

For Members Only: Local and Robust Group Management in DTNs

Abstract

Effectively utilizing groups in delay tolerant networks (DTNs) can both improve the throughput of unicast routing protocols and open the door for a wide range of paradigms, such as anycast and multicast. Unfortunately, in DTN environments, there is no centralized entity that can quickly and reliably transmit group membership lists, and hence group information must be disseminated through unreliable and potentially malicious nodes. In this paper, we propose a local and robust group information dissemination and consolidation protocol, called *MembersOnly*, that both quickly and accurately transmits group membership information to all nodes in the network, even if multiple malicious nodes attempt to disrupt the process. We show via analysis and simulations that *MembersOnly* is able to withstand multiple types of attacks, with only very limited periods of vulnerability that disappear relatively quickly. This is in contrast to current techniques that cannot withstand many of these attacks, resulting in quick and thorough corruption of group membership lists. In addition, we show via simulation that even the most basic routing protocols can gain a performance advantage when using *MembersOnly*.

1 Introduction

Groups are naturally found in many environments. Such grouping is especially common in delay and disruption tolerant networks (DTNs), where much of the communication and mobility is based on human interaction [4, 8, 20]. In these environments, the composition of groups may be based on many different abstractions, including roles (i.e., firefighters or police officers [16]), social networks (i.e., friends communicating using wireless mobile devices [4]), or geographical closeness (i.e., coworkers who meet every day for meetings). Knowledge of such groups can greatly enhance many aspects of communication in DTNs, including unicast [10], anycast and multicast routing, information access control, and privacy. In particular, it has been shown that contacting a particular node's group or affiliation is an effective and efficient first step towards contacting the node itself [10]. However, due to the intermittently connected nature of DTNs, maintaining and disseminating such group information is challenging, especially in the presence of malicious attackers.

The successful enhancement of DTN communication through the use of group information requires that nodes throughout the network be aware of the membership lists for all groups. While such group membership management is not difficult in connected environments, heavy

partitioning and the lack of readily available end-to-end paths in DTNs break centralized membership services, just like they break traditional routing [11]. Instead, group information must be disseminated through the network using DTN-specific mechanisms that leverage contacts between nodes that meet. Unfortunately, the reliance on contact-based dissemination and the presence of malicious or faulty nodes may result in inaccurate knowledge of group membership. While cryptographic techniques (i.e., PKI [14]) could provide proof of group membership, such techniques are not currently feasible in the intermittently connected environment of DTNs due to the lack of availability of a trust anchor. The main challenge then lies in ensuring accuracy of disseminated group membership lists.

Group membership management in DTNs can be broken into four components: group creation, group information propagation, group information consolidation, and group-assisted routing. During *group creation*, members of a group learn about their membership in the group. At this point, nodes outside of the group are not aware of the group's membership, or perhaps even of the group itself. Once the group has been formed, group members *propagate* the group's membership list throughout the network. Simultaneously, faulty and malicious nodes may be propagating inaccurate membership lists. Non-member nodes collect and *consolidate* all group membership lists as they receive them, locally resolving any conflicting information about a group's membership list. Finally, the resolved group membership list can be used to support *routing* and other services in the DTN. While there exists some work on group creation and group routing, current approaches ignore propagation and consolidation issues and do not consider the presence of malicious nodes tampering with group membership information.

Current approaches to disseminating group membership information in DTNs epidemically [23] propagate the information and, working under the assumption of a completely trustworthy environment, do not have any mechanism to handle conflicting information [9, 10]. In stead of resolving any conflicting information, these approaches typically default to selecting the newest information as truth. Unfortunately, this simplistic approach is inappropriate in many DTN environments, especially in the presence of malicious nodes. Essentially, such approaches form inaccurate group membership information, and so, result in ineffective routing. We show that these approaches actually break down in the presence of

even a small number of malicious nodes. The main problem with these approaches stems from the lack of quick and robust propagation and consolidation.

The main contribution of our work comes from the design, analysis and evaluation of *MembersOnly*, our group membership management protocol for DTNs that enables accurate group membership dissemination, even in the presence of multiple malicious nodes, through effective distribution and consolidation of group membership lists. The main strength of *MembersOnly* comes from the lack of reliance on any type of cryptographic techniques, making it an appropriate system for networks where groups are quickly created and destroyed and where centralized authorities do not exist. Given a set of partially conflicting membership lists, *MembersOnly* builds off of techniques from data mining to establish a local view of group membership. Compared to current methods, *MembersOnly* provides more accurate local views during convergence and quickly converges to very accurate views. Given that the maintenance of group membership lists is susceptible to various types of attacks on the consistency of a group's membership list, we show both analytically and through simulations that *MembersOnly* provides accurate results, despite the presence of moderate levels of malicious nodes. Finally, we demonstrate the impact of accurate group membership information through the evaluation of a simple group-based routing protocol, showing that *MembersOnly* can significantly improve routing performance.

The remainder of this paper is laid out as follows. Section 2 provides an overview of groups in DTNs including related work in the area. Section 3 presents *MembersOnly* in-depth, along with the relevant mathematical intuition behind it. Section 4 provides a mathematical analysis of *MembersOnly* under multiple types of attacks, and give insights into how parameters should be tuned. Section 5 presents a comprehensive evaluation that compares *MembersOnly* to other currently popular protocols in both friendly and malicious environments, as well as explores its parameter space. Section 6 explores how *MembersOnly*, as opposed to currently popular but vulnerable systems, improves the routing performance of even the most basic routing protocols. Finally, we conclude in Section 7.

2 Groups in DTNs

Routing in DTNs is currently an active and difficult research problem [1, 2, 5, 7, 13, 15, 17, 21]. While leveraging group information in DTNs can have a significant impact on routing performance [10], current approaches consider all nodes in the network to be trustworthy, and so assume all group membership information to be trusted. Unfortunately, even a small number of malicious nodes can wreck havoc with membership lists if left unchecked. To understand the depth of this problem, in this section, we discuss the process of group membership management in DTNs given the four components: group creation, group information propagation, group information consolidation, and group-assisted routing.

2.1 Group Creation

Every group is formed based on some common feature of the individual nodes, including geography, node roles, and social affiliations. In some cases, groups are created in advance; in other cases, they are rather spontaneous. Take, for instance, a disaster scenario where nodes fall into roles such as firefighter, police officer, ambulance, and civilian. The nodes of one particular role could be thought of as a group. Another example is a group of nodes that are currently or often in close geographic proximity, and decide to form a group to better connect themselves to other nodes. This way, nodes wishing to contact a particular node in the group need only find a single member of the group, and be relatively confident that an intermediate node will meet the destination sometime in the future. Overall, the goal of group creation is to form such groups and allow all nodes of the group to know they are a part of it. Furthermore, group creation also informs group members of other nodes that are part of their group.

The specific algorithms used for group creation depend on the type of group. For example, geographic groups can be created using centralized algorithms for community detection [19] or distributed versions of centralized algorithms [10]. In geographic groups, group centrality is important, and approaches such as weighted network analysis [18] can be implemented in a distributed fashion [10]. On the other hand, some dynamic groups rely on physical contact and verbal agreement, and so group creation is mostly out-of-band. An interesting paradigm would be to create per-person social groups, where each person's group represents their social network. Everyone would also be a member of all of their friends' groups, resulting in a large number of overlapping groups that are each highly individualistic.

Although group creation is an interesting problem, in this paper, we focus primarily on role-based or geographic-based groups where nodes initially know which groups they are in and group membership does not change. However, our algorithms do support dynamic groups where members might not initially know all other members of the group and where group membership changes over time. To demonstrate this, we evaluate our algorithms during convergence, which gives us insight into how these algorithms behave in the face of dynamic groups.

2.2 Group Information Propagation

Once nodes know which groups they are in and who else is in these groups, they can start to propagate group membership information throughout the network. Such propagation enables nodes that are not a member of a group to gather information about the membership list of that group. Propagation is a key component of group membership management, since the consolidation algorithms discussed next calculate their membership lists based on the information collected during propagation. However, effective propagation faces two challenges: convergence speed and malicious nodes.

The quickest way to propagate group membership information is for all nodes to epidemically disseminate

all group information they are aware of. On the other hand, nodes could be more conservative and only disseminate a list of the groups to which they belong [9, 10]. While epidemic approaches are clearly optimal in terms of propagation speed, at least when resources are not limited, they are extremely vulnerable to many kinds of attacks. If nodes can send unrestricted amounts of information about any group, then malicious nodes can convince non-malicious nodes to send false information as well. This effectively increases the number of nodes spreading the false information in an unbounded fashion, making it difficult for any consolidation protocol to produce accurate results. Unfortunately, the more conservative approaches are very slow at propagating information across a network and, without appropriate consolidation, are still vulnerable to some types of attacks. We evaluate propagation speed for both of these approaches in Section 5 and discuss the propagation attacks that can compromise consolidation in Section 4.

Although the speed of epidemic approaches is highly desirable, we show that limiting the nodes to only sending information about groups that they are members of limits the spread of false information, providing a good basis for our consolidation protocol to achieve a high level of protection against malicious nodes. Furthermore, this membership-based propagation technique still disseminates information quickly.

2.3 Group Information Consolidation

Since group membership is propagated through the network, nodes collect multiple, potentially conflicting, pieces of information about each group. The nodes must consolidate this information into a single local view of each group’s membership list. This local view can then be utilized by the routing protocol for use with routing decisions. In DTN environments, there is very little applicable work on consolidating conflicting information. Therefore, we turn to the field of data mining and analysis. The TruthFinder system [25] solves a similar problem by first obtaining a set of “facts” about “objects” from multiple online servers, and then attempting to consolidate these potentially conflicting facts to determine the truth about the object. For example, to learn the true author list of a particular book, TruthFinder first contacts multiple servers, such as Borders and Barnes & Noble, and obtains an author list, or fact, about the particular book from each one. Using these facts, TruthFinder attempts to consolidate them into a single author list.

Unfortunately, TruthFinder cannot be easily adapted to fit a DTN environment. First, TruthFinder assumes that all of the servers, or fact providers, are always available and can be readily contacted, and hence gathers all facts before running the consolidation step. DTN nodes have at most only a very small subset of the entire network in communication range. Even if TruthFinder could be adapted to work in a disconnected environment, the algorithms in TruthFinder are designed to never omit any information from the final result (such as a particular author), and hence result in a relatively high false positive rate. For instance, a fact containing a single author pos-

itively reinforces a fact containing that author plus another author, and so facts with more authors tend to have more reinforcement. While acceptable for TruthFinder’s purposes, a malicious node should not be able to easily append itself to a membership list.

To support group information consolidation in DTN environments, we present an on-line algorithm that collects group membership information from each node it meets and consolidates it on-the-fly without requiring contact with any centralized server. Our algorithm continually refines its decisions as more information becomes available, quickly converging to very accurate decisions about group membership. The combination of this algorithm with our membership-based propagation approach enables successful defense against many types of attacks, even in the presence of a large number of malicious nodes.

2.4 Group-Assisted Routing

Accurate group membership information can be used for many services, but the most obvious is group-assisted routing. For example, BubbleRap [10] utilizes group membership information to improve standard unicast routing. In essence, BubbleRap attempts to transmit a message to nodes that are part of the message destination’s group, assuming that members of the same group have a high probability of contacting one another. Group information can also be used as a foundation for building anycast routing systems, where the goal is to reach at least one member of a particular group [6]. In the presence of faults and malicious users, the accuracy of the group membership information can have a large impact on the performance of any routing protocol.

Although we do not focus on routing protocols in this paper, we evaluate the effect of the accuracy of group membership information on a simple anycast routing protocol, which can be used as a building block for more advanced protocols. Essentially, more accurate membership information, even during convergence, significantly improves even a simple anycast routing protocol. As part of our future research, we are investigating the use of group membership information in unicast, multicast and anycast routing protocols, as well as other group-based services.

3 MembersOnly

The ability to quickly and accurately distribute group information opens up the door for many group-based services. In DTN environments, both unreliable links and malicious nodes make this a challenging problem. In this section, we present MembersOnly, a local and robust propagation and consolidation protocol that provides accurate views of groups even in the presence of malicious nodes. The propagation component, responsible for the quick delivery of information, is first explored followed by the consolidation component, responsible for merging the information obtained locally and filtering out malicious information.

3.1 Membership Dissemination

Effective propagation of membership information requires quick distribution while using mechanisms that support high integrity of the group membership information. To achieve this goal, MembersOnly takes a membership-based approach: nodes propagate group membership lists only for groups of which they are members. In other words, a set of membership lists, one for each group a node is a member of, is transmitted to every contact that node makes. This approach has two major benefits. First, it provides enough information for quick propagation, as opposed to transmitting only a list of groups the node is a member of. Second, it does not attempt to transmit everything, giving consolidation protocols the ability to filter out inaccuracies, as explained in Section 2.2.

Duplication is expected in DTNs, and hence nodes may receive multiple membership lists from the same node. Since membership lists are constantly evolving, the newest list should always be used. Furthermore, if a membership list is not replaced with a newer one from the same node for some time, it is possible to assign a weighting factor to account for aging. We leave the aging of membership lists for future work.

The membership-based propagation approach of MembersOnly, in conjunction with its consolidation protocol, which we describe next, enables quick and accurate group information distribution, even in the presence of multiple malicious nodes.

3.2 Consolidation of Membership Lists

While MembersOnly enables fast propagation of membership lists, its true power lies in its ability to filter out malicious information. At any given time, a node has a set of, potentially old, membership lists for some or all groups in the network. MembersOnly leverages the availability of these multiple lists to build confidence levels for each potential member of a group, enabling well-informed consolidation that filters suggestions from malicious nodes, as illustrated in Figure 1. The goal of the consolidation component is to locally create a single high-confidence membership list for each group in the network. This high-confidence list contains members that the node believes are really part of the group.

During consolidation, MembersOnly looks at all recommendations about membership for each potential member of a group and computes a confidence score about that member. This score is based on *positive* evidence extracted from all membership lists that claim that the node is part of the group and *negative* evidence extracted from all membership lists that do not have that node listed as part of the group. To reconcile these differences of opinion, MembersOnly calculates a function of the difference between the strength of the positive evidence and the strength of the negative evidence. If the result from this function is greater than a threshold, the node is placed on the high-confidence list for that group. Since malicious nodes may wrongfully inflate both the positive and the negative evidence, this function must be designed to filter out malicious evidence.

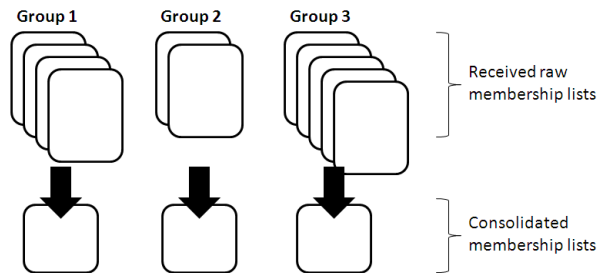


Figure 1. Consolidation Component

The MembersOnly consolidation component achieves high-confidence membership lists as follows. Each node n collects a set of membership lists, L , about a group G . Each entry on a membership list looks like “node m is a member of group G ”. Node n then computes a list, H , that contains, from n ’s point of view, all high confidence members of group G . To determine the confidence of a node m ’s membership in G , MembersOnly creates two subsets of L , L_m and $L_{\bar{m}}$ as follows:

$$L_m = \{l \in L | m \in l\},$$

$$L_{\bar{m}} = \{l \in L | m \notin l\},$$

where $X_m = |L_m|$, the total number of lists m is found on, provides the positive evidence for believing that m is a member of G and $X_{\bar{m}} = |L_{\bar{m}}|$, the total number of lists m is not found on, represents the negative evidence.

Node n can now combine this positive and negative evidence to determine a confidence value about m ’s membership in G . Intuitively, the confidence can be captured by the difference between functions of the positive and negative evidence. To capture this, we build off of techniques used in data mining [25], where confidence should start low and quickly rise only after enough supporting evidence is obtained. This resulting S-shaped curve can be generalized by the popular Sigmoid Function [24], illustrated in Figure 2, and defined as

$$Y = \frac{1}{1 + e^{-X}}.$$

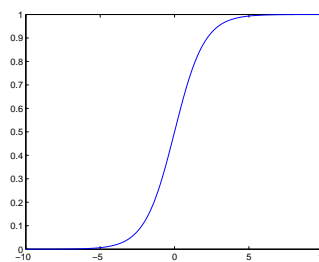


Figure 2. Standard Sigmoid Function

Applying the Sigmoid function, the strength of the

positive evidence for node m can be computed as:

$$\frac{1}{1 + e^{-(X_m - \alpha)}},$$

and the strength of the negative evidence can be computed as:

$$\frac{1}{1 + e^{-(\tau X_m - \alpha)}}.$$

To support flexibility in our model, we have added the parameter α to generalize the standard Sigmoid function. Changing α shifts the function along the x-axis, such that $Y = 0.5$ when $X = \alpha$. Furthermore, we add a weighted factor, τ , to the negative evidence. The effect of these parameters on the resulting confidence levels are described in detail shortly.

The total confidence about a node m 's membership in group G can be found by taking the difference between the positive and negative evidence. Since it is unclear what negative confidence means, we ensure that the confidence does not fall below 0.

$$C(m) = \max \left\{ \frac{1}{1 + e^{-(X_m - \alpha)}} - \frac{1}{1 + e^{-(\tau X_m - \alpha)}}, 0 \right\}$$

$C(m)$ gives an indication of how confident node n should be in node m 's membership. After node n computes this value for all possible members of a group, it selects the high-confident nodes to be part of the consolidated list H :

$$H = \{m | C(m) \geq \gamma\},$$

where γ is a system defined parameter that determines an accuracy threshold for the system. Essentially, if the difference between the strength of the positive evidence and the strength of the negative evidence is greater than the threshold γ , the membership in question is accepted. This process is repeated for all groups the node is aware of, and results in a consolidated list H for each group. The final set of H values can then be passed to the routing protocol.

Since the lack of a member in a valid list can be used to counter the presence of a member in a malicious list, the parameter $\tau \in [0, 1]$, is used as a weighting factor for the negative evidence. Essentially, τ scales up or down the amount of negative evidence needed to counter positive evidence. If τ is close to 1, then negative evidence is given the same importance as positive evidence, and hence, less negative evidence is needed to doubt a node's membership. On the other hand, if τ is close to 0, then more negative evidence is needed to cast doubt. Setting τ properly impacts how a system reacts to different types of attacks. In Section 4, we explore, via mathematical analysis, how different types of attacks can affect the system, and how to appropriately set τ to counter them.

Another important parameter is α . The smaller α is, the faster the propagation speed is when there are no attackers in the network, because a smaller amount of positive evidence is needed to reach γ . However, as α gets

smaller, it has a negative impact on the appropriate setting of τ , which is detrimental to the system in regards to attacks. We recommend setting α to around the square root of the group size, since this supports relatively fast propagation, while being large enough to not negatively impact τ . The interplay between these two parameters is explored in Section 4.

The power of this model is that it is highly parameterized and hence compatible with a large range of group sizes and characteristics. We explore many of these parameters and how they effect system performance throughout the rest of this paper.

4 Model Analysis with Attackers

The primary goal of MembersOnly is to provide quick and accurate group information to all nodes in the network, even in the presence of multiple malicious nodes. In this section, we present two high level classes of attacks and show, via mathematical analysis, how MembersOnly can be set to defend against them.

4.1 Potential Attacks

We assume malicious nodes have similar abilities to normal nodes, in that they can send and receive any information they want during a contact. We now briefly describe two generic attacks, an addition attack and a deletion attack, that give good insight into how attackers can affect and exploit different systems.

The goal of the *addition attack* is to convince as many nodes as possible that the attacking node is part of some or all groups in the network. Therefore, attackers must convince normal nodes that they are valid entries on the local membership lists for those groups. If successful, attackers will be members of many groups and will be considered good intermediate nodes for routing to those groups. This positions the attackers to launch powerful black hole attacks, or other more sophisticated attacks. To demonstrate the effect of this attack, in Section 5 we instantiate a version of the addition attack where each attacker appends itself to all membership lists, and transmits this new information during contacts.

The goal of the *deletion attack* is to convince as many nodes as possible that one or more specific valid members of a group, or multiple groups, are in fact not members. Attackers must therefore provide enough negative evidence about a node to cast doubt about it's membership in a group. Deleting members from membership lists can severely hinder routing performance. Essentially, a denial-of-service attack is launched, since nodes hold data until they meet a member of a particular group. To demonstrate the effect of this attack, in Section 5 we instantiate a version of the deletion attack where attackers simply broadcast blank membership lists for all groups they are currently aware of.

These two types of attacks represent building blocks for more devastating attacks. Therefore, we evaluate how well MembersOnly and other current systems react to and handle these types of attacks. Note that we do not allow malicious nodes to perform Sybil attacks. If we gave them this capability, they could constantly send multiple

membership lists for the same group under fake names, artificially inflating the chance of a node believing that at least one of the fake names is really on the list. Solving the Sybil problem in delay tolerant networks, without a centralized system, is a very difficult problem that we do not attempt to solve with this work.

4.2 Model Analysis with Attackers

Given that malicious nodes have the ability to perform both addition and deletion attacks, we now perform an analysis of our model to determine how best to defend against these attacks. Recall that if

$$\frac{1}{1 + e^{-(X_n - \alpha)}} - \frac{1}{1 + e^{-(\tau X_m - \alpha)}} \geq \gamma$$

then node n is confident to add m to the high confidence list H for group G . Also recall that if malicious nodes perform a deletion attack, their goal is to inflate the negative evidence against m enough to drop the confidence value below γ . Hence, to protect against this attack, τ , the negative evidence weighting factor, should be decreased. This gives less weight to the false negative evidence the attackers are providing. In contrast, if malicious nodes perform an addition attack, their goal is to inflate the positive evidence for their own false information, to bring the confidence value above γ . To protect against this, τ should be increased, so true group members can provide the necessary negative evidence against the false positive evidence.

It is clear that if a system only wishes to defend against deletion attacks, τ should be 0. Similarly, if a system only wishes to defend against addition attacks, τ should be 1. However, it is possible to set τ to defend against both addition and deletion attacks simultaneously by keeping it within a valid range. Larger ranges of τ are best, since this gives more flexibility in the actual choice for τ for a given system. To find the outer limits of this range, we analyze the steady state case, when every node has met every other node, to ensure that both types of attacks are, in the long run, completely defeated. We assume, for simplicity, that attackers cannot convince actual group members of changes in their own groups. While in practice this may not be true, particularly if nodes can join and leave groups without informing all group members of the action, it provides a good approximation.

In the MembersOnly group information propagation component, the only way a node can obtain information about a group is to meet a member of that group (or, at least a node that claims to be a member of that group). Given M , the total number of true members of a particular group, and A , the total number of attackers in the network attacking that group, the total amount of possible evidence for or against a node's membership is $M + A$.

For a deletion attack in the long run, nodes outside of the group obtain A recommendations *against* and M recommendations *for* the node in question. To protect against this attack, the confidence value computed by the non-member node should be greater than γ . In other words,

$$\frac{1}{1 + e^{-(M - \alpha)}} - \frac{1}{1 + e^{-(\tau A - \alpha)}} \geq \gamma.$$

Solving for τ , we find that to protect against the deletion attack,

$$\tau \leq \frac{1}{A} \cdot \left[\alpha - \ln \left(\frac{-(\gamma + e^{\alpha - M} + \gamma \cdot e^{\alpha - M})}{\gamma + \gamma \cdot e^{\alpha - M} - 1} \right) \right].$$

Now consider attackers launching an addition attack against a particular group, where the goal is to get non-members to believe that the malicious nodes are actually part of the group. In the long run, the nodes outside of the group will have, in the worst case, A recommendations for the entries and M recommendations against them. This is analogous to the previous inequality, and hence to protect against the addition attack,

$$\tau \geq \frac{1}{M} \cdot \left[\alpha - \ln \left(\frac{-(\gamma + e^{\alpha - A} + \gamma \cdot e^{\alpha - A})}{\gamma + \gamma \cdot e^{\alpha - A} - 1} \right) \right].$$

It is immediately clear that if the number of attackers is greater than the number of nodes in the group, MembersOnly cannot defend against both types of attacks simultaneously. In this case, the user would have to choose which attack they were most concerned about, and adjust τ accordingly.

To clarify, consider the following example, which we also use for our evaluations. Assume a group of size of $M = 45$ and $\gamma = 0.75$.

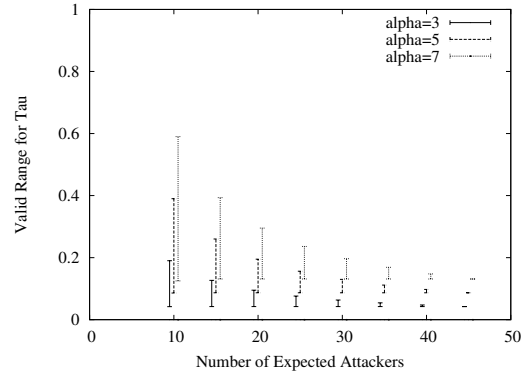


Figure 3. Valid ranges for τ

When choosing τ , it is important to choose a value that is within the valid range for the maximum expected number of malicious nodes in the network, to ensure that the system is within the valid τ range at all times. Figure 3 shows the valid ranges of τ as the number of attackers varies from 10 to 45, as indicated by our previous analysis. This figure shows that MembersOnly is able to defend against both addition and deletion attacks at the same time, as long as the number of attackers is less than the group size of the group in question. However, as the number of malicious nodes increases, the valid range

of τ shrinks. It is also interesting to note that as α decreases, the ranges become smaller, which is undesirable. However, as α increases, the propagation speed increases, which is desirable. We recommend setting α to be around the square root of the group size, as this allows for both quick propagation speeds and large ranges of τ .

It is important to remember that larger ranges for τ are beneficial since they allow for more leeway for errors in predictions of the number of malicious nodes and the group size for the particular group in question. Furthermore, larger ranges make it easier to defend against the attacks in practice, where a group’s members themselves may be swayed by malicious evidence. We use the values of τ derived from this analysis in the following evaluation section to show that MembersOnly can defend against both types of attacks in practice.

5 Evaluation

The goal of our evaluation is two-fold. First, we evaluate the propagation speed and attack resistance of MembersOnly in comparison to existing approaches and show that MembersOnly enables fast propagation and is extremely resistant to attacks, even in the presence of multiple malicious nodes. Second, we evaluate the effect of the parameters τ and α on the behavior of MembersOnly.

5.1 Evaluation Setup

For comparison, we use the two propagation approaches discussed in Section 2: *CopyMyGroups*, where nodes transmit a list of every group they are members of to every contact they meet, and *CopyEverything*, where nodes transmit all group membership information they know to every contact they meet. For both of these protocols, the consolidation component is to simply take the newest version of any membership information as truth.

Average group completion percentage captures the speed and pervasiveness of group membership list propagation by tracking the completion percentage of a group over all nodes and all groups. Note that all nodes have access to an oracle with all correct group membership lists strictly for the purpose of metric computation. This metric will increase as soon as any node becomes aware of any subset of members for any group. The higher the metric’s value is, the faster the system is at propagating group information. Both the normal propagation speed and the deletion attack effectiveness are measured using this metric. It is appropriate for the deletion attack, since goal of attackers is to delete as many members from every local membership list as possible, hindering propagation.

For the addition attack, the *average percentage of corrupt groups* captures how corrupt local membership lists are. A conservative approach is taken to say that a node’s view of a group membership list is corrupt if that node (falsely) believes at least one attacker is actually a member of the group. It is the attackers’ goal to corrupt as many groups as possible, driving the metric up. Hence, the lower the metric’s value is, the better the system is at protecting against the addition attack.

All metrics are evaluated over time. Each of the evaluated systems eventually converges to either 0 or 1 for

all metrics, and therefore it is most interesting to see how the curves progress over time, and how they look relative to one another. The exact time values are not as important as the characteristics of the curves, since these values change with properties of the network such as movement speed, transmission range, number of nodes, etc. Essentially, even though attackers may lose out in the end, there can be periods of vulnerability where attackers can make gains.

All simulations use the ONE [12] simulator and the random waypoint model [3], with nodes moving between 3 and 7 meters per second. There are a total of 250 nodes divided into 5 non-overlapping groups of either 50 nodes (if there are no attackers), 47 nodes (if there are 15 attackers), or 45 nodes (if there are 25 attackers). The transmission range of each node is 250m and the world size is 3.5 km x 3.5 km. All data points are the average of 10 runs with 95% confidence intervals surrounding them. There are data points every 50 simulation seconds; however, many markers have been omitted for clarity. For all simulations, $\gamma = 0.75$.

5.2 Comparative Evaluation

To understand the impact of the propagation algorithm, MembersOnly is compared to both *CopyMyGroups* and *CopyEverything* by looking at membership list propagation speed as well as the effectiveness of the addition and deletion attacks. The number of malicious nodes, if any, in the simulations are denoted by parentheses next to the system name. The two numbers next to MembersOnly represent the parameters α and τ . As previously described, $\alpha = 7$ for these simulations, which constrains the choice of τ from around 0.13 to around 0.24, which handles up to 25 attackers. Therefore, we chose $\tau = 0.2$.

Propagation algorithms aim to spread membership lists throughout the network. As expected, *CopyEverything* shows the optimal speed since it epidemically disseminates all information (see Figure 4(a)). Virtually all nodes are correctly aware of all membership lists in around 250 seconds. In contrast, *CopyMyGroups*, is relatively slow since it only transmits a list of groups a node is a part of during each contact, not a membership list of those groups. Therefore, to reach 100%, every node would have to come in contact with every other node. By the end of the simulation, at 5,000 seconds, this approach reaches only around 85% completion. MembersOnly, which transmits group membership lists for all groups a node is a part of, starts off slightly slower than the other systems since, for security reasons, it waits for sufficient evidence before accepting information. However, after a sufficient amount of evidence is collected, nodes propagate the information very quickly.

Once attackers are introduced into the system, it is interesting to consider the average percentage of corrupt groups. For the addition attack (see Figure 4(b)), the higher the percentage, the more penetration the attackers gain, and hence the less resistant the system is to the attack. *CopyEverything* is slightly worse than *CopyMyGroups*. However, both are terrible at resisting the addi-

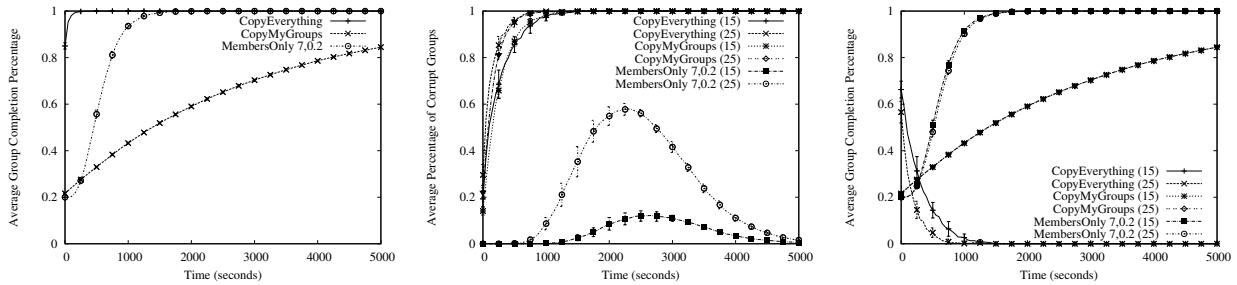


Figure 4. (a) Group Completion, (b) Addition Attack, (c) Deletion Attack

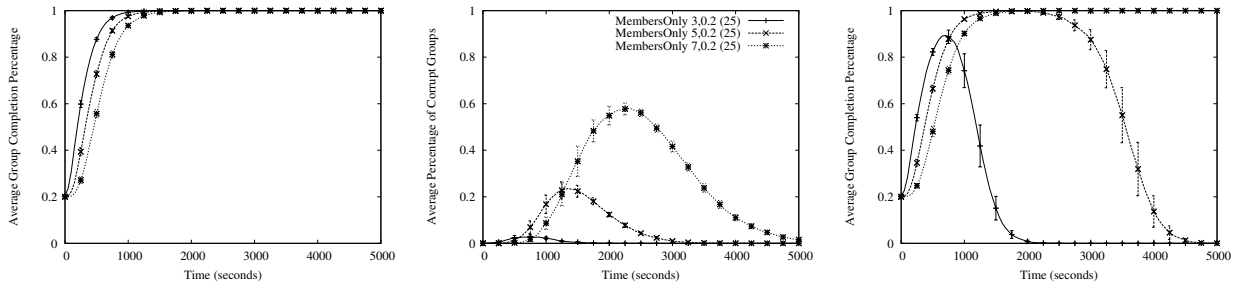


Figure 5. (a) Group Completion, (b) Addition Attack, (c) Deletion Attack

tion attack since the attack nodes persistently claim to be part of every group they know of. When a node meets enough attack nodes, it is convinced that at least some of the attack nodes are part of the group. This node then propagates that false information, convincing other nodes to do the same. This degenerating process is quite fast for both systems. In contrast, *MembersOnly* is more careful and considers the absence of information (i.e., a membership list without an attacker on it) as evidence against that information. Hence, the attackers can gain a temporary advantage with some nodes. However, in the long run, the attackers will not be able to overcome the honest nodes. The period of time where the metric is non-zero is a vulnerability period where some nodes wrongfully believe attackers are part of a group. As expected, the duration and prominence of this period increases as the number of attack nodes increases. However, even with 25 attackers, *MembersOnly* keeps the vulnerability period limited, and eventually the percentage goes to zero.

Finally, in the deletion attack, the attackers try to disrupt the propagation of group membership lists. With *CopyEverything*, membership lists quickly propagate and some gains are made (see Figure 4(c)). However, attackers continuously promoting blank membership lists eventually cause a larger and larger number of nodes to believe that the membership list is actually blank. This results in the percentage going to zero, indicating the attack was successful. Interestingly, in *CopyMyGroups* the attack is not only unsuccessful, but useless since *CopyMyGroups* is immune from this attack because only a list of groups is propagated, never a membership list of those groups. Hence, there is no way for an attack node to convince another node of any membership list, let alone a blank one. The result of the attack on *MembersOnly* is simply

a shift in time of the curve. The attackers are able to convince nodes to delay accepting membership lists as true for some time. However, eventually nodes running *MembersOnly* receive enough evidence from honest nodes to override the evidence given by attack nodes. Therefore, attackers simply delay the inevitable.

In summary, although *CopyEverything* is extremely fast, it is also extremely susceptible to both addition and deletion attacks. While *CopyMyGroups* is immune from deletion attacks, it is very susceptible to addition attacks and also too slow for practical use. In comparison, *MembersOnly* is both resistant to addition and deletion attacks, and can propagate group membership at a quick speed.

5.3 Parameter Evaluation

The parameters of *MembersOnly* determine both the propagation speed and resiliency to attack. Particularly, there is an interesting interplay between α and τ , which was explored in the analysis from Section 4. Recall, that as α decreases, the propagation speed should increase, since nodes can more quickly reach the threshold γ . However, as the analysis shows, this also decreases the valid range of τ . If τ is too low, addition attacks will succeed, and if τ is too high, deletion attacks will succeed.

When there are no attackers in the network, smaller values of α result in faster group information propagation due to the a smaller amount of positive evidence required to reach γ (see Figure 5(a)). With addition attacks, *MembersOnly* successfully defends against the attack for all values of α (see Figure 5(b)), because $\tau = 0.2$, which is always greater than the minimum τ value, according to the model. Recall that lower values of α actually drop the range numerically, while shrinking it, and hence lower values of α , for the same τ , result in *better* protection against an addition attack. However, in the event of a

deletion attack, the value of τ is greater than the maximum τ value for $\alpha = 3$ and $\alpha = 5$, according to the analysis. Essentially, *MembersOnly* cannot defend against the deletion attack for these two values of α , while it can for $\alpha = 7$ (see Figure 5(c)).

6 Group-based Routing

Obtaining quick and accurate group information about a network opens the door for a wider range and greater efficiency of routing protocols. One immediate result is the ability to efficiently perform anycast routing, which attempts to deliver a message to at least one member of a particular group [6]. Anycast routing is useful as a stand-alone routing technique for many scenarios. For instance, in emergency response networks, it may be more beneficial for an injured person to contact any emergency responder rather than a particular one. It is also useful as a means to improve unicast routing by first anycasting a message to a member of the destination’s group, and then unicasting it from there.

To understand the impact of the accuracy of group membership on routing protocols, we evaluate a basic single-copy anycast routing protocol that utilizes group information to make routing decisions. We show that routing protocol performance is very dependent on the underlying group membership management. By using *MembersOnly*, as opposed to current popular systems, routing performance can be improved by close to 8% under certain attack scenarios. For simplicity, a single-copy protocol was implemented, where replicas of messages are not created [22], and hence resource management is less important. This basic protocol acts as a building block for more advanced anycast routing techniques.

6.1 Anycast Routing and Attacks

One prominent building block for routing in DTNs is *direct delivery*, where a node simply holds a message until it comes in contact with the destination of that message. This building block allows for a store-and-carry approach to DTN routing and can even act as a very basic stand alone protocol. Similarly, a popular building block for anycast routing is to perform direct delivery to *destination groups* instead of destination nodes. This very simple protocol stores and carries all messages, which are destined for a group rather than a node, until it meets a target group member. Since the message was successfully delivered (at least in the eyes of the deliverer) to the group, delivery is consider successful.

This group-based anycast routing protocol is reliant on the underlying group system for quick and accurate group information and so it is interesting to see how malicious nodes spreading bad group information affect the performance of the protocol. The malicious nodes attempt to perform multiple *black hole* attacks, where the goal of each is an aggressive addition attack to get on as many membership lists as possible. By getting on multiple membership lists, attackers are then able to intercept and drop messages destined for those particular groups.

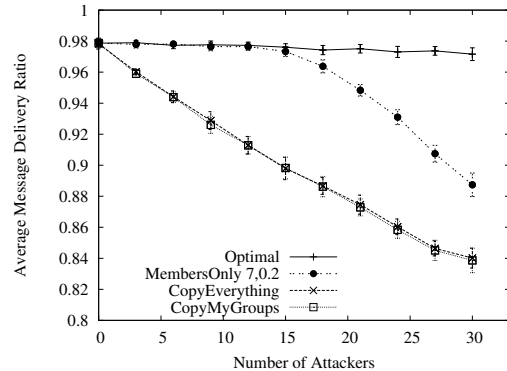


Figure 6. Routing Performance

6.2 Performance Comparisons

The goal of this evaluation is to see how different group membership management approaches affect the performance of anycast routing under attack scenarios. We implemented the basic anycast routing protocol described above in the ONE simulator. For group membership management, we evaluate the performance of the anycast routing protocol using each of the following approaches: *MembersOnly*, *CopyEverything*, and *CopyMyGroups*. Additionally, we implemented an oracle module to provide a baseline that gives the routing protocol perfect group information at all times.

All simulations were done using the same parameters as before, except with a total of 150 nodes and groups of size $50 - A/3$, where A is the number of attackers. Messages are sourced from random non-malicious nodes and are destined for a particular group, ensuring that the group is different from the group of the message source. Every node sources a single message at a random time during every 200 second interval. Each message is 50kB in size, and buffers are large enough to hold all messages. Message delivery ratio, the metric used, is the total number of messages successfully delivered over the total number of messages sourced in the network. Every data point is the average of 10 runs.

By utilizing *MembersOnly*, the group-based anycast routing protocol achieves an approximately 8% over current group approaches during some attack scenarios (see Figure 6). Furthermore, since the vulnerability window is relatively small during a low to moderate attack level, malicious nodes had trouble getting on more than a few local membership lists. Hence, *MembersOnly* performs very close to optimal much longer than other protocols in this environment. Conversely, in both *CopyEverything* and *CopyMyGroups*, attackers are quickly able to compromise many membership lists, and hence significantly hurt routing performance.

7 Conclusions and Future Work

Groups are inherent to DTNs, where much of the communication and mobility patterns are based on human interaction and social paradigms. Utilizing grouping information can enhance communication in these environ-

ments [10]. Unfortunately, due to the high level of disconnectivity and untrustworthy environments, distributing group membership information is a difficult problem. In this paper, we have presented *MembersOnly*, a local and robust group propagation and consolidation protocol that both quickly and accurately distributes group membership lists, even in the presence of multiple malicious nodes. We have shown via analysis and simulation that *MembersOnly* can withstand multiple types of attacks while still delivering membership lists quickly. Finally, we have shown that even the most basic routing protocol can gain a performance advantage when using *MembersOnly* over existing protocols.

In the future, we plan to extend our research of groups in DTNs in three directions. First, we plan to investigate automated group creation. By analyzing trends in mobility and communication patterns, many types of groups, including social and geographic, can be automatically detected. This helps minimize user interaction and, when used in conjunction with *MembersOnly*, gives routing protocols more forwarding options. Second, to improve the attack resistance of *MembersOnly* even further, we plan to utilize past information to help detect and limit malicious behavior in a network. For instance, if a node continuously disagrees with the vast majority of other nodes, this should raise suspicion and any information that node propagates should be weighted accordingly. Third, we plan to investigate more advanced group-based routing, including unicast, anycast, and multicast.

8 References

- [1] A. Balasubramanian, B. Levine, and A. Venkataramani. DTN routing as a resource allocation problem. In *Proc. ACM SIGCOMM*, 2007.
- [2] J. Burgess, Br. Gallagher, D. Jensen, and B. Levine. MaxProp: Routing for vehicle-based disruption-tolerant networks. In *Proc. IEEE INFOCOM*, 2006.
- [3] T. Camp, J. Boleng, and V. Davies. A survey of mobility models for ad hoc network research. *WCMC: Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications*, 2:483–502, 2002.
- [4] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott. Impact of human mobility on opportunistic forwarding algorithms. *IEEE Transactions on Mobile Computing*, 6(6):606–620, 2007.
- [5] V. Erramilli, M. Crovella, A. Chaintreau, and C. Diot. Delegation forwarding. In *ACM MobiHoc*, 2008.
- [6] Y. Gong, Y. Xiong, Q. Zhang, Z. Zhang, W. Wang, and Z. Xu. Anycast routing in delay tolerant networks. In *IEEE Global Telecommunications Conference*, 2006.
- [7] D. Henriksson, T. Abdelzaher, and R. Ganti. A caching-based approach to routing in delay-tolerant networks. In *ICCCN*, 2007.
- [8] P. Hui, A. Chaintreau, J. Scott, R. Gass, J. Crowcroft, and C. Diot. Pocket switched networks and human mobility in conference environments. In *ACM SigComm workshop WDTN*, 2005.
- [9] P. Hui and J. Crowcroft. How small labels create big improvements. In *Proc. of IEEE ICMAN*, 2007.
- [10] P. Hui, J. Crowcroft, and E. Yoneki. Bubble rap: social-based forwarding in delay tolerant networks. In *Proceedings of ACM MobiHoc*, 2008.
- [11] S. Jain, K. Fall, and R. Patra. Routing in a delay tolerant network. In *Proc. ACM SIGCOMM*, 2004.
- [12] A. Keränen, J. Ott, and T. Kärkkäinen. The ONE Simulator for DTN Protocol Evaluation. In *Proceedings of SIMUTools '09*, 2009.
- [13] A. Lindgren, A. Doria, and O. Scheln. Probabilistic routing in intermittently connected networks. In *MobiHoc 03*, 2003.
- [14] A. Menezes, P. van Oorschot, and S. Vanstone. Handbook of applied cryptography, 1996.
- [15] M. Musolesi, S. Hailes, and C. Mascolo. Adaptive routing for intermittently connected mobile ad hoc networks. *Proceedings of IEEE WoWMoM*, 2005.
- [16] S. Nelso n, A. Harris, and R. Kravet s. Event-driven, role-based mobility in disaster recovery networks. In *Proceedings of ACM CHANTS*, 2007.
- [17] S. Nelso n and R. Kravet s. Encounter-based routing in dtns. *Proceedings of IEEE InfoCom*, 2009.
- [18] M. E. J. Newman. Analysis of weighted networks. *Physical Review E*, 70:056131, 2004.
- [19] G. Palla, I. Dernyi, I. Farkas, and T. Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, pages 814–818, 2005.
- [20] J. Scott, P. Hui, J. Crowcroft, and C. Diot. Hagggle: A networking architecture designed around mobile users. In *WONS*, 2006.
- [21] T. Spyropoulos, K. Psounis, and C. Raghavendra. Spray and wait: An efficient routing scheme for intermittently connected mobile networks. In *Proceedings of ACM WDTN '05*, 2005.
- [22] T. Spyropoulos, K. Psounis, and C.S. Raghavendra. Single-copy routing in intermittently connected mobile networks. In *(IEEE SECON)*, 2004.
- [23] A. Vahdat and D. Becker. Epidemic routing for partially connected ad hoc networks. Technical Report CS-2000-06, Department of Computer Science, Duke University, April 2000.
- [24] D. von Seggern. *CRC Standard Curves and Surfaces with Mathematics*, 2nd ed. CRC Press, 2007.
- [25] X. Yin, J. Han, and P. Yu. Truth Discovery with Multiple Conflicting Information Providers on the Web. In *IEEE Transactions on Knowledge and Data Engineering*, pages 796–808, November 2008.