Dandelion: Mobility-assisted Reliable Message Propagation Protocol in MANETs

Wenbo He Ying Huang Klara Nahrstedt
University of Illinois at Urbana Champaign
Urbana, IL 61801-2302

Whay C. Lee
Motorola Labs
Marlborough, MA 01752

***Abstract*— Reliable message propagation is an important means of communication in mobile ad hoc networks (MANETs), and serves as a fundamental component for various applications, such as mobile advertising, inter-vehicle safety message propagation, alert propagation to defend against cyber-attacks in MANETs. Reliable message propagation requires that a message can be delivered to a large percent (e.g. 90%) of network nodes even under the presence of malicious/selfish nodes and intermittent network partitions. Inspired by the proliferation of dandelion in botanic world, we propose a reliable epidemic routing protocol for message propagation, called *Dandelion*. The *Dandelion* protocol relies on node mobility and periodic retransmission of a message. Comparing with flooding-based epidemic routing protocol, our mobility-assisted reliable epidemic routing protocol demonstrates excellent reliability, efficiency, and robustness.**

I. Introduction

Due to the proliferation of mobile devices, such as PDAs, laptops, sensors, cell phones, there is a growing demand for reliable message propagation protocols in mobile ad hoc networks (MANETs) for imminent commercial purposes. For example, in *Mobile Advertising* applications, department stores/service providers/sellers desire to advertise goods/immediate sales across customers who carry mobile devices. The goal of advertisement is to reach largest possible target population in ad hoc manner. The ad hoc mobile advertising platform requires that the relevant messages can be distributed in an inexpensive and efficient manner.

In the context of mobile ad hoc networks, we need to fully exploit node mobility to design an efficient, reliable and robust message propagation protocol. It has been proved that mobility can increase network capacity and overcome network partitions. However, the design of reliable and robust message propagation protocol is very challenging. First, it is desired to propagate a message throughout a network with minimal resource consumption. For more general scenarios, mobile nodes should operate without actively pulling neighborhood/topology information or controlling mobility of network nodes. Second, there can be small portion of malicious/selfish nodes which may not comply with the protocol specification. Under the presence of malicious/selfish nodes, a reliable message propagation protocol should still be able to deliver a message to large population of network nodes. In this paper, we propose the *Dandelion* protocol for message propagation, and demonstrate that the *Dandelion* protocol is efficient, reliable and robust.

Our solution is inspired by the wide spread of dandelion seeds. When a puff ball bursts, dozens of floating seed-bearing parachutes fly with wind across its neighborhood, and may travel a long distance before landing. In this way, dandelion flowers proliferate everywhere. We borrow this idea to achieve efficient message propagation at the cost of the acceptable delivery latency. In our mobility-assisted message propagation protocol, messages are carried by mobile nodes, which mimics that dandelion seeds are carried by floating parachutes. When the parachutes travel a certain distance, the dandelion seeds are brought to a new neighborhood. As dandelion seeds landed on ground, they grow and further propagate later on. Similarly, due to the node mobility, mobile nodes bring the messages to a new neighborhood within a certain period of time. When new neighbors hear the message, they will further propagate the message. Our *Dandelion* message propagation protocol relies on periodically retransmission of messages by mobile nodes.

The key design issue is when to terminate the propagation of a message, so that the message is able to reach all the network nodes without consuming extra bandwidth. We adopt a parameter *times-to-send (TTS)* to control the termination of message propagation. A *times-to-send (TTS)* field is included in the header of a message. In each step, when a mobile node retransmits a message, *TTS* value is reduced by one. When *TTS* reaches zero, a node stops forwarding the message. Hence, the message propagation procedure is terminated. In this paper, we show that if we design the initial *TTS* value properly, a message can be disseminated to almost all network nodes with a high probability, but any smaller *TTS* value may not attain such high coverage of message delivery. We demonstrate that the *Dandelion* protocol is reliable and efficient through both analysis and simulations. We will also show that the *Dandelion* protocol is able to tolerate a non-trivial portion of selfish nodes, and moreover it is robust to malicious attackers which attempt to modify *TTS* values during message propagation.

The rest of the paper is organized as follows: Section II summarizes the related work. Section III describes the *Dandelion* message propagation protocol. Section IV addresses the selection of parameters in the *Dandelion* protocol. Section V discusses the robustness of the protocol under the presence of selfish and/or malicious nodes. Section VI evaluates the performance of the *Dandelion* protocol by simulations. We conclude the paper in Section VII.

II. Related Work

Flooding-based protocols [1], [2], [3], [4]are designed for message propagation, where every node broadcasts every message once. Due to message redundancy, flooding protocols consume scarce bandwidth resources, lead to heavy contention, and cause packet loss over wireless links. Another class of broadcasting protocols utilizes node mobility to achieve high coverage [5] [6] [7].

It has been proved that mobility can increase network capacity [8] and overcome network partitioning [9] [10] [11]. Mobility-assisted routing schemes trade off delay for capacity and/or efficiency. Along this direction, various trade-offs between capacity and delay have been explored [12], [13], [14], [15].

Recently, mobility-assisted routing schemes have attracted much research effort in delay-tolerant networks (DTNs), where mobility is a necessary component of the routing functionality. A comprehensive overview of mobility-assisted routing schemes is provided in [16]. An epidemic routing scheme [17] has been proposed for message propagation. To avoid unnecessary message transmission, a *summary vector* is used to indicate whether or not a given message is in the local buffer of a certain node. When two nodes come into the communication range of each other, they exchange their *summary vectors*. Hence, each node knows which messages it has not yet received. Then the two nodes exchange messages accordingly. The protocol based on *summary vectors* is designed for partially connected networks, which are usually sparse networks. A reasonably dense mobile ad hoc network can be overloaded when many nodes exchange pairwise *summary vectors*. In [18], authors show that the *encounter-based* broadcast protocol achieves very high coverage (close to 100%) of message delivery, at the cost of extra message overhead at $O(n \ln n)$ level. To reduce the message overhead, *encounter-based* gossip scheme is proposed in [19]. The scheme requires each node to maintain an *encounter-list*, which stores the *ID*s of all nodes it has encountered since the first reception of a given message. The overhead to update *encounter-list* can be large too.

There are several key questions to answer in mobility-assisted epidemic routing protocols, but has been overlooked, such as (1) How to control message overhead to use minimum transmissions while achieving high message delivery coverage? (2) What is the delay to propagate a message throughout the network? We answer these questions in this paper, and present an efficient, reliable and robust mobility-assisted epidemic routing protocol.

III. Dandelion: A Mobility-assisted Epidemic Routing Protocol

In this section, we will introduce a mobility-assisted epidemic routing scheme for message propagation.

A. Assumptions

To make our design fit in a wide range of mobile ad hoc network systems, the *Dandelion* protocol does not require network nodes to maintain neighborhood information. Either, the protocol does not rely on any specific network model and mobility pattern. We believe that with further information on network characteristics and mobility pattern, the *Dandelion* can be easily adapted to specific environment to achieve better performance. To set up context of this paper, we make assumptions as follows:

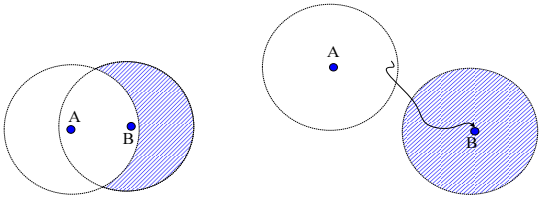
(1) *Agility*: Each network node has a reasonable likelihood of moving around. Network nodes (e.g. pedestrians and vehicles) may move at various speeds and may pause for a while from time to time. Hence, *agility* assumption is naturally satisfied in mobile ad hoc networks.

(2) *Autonomy*: Each node has independent control over its movement and the route of a mobile node is determined by the node itself. Other nodes cannot interrupt the movement of the mobile node.

B. Advantages of Mobility-assisted Epidemic Routing

It is well known that the *flooding-based protocol* (without utilizing mobility) usually has poor coverage of message delivery in sparse networks. Even in a reasonably dense network, node mobility frequently causes uneven node distribution, and results in network partitions in a snapshot view of the network. On the other hand, mobility-assisted epidemic routing schemes allow *message carriers* to forward the message in parallel from different locations, and assimilate "multi-source broadcasting". Hence, mobility-assisted routing schemes can achieve large coverage of message delivery.

Another advantage of mobility-assisted schemes is that message transmissions may achieve larger coverage than those in static case (see Figure 1). Hence, transmissions in mobility-assisted epidemic routing protocols can be more efficient. Based on such observation, we design a mobility-assisted epidemic routing protocol, where a message carrier forwards the message periodically.



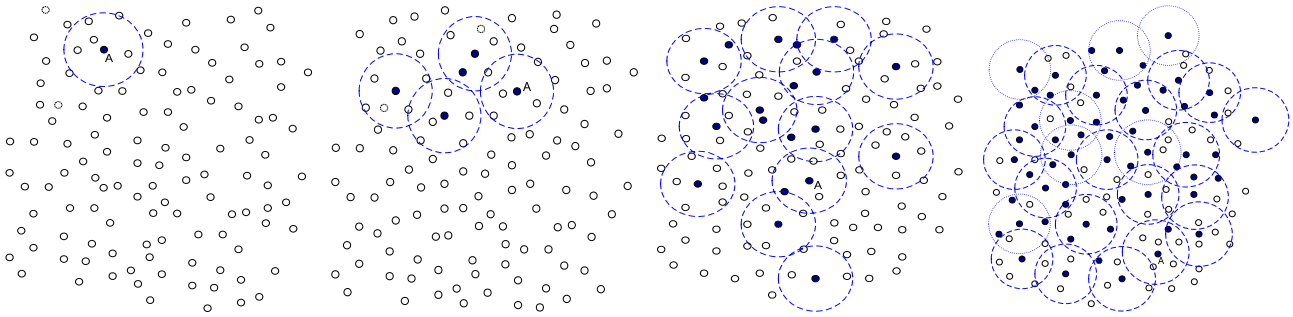
(a) Coverage of B's transmission without utilizing mobility

(b) Coverage of B's transmission in mobility-assisted protocols

Fig. 1. Mobility helps to disseminate message efficiently: When node A spreads a message, node B hears it. Without utilizing mobility in (a), B forwards the message to its neighbors upon reception of the message. The additional coverage of B's transmission is shown as the shaded area in (a). In (b), after B receives the message, it moves from the original location to a new location, and forwards the message in the new location, then the message coverage of B's transmission is the shaded area in (b).

C. Overview of Dandelion Protocol

Without considering *slander attacks*, upon receiving a message, a network node becomes a *mobile message carrier* and begins forwarding the message periodically. Each period



(a) At time $t = t_0$ (b) At time $t = t_0 + T$ (c) At time $t = t_0 + 2T$ (d) At time $t = t_0 + 3T$

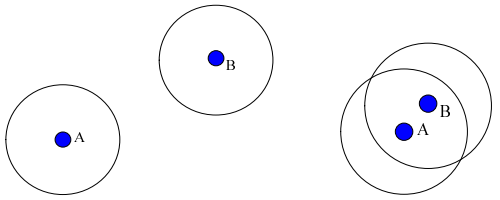
Fig. 2. Illustration of *mobility-assisted* epidemic routing

between retransmissions of a message has a duration T. Figure 2 demonstrates the *Dandelion* protocol. Initially, a node carries a message. At $t = t_0$, the node (dark node in Figure 2) broadcasts the message to other 4 nodes in its neighborhood as shown in Figure 2(a). Thus, t_0 marks the time when the message propagation begins. A node becomes a message carrier, if it has received at least one copy of the message. Immediately after t_0, 5 nodes have received the message, and become *mobile message carriers*. During time $[t_0, t_0+T)$, the 5 nodes (dark nodes) carrying the message move to new locations, and propagate the message in their respective neighborhoods in (Figure 2(b)). Note that one of the 5 nodes suppressed transmission in this period (we describe the suppress scheme in Section III-D). In such a way, more and more nodes receive the message and become *message carriers* (dark nodes) to further propagate the message (Figure 2(c)). With node mobility, a node encounters different neighboring nodes from time to time. In this way, *mobile message carriers* (the nodes which carry a message) bring the message to more and more nodes, and more and more nodes become *mobile message carriers* to accelerate the message propagation. Therefore, after a few steps (e.g. by time $t = t_0 + 3T$ in Figure 2(d)), almost all the network nodes have received the message.

D. Suppression of Transmissions

For the sake of efficiency in terms of message overhead, a node should suppress transmission if necessary. Assuming two nodes carry a message in Figure 3, if their transmission areas do not overlap (in Figure 3(a)), then they can retransmit their copies of the message without redundant coverage. However, if the two nodes are close enough to each other, the transmissions by these two nodes do not yield much additional coverage than a single transmission, as shown in Figure 3(b). In this case, one of the retransmissions could be suppressed to reduce message overhead. In our *Dandelion* protocol, if a node hears the transmission of a message within a short period from its neighborhood, the node suppresses the transmission of the message.

Figure 4 demonstrates message transmission schedule at each *mobile message carrier*. In order to design efficient suppression scheme, retransmissions of a message should occur within a small time slot Δ of each local time period. Assuming



(a) Carriers are far apart from each other (b) Carriers are close to each other

Fig. 3. If two message carriers are apart from each other in (a), then the covered area by two transmissions does not overlap, thus two transmissions yield the maximum coverage. If they are close to each other in (b), the covered area by two transmissions is reduced, hence one of the transmissions should be suppressed for efficiency purpose.

that a node schedules the transmission of a message at time t, if the node heard the transmission between $[t - \Delta - \tau_{max}, t)$, where τ_{max} is the upper bound of clock drift between two nodes, then it suppresses the transmission. Upon receiving a message initiated at t_0, each node retransmits the message at most once during local time $[t_0 + \delta T - \Delta, t_0 + \delta T]$, where $\delta = 1, 2, 3, \cdots$ and $\Delta \ll T$.

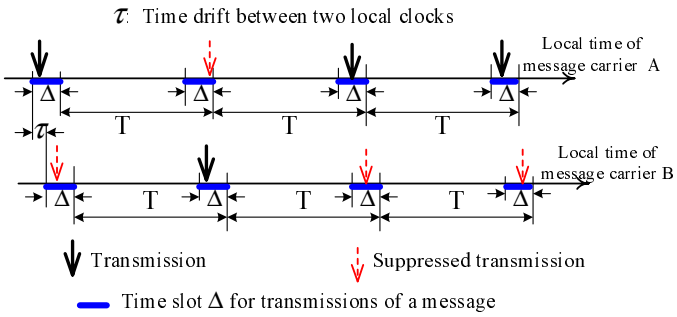


Fig. 4. Transmission schedule of two neighboring message carriers

Note, although we use the same T for all *message carriers* to retransmit, the nodes are not required to be synchronized. Actually, the intrinsic imperfect clock synchronization can help to avoid concurrent transmissions in a neighborhood, therefore reduce communication interference. Otherwise, randomness is utilized to achieve this goal.

E. Algorithm

A *times-to-send (TTS)* field is included in a message. When a source node initiates the message m, it attaches a positive *times-to-send (TTS)* value to the message m and serves as a message carrier of m at time t_0. With the message propagation procedure going on, more and more network nodes become *message carriers* which actively propagate message m to their neighbors. These mobile *message carriers* follow *Algorithm 1* for epidemic routing in each period. At time $t_0 + \delta T$ $(\delta = 1, 2, 3, \cdots)$, if *TTS* is larger than zero, the mobile *message carriers* reduce the *TTS* by one and retransmit message m to their neighbors unless a node hears someone else has transmitted the message around time $t_0 + \delta T$ in its neighborhood. *Algorithm 1* demonstrates the protocol for *message carriers* to propagate a message.

Algorithm 1:
Dandelion protocol to propagate a message m

While ($m.TTS > 0$) {
  Wait for period T.
  Set $m.TTS = m.TTS - 1$.
  If the node heard another copy of message m, denoted as m', from its neighbor in $[t_0+\delta T-\Delta, t_0+\delta T)$ time slot, where $\Delta \ll T$, then
  {
    Suppress the transmission;
    set $m.TTS = \min(m.TTS, m'.TTS)$*;
  }
  Else
    Transmit the message.
}

In *Algorithm 1*, (*) operation makes the algorithm robust to malicious nodes which use a large number to replace *TTS*, therefore cause unnecessary retransmissions. The original *TTS* value indicates the maximum number of transmissions that a mobile node broadcast a message. For example, the message carrier A in Figure 2 transmits its message 4 times, if originally we set *TTS=4*. It is an important design issue to determine *TTS* in a mobile ad hoc network so that almost all nodes can receive a message before *TTS* of the message reaches zero. We will discuss this issue in Section IV. When a node receives a new message m, it triggers *Algorithm 1* to further propagate the message.

IV. Parameter Selection

In previous sections, we introduced the *Dandelion* protocol. In this section, we discuss how to select parameters of the *Dandelion* protocol in order to optimize performance of the protocol.

A. Parameter T

In the *mobility-assisted* protocol, nodes carrying a message periodically broadcast the message to its neighbors. Intuitively, if the period T is large, then the delay of the message delivery is very large. However, if T is too small, the neighborhood of a node does not change too much from time $t_0 + (\delta - 1)T$ to $t_0 + \delta T$, hence the message transmitted at $t_0 + \delta T$ only reaches a small number of new neighbors which have not received message before. To maximize efficiency, two continuous transmissions by a mobile node should cover no common area (in Figure 5). Therefore, we have a lower bound of T as follows:

$$T > \frac{2r}{v_{avg}} \tag{1}$$

where r is the transmission range, and v_{avg} is the average velocity that a network node moves.

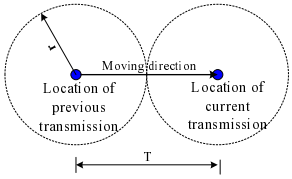


Fig. 5. Determination of T

B. Deduction of TTS

As alluded before, *TTS* controls the termination of message propagation. If the propagation procedure of a message is terminated early, then it is likely that only a portion of the network nodes receives the message. But if the propagation procedure is terminated late, a large amount of bandwidth can be wasted. Therefore, we aim to find the minimum *TTS* to ensure that almost all nodes can finally receive the message.

To deduce *TTS*, we assume there are $\mathcal{N}$ nodes in the network and all the nodes remain in the area $\mathcal{A}$ with size $|\mathcal{A}|$ during their mission period. A node may move arbitrarily within the area $\mathcal{A}$. Transmission range of a wireless node is denoted as r. In mobile ad hoc networks, node positions follow continuous processes in continuous time. However, the performance of the *Dandelion* protocol depends on the positions of the mobile nodes when they broadcast the message periodically. Hence, our concerns are the snapshot views of the network in different time slots with the interval (period) T.

We assume that mobile nodes including message carriers are evenly distributed in the incident area in any given time slot. This assumption is made for the convenience of the theoretical analysis, however, it is not the requirement of the *Dandelion* protocol. Let's consider an example in the circular incident area $\mathcal{A}$ with radius R (in Figure 6), the size of the incident area can be approximated as $|\mathcal{A}| = \pi R^2$. Also for theoretical analysis reasons, we divide the incident area $\mathcal{A}$ into hexagons instead of circles. If a node broadcasts a message, all the neighbors within its transmission range r can hear the message. We make an approximation that if a node within the hexagon transmits, all nodes in the hexagon can hear the transmission (Figure 6).

There are M hexagons covering the incident area $\mathcal{A}$. We estimate $M = \frac{|\mathcal{A}|}{6\times\frac{\sqrt{3}}{4}r^2} \approx \frac{|\mathcal{A}|}{2.6\times r^2}$. Let m be the message to

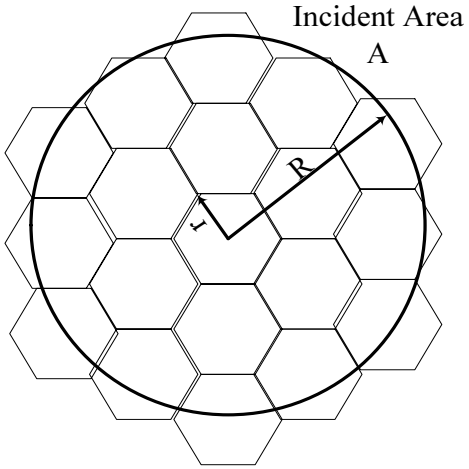


Fig. 6. Illustration of a circular incident area

be propagated in the network. The expected number of nodes which has received the message m at time t is denoted by $X(t)$, hence $X(t) \geq X(t')$ for $t > t'$ ($X(t)$ is non-decreasing). The *TTS* should be the minimum of δ to make $X(t_0 + \delta T) \approx \mathcal{N}$, where $\mathcal{N}$ is the total number of nodes in the network. Given $X(t_0 + \delta T)$, *Proposition 1* derives the average number of nodes which are aware of the message by the next period, $X(t_0 + (\delta + 1)T)$. Later on, we utilize *Proposition 1* to derive *TTS* (see *Algorithm 2*).

Proposition 1: The expected number of nodes which have received a given message at time $t_0 + (\delta + 1)T$ satisfies $X(t_0 + (\delta + 1)T) \geq \mathcal{N} \times (1 - e^{-X(t_0+\delta T)\frac{1}{M}})$.

Proof: For $t_0 + \delta T < t < t_0 + (\delta + 1)T$, there are $X(t_0 + \delta T)$ nodes carrying message m. Then the probability that a given hexagon with radius r does not contain any node which has received the message m is $(1 - \frac{1}{M})^{X(t_0+\delta T)}$. Hence, at time $t_0 + (\delta + 1)T$, the expected number of hexagons which contain at least one node carrying the message m is $M(1 - (1 - \frac{1}{M})^{X(t_0+\delta T)})$. Since each hexagon contains $\frac{\mathcal{N}}{M}$ nodes on average, we have:

$$\begin{aligned} X(t_0 + (\delta + 1)T) &= \frac{\mathcal{N}}{M}\{M[1 - (1 - \frac{1}{M})^{X(t_0+\delta T)}]\} \\ &= \mathcal{N} - \mathcal{N} \times (1 - \frac{1}{M})^{X(t_0+\delta T)} \quad (2) \\ &\geq \mathcal{N} \times (1 - e^{-X(t_0+\delta T)\frac{1}{M}}). \square \end{aligned}$$

Note that $X(t)$ is non-decreasing. Based on *Proposition 1*, we device an algorithm to determine *TTS* of the *mobility-assisted* epidemic routing protocol in *Algorithm 2*. Figure 7 shows the theoretical result of *TTS* depending on density of the network and according to *Algorithm 2*. Our observation is that the proposed *mobility-assisted* epidemic routing protocol is more efficient in a denser network. In Figure 7, *average degree* of a network is defined as the average number of neighbors within the transmission range of a node, so *average degree* indicates node density in a network.

Algorithm 2: Estimation of *TTS*

(1) Given network size $\mathcal{N}$ and $M = \frac{|\mathcal{A}|}{2.6 \times r^2}$.

(2) Initialize $\delta = 0$, $X(t_0) = 1$.

(3) While ($X(t_0 + \delta T) < \mathcal{N}$
do {
$X(t_0 + (\delta + 1)T) = \mathcal{N} - \mathcal{N}(1 - \frac{1}{M})^{X(t_0+\delta T)}$;
$\delta++$;
}

(4) Return *TTS* = δ.

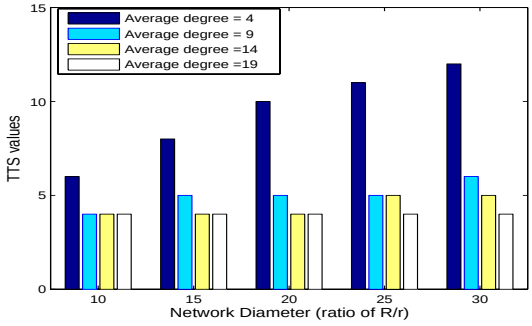


Fig. 7. Analytical results on TTS value

Next, we show via simulations that the message propagation demonstrates threshold property that a certain *TTS* ensures almost all network nodes receive the message, but any smaller *TTS* can barely achieve it. Let us consider *Random Waypoint* mobility model, and assume square incident area with size $|A| = 500 \times 500m^2$, transmission range $r = 30$ and network size $\mathcal{N} = 600$. According to the analytical result by *Algorithm 2*, we can deduce *TTS=6* for message distribution. In our experiment, among 30 runs of simulations, 28 runs take 6 periods to deliver the message to at least 98% percent of the network nodes. Only 2 runs take 5 periods to deliver the message to 98% of all nodes. It shows threshold behavior in the message propagation, so that with $TTS = 6$, almost all nodes receive a copy of the message, but with $TTS < 6$, the chance that the message can be delivered to the whole network is very small. Such threshold behavior implies that an initiator of a message may attach a predetermined value of *TTS* to the message from the beginning of the message propagation procedure, so that the message can be propagated to the whole network within *TTS* periods. We can see that the simulation results and the theoretical analysis agree with each other.

V. Robustness of Dandelion Protocol

In a mobile ad hoc network, a concern is the existence of selfish and malicious nodes. These nodes do not comply with the designed protocol, and even worse they may alter

parameters of a protocol to make the protocol less efficient or consume more resources. We discuss robustness of the *Dandelion* protocol against selfish and malicious nodes in this section.

A. Against Selfish Nodes

Upon receiving a message, a *selfish node* does not forward it, regardless of the *TTS* value of the message. Besides non-cooperative nodes, nodes with failure or insufficient power belong to the category of *selfish nodes*. It is important that the *Dandelion* protocol is able to tolerate the inevitable existence of selfish nodes.

In the *Dandelion* protocol, all mobile nodes may serve as *message carriers*. If a portion of selfish nodes suppress the transmission of a message, the other message carriers actively forward the message until its *TTS* reaches zero. On the other hand, if a single message carrier forwards a message in a given neighborhood, all the nodes in the neighborhood can receive the message. Hence, the *Dandelion* protocol is able to tolerate non-trivial portion of selfish nodes. Our simulation study in Section VI verifies this property of the *Dandelion* protocol.

B. Against Malicious Nodes

Next, we consider attacks where malicious nodes attempt to modify *TTS* values during message propagation. We study robustness of the *Dandelion* protocol against such attacks.

1) Maliciously Increase TTS Value: A malicious node may replace *TTS* value of a certain message with a larger value in order to consume extra resources for unnecessary transmissions. In the *Dandelion* protocol (*Algorithm 1*), let us assume that a node carries a copy of a message to be propagated, say m. When the node hears a copy of the message from its neighbors, it updates its *m.TTS* to set $m.TTS = min(TTS, m.TTS)$. Such update invalidates the attempt to increase *TTS* value of a message as long as an appropriate portion of network nodes still comply with the protocol. This is because the malicious nodes cannot change the mobility pattern of legitimate nodes, and they will not cluster all around the source/legitimate message carriers. Hence correct TTS value is able to propagated among legitimate nodes. Simulation results in Section VI are provided to show the robustness of the *Dandelion* protocol.

2) Maliciously Decrease TTS Value: Let's consider another type of *TTS* manipulating attack, where attackers reduce *TTS* value. The goal of reducing *TTS* values is to affect the reliability of the message propagation so that the message is only delivered to a portion of the network nodes. A malicious node sets $m.TTS = 0$. Upon receiving a message with zero *TTS*, a node stops forwarding the message. However, in this case a malicious node only suppresses its neighbors' transmissions in a single hop, and at the same time propagates the message m to its neighbors. If a malicious node tries to propagate the message with zero *TTS* multiple times to suppress more transmissions, it actually helps to propagate the message. It is a dilemma for malicious nodes. In Section VI, we use simulations to show that such a malicious attack is confined when there exist a small portion (e.g. 10%) of attackers.

VI. Evaluation

We have presented the *Dandelion* protocol for message propagation in previous sections, where we derived relevant parameters through theoretical analysis. In this section, we evaluate the performance of the proposed *Dandelion* protocols in terms of the end-to-end delay, the coverage (or reliability) of message delivery, and the efficiency in terms of message overhead, and robustness to non-cooperative nodes through simulations. Each simulation result is based on average of 50 runs.

A. Simulation Setup

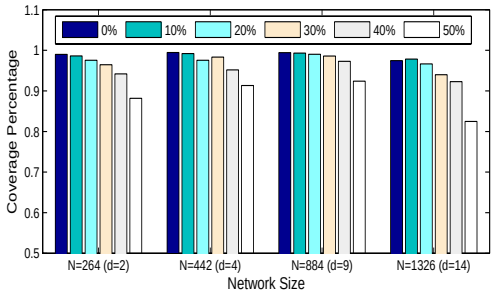
In this section, we investigate the *Dandelion* message propagation protocol under a commonly used mobility model, *Random Waypoint* mobility model, where a node pauses and moves. A node under *Random Waypoint* mobility model moves from its current position to a new randomly selected location (destination) at a random speed. The pause time of a node is also randomly chosen when it reaches the destination. After the pause time, the node chooses a new destination, speed, and pause time. This procedure is followed by every node until it reaches the end of simulation. We set the maximum pause time as 60 second and the range of moving speed is from $0.5mps$ (meter per second) to $5mps$ in the simulation.

We assume that the ad hoc network spans a square area with edge length $L = 500meter$ ($|\mathcal{A}| = L \times L$). We use variant transmission range r and the network size $\mathcal{N}$ to simulate different network diameters and different node densities in the network. We randomly select a node to initiate the message to be propagated. The node density of a network is represented by the average degree d. The period of message transmission is $T = 60\ second$.

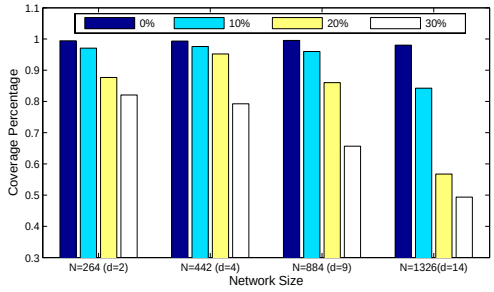
B. Coverage of Message Propagation

Coverage shows the reliability of the message propagation. It is measured by percentage of network nodes which has been received the message-to-be-propagated. Figure 8 illustrates the coverage of the *Dandelion* protocol. It tells us that even under the presence of selfish and malicious nodes, the coverage approximates 100%. We set TTS be 5, 6, 7, 8 respectively for different network sizes $N = 264,\ 442,\ 884,\ 1326$. Note that a network size determines the average degree d (node density) of the network, and $d = 2,\ 4,\ 9,\ 14$ for the given network sizes. Figure 8(a) illustrates the coverage under a certain portion of selfish nodes. Let's consider no-selfish-node case (0% of selfish nodes). In this case, when network is not dense enough, the coverage by the *flooding-based* protocol poor. However, in the *Dandelion* protocol, even if the network density is very low ($d = 2$), the coverage of message propagation approaches to 100%. We can conclude that the *Dandelion* protocol achieves much better message coverage in sparse networks. Figure 8(a) shows the coverage of the protocol under the presence of selfish nodes, which do not forward messages regardless of

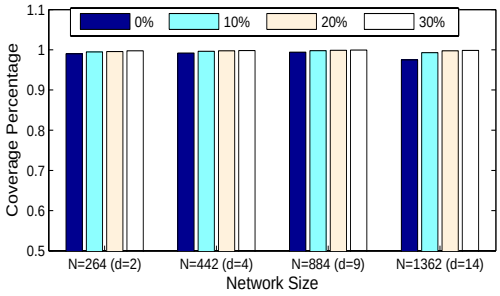
TTS value. When a portion of nodes are selfish, the network can be viewed as less dense network for unselfish nodes. Note that our theoretical analysis result in Figure 7 tells that in most of the cases *TTS* value is not very sensitive to node density. Therefore, with the *TTS* value designed for a given network, the coverage of the *Dandelion* protocol under less dense network is still satisfactory. This is verified by Figure 8(a), where we observe that the *Dandelion* protocol is able to tolerate the non-trivial portion of selfish nodes.



(a) Against selfish nodes



(b) Against malicious nodes which decrease the TTS



(c) Against malicious nodes which increase the TTS

Fig. 8. Robustness of the *Dandelion* protocol

Besides selfish nodes, we simulate the presence of malicious nodes, which may manipulate TTS values. In the simulation, both malicious nodes and legitimate nodes follow *Random WayPoint*. Figure 8(b) depicts the message coverage, when a certain percent of nodes maliciously replace TTS value with zero in order to prevent message propagation. For $d = 2, 4, 9$, if 10% of network nodes issue such an attack, the coverage still reaches 95%. For $d = 14$, the coverage is 85%, if 10% of nodes are malicious. When we consider the presence of more malicious nodes (e.g. 30% nodes are malicious), a message can be propagated to 50% and more network nodes. Thus, the *Dandelion* protocol is robust to such attack. Figure 8(c) shows the coverage, when a certain percent of nodes try maliciously to increase TTS value to consume extra resources for unnecessary transmissions. It is easy to understand that in this case coverage of the message propagation will not be degrade. Judging from Figure 8(b) and 8(c), we can conclude that the *Dandelion* protocol is robust to malicious attacks which try to manipulate TTS values. Actually, the *Dandelion* protocol is able to tolerate the existence of non-cooperative network nodes.

C. Message overhead

We have shown that the *Dandelion* message propagation protocol achieves high coverage. Next, we study the message overhead of the *Dandelion* protocol. To make a fair comparison of the *Dandelion* protocol and the *flooding-based* protocol, we define *normalized message overhead* as the average transmissions needed to cover each network node. The *normalized message overhead* is measured by total number of transmissions over the number of nodes which received the message, $i.e.\ Overhead_{msg} = \frac{\#\ of\ transmissions}{\#\ of\ nodes\ which\ received\ the\ message}$. $Overhead_{msg}$ can be interpreted as the average number of messages transmitted by each node which has received the message.

For the *flooding-based* protocol, we can easily conclude that the *normalized message overhead* tends to 1, since each node forwards the message once. *Normalized message overhead* of the *Dandelion* protocol is obtained through simulations as shown in Figure 9. If without malicious nodes (where 0% of nodes increase *TTS* value), we observe that in a sparse network (e.g. average degree of network nodes $d = 2$), *normalized message overhead* of the *Dandelion* protocol is slightly larger than 1. As a contrast, in sparse network the coverage of message propagation by the *Dandelion* protocol is much larger than that by the *flooding-based* protocol, because the reliability of message propagation under *flooding-based* protocols suffers from frequent network partitions. In reasonable dense networks ($d \geq 4$), the *Dandelion* protocol achieves much smaller *normalized message overhead* than that of the *flooding-based* scheme. This is because many nodes do not need to forward a message if they received the message with $TTS = 0$, or they hear the transmission from their neighbors. We can conclude that the *Dandelion* achieves good balance between reliability of message propagation and message overhead.

Figure 9 also demonstrates *normalized message overhead* under the TTS manipulating attack, where a portion of malicious nodes double TTS value to trigger extra unnecessary transmissions. From the simulation result, we conclude that the increment of message overhead under such malicious attack is moderate. This is because nodes update their *TTS* according to $m.TTS = \min(m.TTS, m'.TTS)$ every time when they receive a new copy of the message in *Algorithm 1*. Hence, the malicious effect is limited.

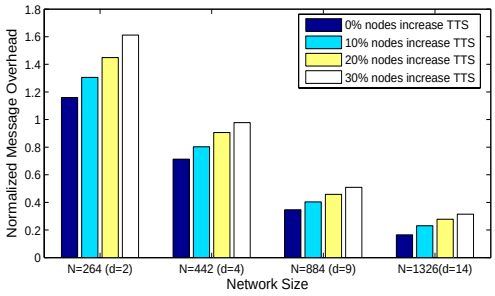


Fig. 9. Normalized message overhead under attacks where a portion of malicious nodes increase TTS values

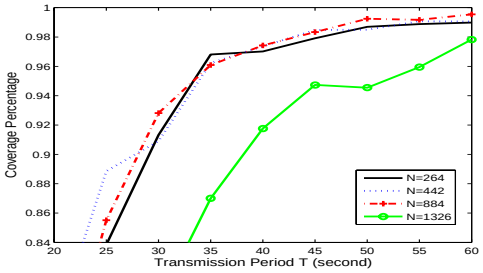
D. Influence by Parameters

As we mentioned before that there are two very important design parameters, which affect performance of the *Dandelion* protocol. In previous section, we explained how to determine these parameters by analysis. Now, we illustrate the influence of the parameters through simulations.

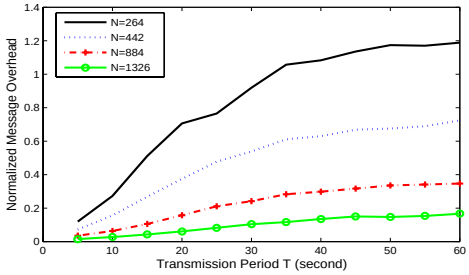
1) Parameter T: As we mentioned before that parameter T in the *Dandelion* protocol cannot be too small for efficiency. We suggested a lower bound of T by Eq. (1). Given that the speed of mobile nodes are randomly chosen from 0 to 5 *mps* and transmission range $r = 30\ m$, we can estimate $T > 24\ second$ accordingly. In this simulation, we set TTS equal to 5,6,7,8 respectively for different network density with $d = 2, 4, 9, 14$ (the corresponding network size is $N = 264, 442, 884, 1326$). We fix TTS value and investigate coverage and message overhead for different T. Figure 10(a) shows that the selection of parameter T affects performance of the *Dandelion* protocol. If T is too small the coverage of message propagation is poor. But as long as T is large enough (e.g. $T \geq 60\ s$), the coverage of message propagation is larger than 98%. Figure 10(a) implies that in a dense network (e.g $d = 14$), T should be large. Because in a dense network there are large number of transmissions in parallel in each time slot T, and a small original TTS is applied, (e.g. $TTS = 5$). To achieve better coverage, T should be larger to allow *message carriers* bring the message to more network nodes.

In figure 10(b), we observe that normalized message overhead is small when T is small in sparse networks. When T is small, the neighborhood of a message carrier does not change very much since its previous transmission. In the *Dandelion* protocol, if two mobile message carriers are within the transmission range of each other, one transmission will suppress the transmission of the other. Such suppression scheme causes small message overhead for a small T value.

2) Parameter TTS: To show the influence of TTS, let us examine the coverage of message delivery under the *Dandelion* protocol with different *TTS* values. The simulation result is shown in Figure 11. In the simulation, we take the transmission range $r = 30$. It indicates that 100% of network nodes can finally receive the message when *TTS* exceeds a certain value. Also, in a denser network, the coverage of message delivery converges to 1 faster. This is because in a denser network,



(a) Coverage of message propagation



(b) Message overhead

Fig. 10. We set $TTS = 5$ for $d = 2$, $TTS = 6$ for $d = 4$, $TTS = 7$ for $d = 9$ and $TTS = 8$ for $d = 14$ respectively, and investigate performance of the *Dandelion* protocol under different T values

one transmission of a message can reach more nodes, and more transmissions occur in parallel in each step. In Figure 11, we observe that the number of message carriers grows slowly at the beginning, but it proliferates when the number of message carriers exceeds a certain value. It tells us that if we take TTS values as 5, 6, 7, 8 respectively for different node density $d = 14, 9, 4, 2$, a given message can be successfully propagated to almost all network nodes. Such a result agrees with our theoretical estimation given by *Algorithm 2*.

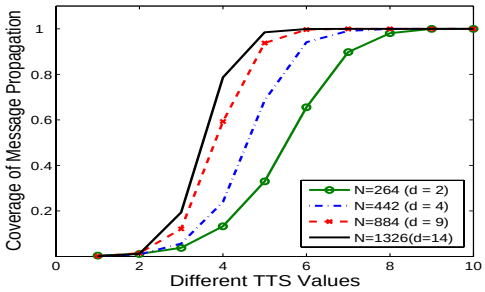


Fig. 11. Coverage of message propagation under different TTS value

E. End-to-end Delay of Message Propagation

In the simulation, the end-to-end delay of message propagation is the total time needed to propagate a message to the whole network since the message is generated. The node density of a network is represented by the average degree d. With L, r and network size $\mathcal{N}$, we can estimate $d = \mathcal{N}\frac{\pi r^2}{L^2} - 1$ for circular transmission areas. Figure 12 shows the delay of the message propagation. The result represents the average

of 10 simulation runs. We can observe that node density and transmission range affect the delay of the message propagation. *First*, we expect a less delay of message propagation in a denser network, because the denser a network is, the more nodes receive the message in each step. *Second*, with the same node density (average degree), a larger transmission range implies smaller L/r ratio (network diameter), which means that we can use smaller hops to cover the whole network. Hence, it takes a shorter time to propagate a message throughout the network with a larger transmission range r. With the delays shown in Figure 12, majority of network nodes are able to receive the given message in a short time, which is the number of periods times the retransmission interval T.

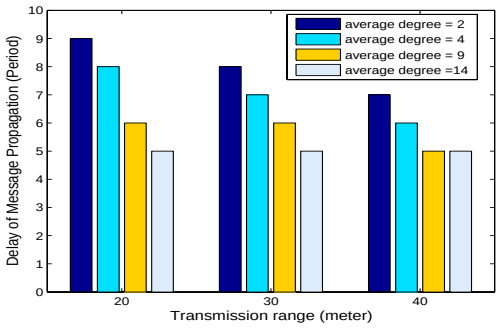


Fig. 12. Delay of message propagation

VII. Concluding Remarks and Future Work

The paper presented a novel message propagation protocol, the *Dandelion*, which relies on periodically retransmission of messages by mobile network nodes.

To minimize message overhead for message propagation, the transmissions of a message is limited by the parameter *TTS*. The *TTS* value demonstrates the threshold property that a certain *TTS* guarantees almost all network nodes receive the message, but any smaller *TTS* can barely achieve it. We deduce the value of *TTS* to achieve the threshold behavior by theoretical analysis, and verify it through simulation. Our simulation is based on *Random Waypoint* mobility model. The simulation results show that the *Dandelion* message propagation protocol achieves high reliability and robustness of message delivery with reasonable message overhead and acceptable delay. In the future, we will evaluate the *Dandelion* message propagation protocol with the real world mobility trace. To further improve performance of the *Dandelion* protocol, we will make every network node select its own T according to its velocity. We also consider to use hybrid (flooding and mobility-assisted) protocol for message propagation, so that in both static and mobile ad hoc networks, the message propagation protocol achieves optimum in terms of coverage, delay and message overhead. We will also evaluate our work under variety of mobility patterns.

VIII. Acknowledgement

The research in this paper is supported by Motorola grant 1-557641-239016-191100.

References

[1] S.-Y. Ni, Y.-C. Tseng, Y.-S. Chen, and J.-P. Sheu, "The Broadcast Storm Problem in A Mobile Ad Hoc Network," in *the Fifth Annual ACM/IEEE International Conference on Mobile Computing and Networking*, August 1999, pp. 152–162.

[2] B. Williams and T. Camp, "Comparison of broadcasting techniques for mobile ad hoc networks," in *Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing (MobiHoc)*, 2002, pp. 194–205.

[3] Y.-C. Tseng, S.-Y. Ni, and E.-Y. Shih, "Adaptive Approaches to Relieving Broadcast Storms in a Wireless Multihop Mobile Ad Hoc Network," *IEEE Transactions on Computers*, no. 5, pp. 545–557, May 2003.

[4] S. Pleisch, M. Balakrishnan, K. Birman, and R. van Renesse, "MISTRAL: efficient flooding in mobile ad-hoc networks." in *Proceedings of the 7th ACM Interational Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, May 2006.

[5] G. Karlsson, V. Lenders, and M. May, "Delay-tolerant Broadcasting," in *SIGCOMM'06 Workshops*, September 2006.

[6] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and Focus: Efficient Mobility-Assisted Routing for Heterogeneous and Correlated Mobility," in *Fifth Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops'07)*, March 2007.

[7] U. Lee, E. Magistretti, B. Zhou, M. Gerla, P. Bellavista, and A. Corradi, "Efficient Data Harvesting in Mobile Sensor Platforms," in *the 2nd IEEE International Workshop on Sensor Networks and Systems for Pervasive Computing (PerSeNS'06)*, March 2006.

[8] M. Grossglauser and D. N. C. Tse, "Mobility increases the capacity of ad-hoc wireless networks," in *Proceedings of the Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, April 2001, pp. 1360–1369.

[9] W. Zhao, M. Ammar, and E. Zegura, "A Message Ferrying Approach for Data Delivery in Sparse Mobile Ad Hoc Networks," in *the 5th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, May 2004.

[10] K. Harras, K. Almeroth, and E. Belding-Royer, "Delay Tolerant Mobile Networks (DTMNs): Controlled Flooding Schemes in Sparse Mobile Networks," in *IFIP Netwoking*, May 2005.

[11] N. Sarafijanovic-Djukic, M. Piorkowski, and M. Grossglauser, "Island Hopping: Efficient Mobility-Assisted Forwarding in Partitioned Networks," in *the 3rd Annual IEEE Communications Society on Sensor and Ad Hoc Communications and Networks, 2006. (SECON'06)*, September 2006.

[12] R. de Moraes, H. Sadjadpour, and J. GarciaLuna, "Throughput-delay analysis of mobile adhoc networks with a multi-copy relaying strategy," in *Proceedings of IEEE SECON*, October 2004.

[13] A. E. Gamal, J. Mammen, B. Prabhakar, and D. Shah, "Throughput-delay trade-off in wireless networks," in *Proceedings of the Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, March 2004.

[14] G. Sharma, R. R. Mazumdar, and N. B. Shroff, "Delay and capacity trade-offs in mobile ad hoc networks: A global perspective," in *Proceedings of the Twenty-fifth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, April 2006.

[15] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Performance Analysis of Mobility-assisted Routing," in *Proceedings of the 7th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, May 2006.

[16] Z. Zhang, "Routing in Intermittedtly Connected Mobile Ad Hoc Networks And Delay Tolerant Netowrks: Overview And Challenges," *IEEE Communications Surveys & Tutorials*, vol. 8, no. 1, 1st Quarter 2006.

[17] A. Vahdat and D. Becker, "Epidemic routing for partially connected ad hoc networks," *Technical Report CS-200006, Department of Computer Science, Duke University*, April 2000.

[18] D. E. Cooper, P. Ezhilchelvan, and I. Mitrani, "A Family of Encounter-Based Broadcast Protocols for Mobile Ad-hoc Networks," in *the 1st International Workshop of the EURO-NGI Network of Excellence*, June 2004.

[19] D. E. Cooper, P. Ezhilchelvan, I. Mitrani, and E. Vollset, "Optimization of Encounter Gossip Propagation in Mobile Ad-hoc Networks," in *the 13th IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems*, September 2005.