Richa's was the recipient of an NSF CAREER Award in 1999.

Luke Ritchie is a Ph.D. student at Arizona State University (ASU), Tempe. He received B.S. and M.S. degrees in Electrical Engineering from Arizona State University in 2003 and 2004, respectively. His research interests are in the areas of interaction between medium access control (MAC) and routing in mobile ad-hoc networks (MANETs) and wireless sensor networks (WSNs).

Hyo-Sik Yang is an assistant professor at Sejong University, Department of Computer Engineering, Seoul, Korea. Before he joined Sejong University, he was an assistant professor at Kyungnam University, Masan, Korea. He joined Sejong University on Fall 2006. He was a faculty research associate at Arizona State University (ASU), 2005. He received the B. E. degree in the Department of Information and Communication Engineering form Myongji University, Yongin, Korea, in 1998 and the M. S. and Ph. D. in Electrical Engineering from Arizona State University, Tempe, AZ, U.S.A., in 2000 and 2005, respectively.

His research interests are wavelength-division-multiplexing (WDM) all-optical networks, WDM packet switching, and WDM metropolitan area networks including node architecture, optimization, medium access control (MAC), traffic analysis, and routing in mobile ad-hoc networks (MANETs). He is also working on Substation Automation System (SAS) compying with IEC 61850 - communication networks and system in substations.