PANDA: A Novel Mechanism for Flooding Based Route Discovery in Ad Hoc Networks

Jian Li and Prasant Mohapatra
Department of Computer Science
University of California, Davis
Davis, CA 95616, USA.
Emails: {lijian, prasant}@cs.ucdavis.edu.

Abstract

Flooding technique is often used for route discovery in on-demand routing protocols in mobile ad hoc networks (MANETs) such as Dynamic Source Routing (DSR) and Ad hoc On-demand Distance Vector (AODV) routing. In this paper we present a Positional Attribute based Next-hop Determination Approach (PANDA) to improve the performance of flooding-based route discovery in MANETs using positional attributes of the nodes. These attributes may be geographical, power-aware, or based on any other quality of service (QoS) measure. We identify the "next-hop racing" phenomena due to the random rebroadcast delay (RRD) approach during the route discovery process in DSR and AODV, and show how the PANDA approach can resolve this problem. We assume that each node knows its positional attributes, and an intermediate node can learn the positional attributes of its previous-hop node via the received route-request message. Based on the attributes such as the relative distance, estimated link lifetime, transmission power consumption, residual battery capacity, an intermediate node will identify itself as good or bad candidate for the next-hop node and use different rebroadcast delay accordingly. By allowing good candidates to always go first, our approach will lead to the discovery of better end-to-end routes in terms of the desired quality of service metrics. Through simulations we evaluate the performance of PANDA using path optimality, end-to-end delay, delivery ratio, transmission power consumption, and network lifetime. Simulation results show that PANDA can: (a) improve path optimality, and end-to-end delay, (b) help find data paths with only 15%~40% energy consumption compared to the RRD approach at a moderate cost of increased routing messages, (c) balance individual node's battery power utilization and hence prolong the entire network's lifetime.

1 Introduction

A mobile ad hoc network (MANET) consists of a set of wireless devices that are capable of moving around freely and cooperate in relaying packets on behalf of one another. It does not require any fixed infrastructure or centralized administration. Instead, it is completely self-organizing and self-healing. MANETs have many potential applications in a variety of fields, like military tactical communication, disaster rescue and recovery, and collaborative group meetings.

MANETs have gained more and more attention from researchers in recent years. Many routing protocols have been proposed for use in MANETs [1]. Most of these proposals can be classified into two main categories: proactive protocols (e.g., DSDV [2]) and reactive (or on-demand) protocols (e.g., TORA [3], DSR [4] and AODV [5]). In general, proactive protocols rely on periodic exchange of routing information and each node maintains knowledge of the entire network topology, while reactive protocols depend on a query-based approach where a mobile node performs route discovery and route maintenance only when needed. Some of the on-demand protocols, like DSR and AODV, use flooding based query-reply mechanisms to search for a new route. LAR [7] is an improvement to DSR, which attempts to utilize geographic location information to restrict the flooding region. Some position-based (in contrast to topology-based) routing protocols, like GPSR [15] and GRA [17], attempt to utilize geographic information even more aggressively, in which packet routing is done on the basis of a greedy forwarding approach. In this paper, we restrict our discussion to on-demand protocols with route discovery based on flooding techniques. In particular, we propose to utilize various positional attributes such as geographical location, velocity, transmission power consumption, residual battery life to improve the performance of flooding-based route discovery in MANETs.

Flooding based route discovery works as follows. When a node S has some data to send to node D but has no existing route to the destination, it will initiate a route discovery process by broadcasting a route-request packet. An intermediate node I, upon receiving the route-request packet for the first time, will rebroadcast the route-request again if it does not know a route to the destination node D. Finally, when the route-request packet reaches a node (which may be the destination node D itself) that has a route to node D, a route-reply packet is sent back to the sender node S. To prevent broadcast storm due to synchronization, it was proposed in [16] that a random delay can be introduced before rebroadcasting a message and responding to a broadcast message. In particular, the delay time is uniformly distributed between 0 and 10 milliseconds. We argue that although this Random Rebroadcast Delay (RRD) approach is adequate for solving the problem of broadcast storm, it is not the most suitable one in term of searching for a better route to the destination. A better route may be based on metrics like shortest hops, delays, transmission power, and battery lifetime. In this paper we first consider the smallest number of hops for finding a feasible route. We will also discuss energy conserving route discovery using flooding technique, which is highly desirable in wireless sensor networks where sensor nodes have limited battery source.

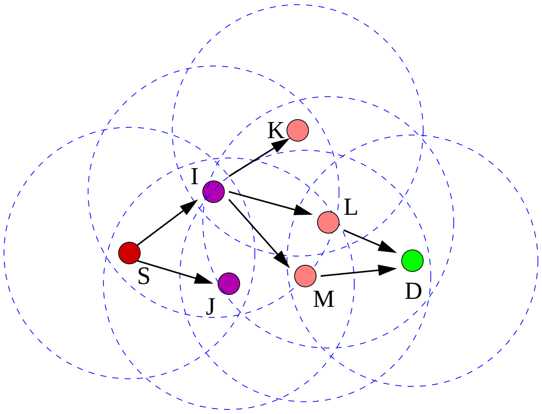


Figure 1. "Next-hop racing": A scenario using uniformly distributed rebroadcast delay in the flooding based route discovery process

Let us consider a scenario shown in Figure 1. Two intermediate nodes, I and J, receive a route-request packet from node S almost at the same time. Assume that node I is moving much faster than node J such that node I will move out of node S's range sooner than node J does. So we can say node J is a better candidate as the next hop in term of link lifetime. Since the rebroadcast delay is uniformly distributed between 0 and 10 milliseconds, it is possible that node I will rebroadcast the route-request message earlier than node J. In order to reduce routing overhead, each node will only rebroadcast a route-request packet for the same source-destination pair once within a certain period. Thus nodes K, L and M will relay the packet sent from node I and ignore the one sent from node J. In other words, node I, which is a worse next-hop candidate in term of link lifetime, "wins" over node J which is instead a better choice. We term this behavior as "next-hop racing." Another scenario can be recreated using the same example. Assume that the nodes of the network shown in Figure 1 do not have rechargeable batteries (e.g. sensor networks). If node I has very low residual battery power than node J, and node I "wins" the next-hop race then its residual battery power will get drained out soon. This situation will not only lead to a disconnection but may also reduce the useful lifetime of the entire network.

Our motivation for this paper is based on this observation. We propose to use positional attributes such as location and velocity information in determining the rebroadcast delay, while aiming at finding a longer-lived route with a smaller number of hops to the destination. We term our approach as Positional Attribute based Next-hop Determination Approach (PANDA). We will show that the PANDA approach can also be applied to discover routes based on other constraints like minimal transmission power consumption or residual battery life. In some cases like sensor networks, end-to-end delay may not be as important as energy conservation consideration. In these networks, wireless nodes are basically stationary (or have low mobility) and have limited battery source. It is desirable to discover routes that incur less power consumption when transferring data from source to destination. Our proposed approach is shown to perform very well in these scenarios, saving transmission power up to 60%~85% compared to the RRD approach. In other cases, it is desirable to balance the traffic among different nodes such that the battery drain rates are similar and different nodes have balanced lifetime.

The novelty of PANDA approach is in its simplicity of the algorithm itself, and its versatile capability of accommodating different purposes in route discovery process. By choosing the rebroadcast delays smartly according to different properties, PANDA algorithms can be used to achieve a variety of routing goals such as shortest hop routing, power conservation routing, and load balancing routing. The design goal of PANDA algorithms is to utilize location information in searching for better end-to-end route in term of desired QoS metrics. This is different from many existing broadcast techniques which utilize location information to suppress unnecessary rebroadcast (e.g., [7, 8, 20, 21, 35]). PANDA algorithms are fully distributed because they only rely on local information. Neighboring nodes cooperate in searching for better end-to-end route while competing to be chosen as next hop nodes.

The rest of this paper is organized as follows. In Section 2 we discuss different possible approaches to location aided route discovery in MANETs, and the proposal outline of PANDA. In Section 3, we discuss the basics of PANDA approach. The detailed designs of PANDA algorithms are presented in Section 4. In Section 5 we discuss the simulation of PANDA, and show its performance improvement by comparing the results of PANDA and the RRD approach. Related works in flooding based routing, power conserving routing, and location aided rebroadcast delay are discussed in Section 6. The paper is concluded in Section 7.

2 Different Approaches for Route Discovery

Much work has been done in the field of utilizing geographical location information in ad hoc routing protocols. Some flooding-based protocols, such as LAR [7] and DREAM [8], attempt to use location information to restrict the flooding region and thus reduce the flooding overhead. Some position-based protocols, such as GPSR [15] and GRA [17], attempt to use greedy geodesic forwarding schemes in routing packets. A survey on position-based routing protocols in MANETs can be found in [24]. We will further discuss these related works in the context of flooding based routing protocols later in Section 6. In this section we will address the topic of location aided route discovery from a different perspective. In particular, we identify three general approaches in existing literature to utilize geographical location information for on-demand route discovery. We will also state our proposal outline in this context: a Positional Attribute based Next-hop Determination Approach (PANDA) to improve the performance of flooding-based location-aided route discovery in ad hoc networks.

Approach 1: Periodic Beaconing + Sender-based Next-hop Selection

In this approach, each node periodically broadcasts a HELLO message to its neighbors, so that each node has the local knowledge (such as location and velocity) of all its neighbors. In route discovery phase, upon receiving a route-request packet, an intermediate node will choose the best neighboring node as the next hop (in terms of some metrics like farthest hop distance towards the destination location), and forward (unicast) the route-request packet to it. This procedure is iteratively performed, hop by hop, until a route to the destination is found. Note that in this approach, it is the sender (or upstream intermediate node) that decides which neighboring node has to be chosen as the next hop. Hence we term this approach "sender-based next-hop selection." A similar approach was proposed in ABR [18], which attempts to choose the best neighbor as the next-hop node in terms of *associativity* and other metrics. Several position-based routing protocols, like GPSR [15] and GRA [17], also use this approach in deciding the next-hop node when forwarding packets.

Compared to proactive protocols, like DSDV [2], this approach reduces the advertising overhead in the sense that it restricts the periodic beaconing messages within one hop range. It is also more feasible to keep track of the status of direct neighbors than to maintain knowledge of the entire network topology. The periodic beaconing, however, is still too expensive for mobile wireless nodes with limited bandwidth and battery power.

Approach 2: Flooding + End-to-End Path Evaluation

This approach is similar to DSR [4]. It does not require periodic HELLO messages, which dramatically reduces routing overhead in terms of number of routing messages. Without any global or local knowledge (except existing routes in the local cache), it uses flooding techniques to search for a new route when needed. Upon receiving a route-request packet, an intermediate node, without having an existing route to the destination in the local cache, will append its ID and other necessary information (e.g., location, velocity, battery capacity) to the route-request packet and rebroadcast it again. As mentioned earlier, a random rebroadcast delay is applied to prevent broadcast synchronization. Finally and hopefully, multiple copies of the route-request message along different routes will reach the destination node, that is, multiple routes may be found for a source and destination pair at the end of a route discovery process. Route evaluation is done at the end nodes (source or destination) based on some metrics. DSR (without using location information though) uses shortest hops as a selection metric at the sending side. A similar idea was proposed in [19], in which end-to-end routes are evaluated in term of nodes' behavior (malicious or beneficial).

Note that in this approach the sender (or upstream intermediate node) does not specify which neighboring node will be the next-hop node. A uniformly distributed rebroadcast delay is introduced mainly to address the problem of wireless broadcast storm. This uniform randomness also implies that there is no discrimination among the next-hop neighboring nodes. In other words, all neighboring nodes that receive the route-request message have an equal probability to become the actual next-hop node, which in turn implies that the set of routes resulting from a route discovery process may not include the better routes that exist in the topology. This case was demonstrated in Figure 1. Another drawback of this approach is that it has to carry a set of parameters (e.g., location, velocity, battery capacity, etc.) of all the intermediate nodes along the path in the route-request packet, which may make the route-request packet way too large as the network diameter increases.

Approach 3: Flooding + Sender-based Next-hop Selection

Similar to the previous approach, this approach does not require periodic HELLO messages. It also depends on flooding techniques to find new routes. What makes it different from Approach 2 is that the sender (or a upstream node) will specify, explicitly or implicitly, the qualifications of next-hop candidates. Upon receiving the broadcast route-request packet, only those neighboring nodes that satisfy the specified requirements will rebroadcast it again. Other neighboring nodes just drop the route-request packet silently. Since not all nodes in the network participate in rebroadcasting, route-request flooding overhead is reduced.

This approach is used in LAR [7], in which a *request zone* is specified in the route-request packet and only those nodes located inside the *request zone* will rebroadcast the route-request packet. DREAM [8] also uses this approach, which instead specifies a directional angle as the flooding zone. Note that in this approach, it is the sender (or an upstream node) that specifies a restricted area as the flooding candidates; hence the name "sender-based next-hop selection."

Approach 4: Flooding + Receiver-based Next-hop Determination

We propose a new approach for flooding based route discovery which relies on the characteristics of the receiver, and thus it can be categorized as "receiver-based next-hop determination" approach. The primary motivation of this approach is to address the problem of "next-hop racing" due to uniformly distributed rebroadcast delay. Our approach will give preference to good next-hop candidates (further discussed in Section 4) and hence will lead to the discovery of better end-to-end routes in terms of the desired QoS metrics.

Similar to Approaches 2 and 3, our approach does not require periodic beaconing messages. When a new route is needed, a route-request packet will be broadcasted by the sender. The sender (or upstream node) does not specify any requirement for next-hop candidates. Instead, the receiver (or downstream node) will identify itself as a good or bad candidate, and apply different rebroadcast delay accordingly. In particular, upon receiving a route-request packet, a neighboring node that identifies itself as a good candidate for next-hop will wait for a shorter time before rebroadcasting it, while a bad candidate will collaboratively defer its rebroadcast until the good candidates, if any, are done. In this way, good candidates will always go first and thus "win" in the route discovery process. Hence the "next-hop racing" phenomena (shown in Figure 1) is suppressed, and better end-to-end routes are discovered.

3 PANDA Basics

The basic idea of PANDA is to discriminate neighboring nodes as good or bad candidates for the next hop on the basis of positional or energy attributes that are of interest. These attributes can be relative distance and link lifetime estimation, transmission power consumption, and residual battery life. Note that this decision is made with local knowledge, i.e., information of current intermediate node and previous hop node. Good candidates will use shorter rebroadcast delay, while bad candidates use longer delay such that the good candidates always go first. As mentioned earlier, discrimination is done at the downstream node side. Since good candidates always go before bad ones, a better route in terms of metrics such as hop count, delay, power consumption, or residual battery, can be found.

We assume that each mobile node in a MANET is aware of its geographical location and velocity information. Similarly, in sensor networks, we assume a sensor node is aware of its location information. Later in Section 4, we will give more discussion on how a node can get its location and/or velocity information, with or without the Global Positioning System (GPS). To let the downstream nodes learn the previous-hop node's location (and velocity) information, these information is carried with the route-request message in each hop. Upon receipt of a route-request packet, an intermediate node can compare its own location and velocity with that of the previous-hop node and then determine the rebroadcast delay according to the algorithm it uses, namely, PANDA-LO (Location Only), PANDA-LV (Location & Velocity), or PANDA-TP (Transmission Power). Note that this decision is made at the downstream node side. Then the intermediate node replaces the related fields in the route-request packet with its own location and velocity information and rebroadcasts it after the locally-determined delay.

The addition of the location (and velocity) information in the routing packet naturally increases per packet overhead of routing messages. This extra overhead is, however, amortized since the discovered route will be used to transfer a larger number of data packets. Moreover, as we will see in the simulation results in Section 5, PANDA can reduce the number of routing messages, which also cancels out the effect of this increased overhead in individual routing messages.

Although PANDA approach is developed with the motivation as a location aided routing method, it can also be applied to improve routing performance without using location information. Along this line, we develop PANDA-RB (Residual Battery) algorithm. In PANDA-RB, actually no location information is utilized to determined the rebroadcast delay. Instead, each intermediate node considers its own residual battery capacity in determining the rebroadcast delay. The motivation behind taking the residual battery life into account is to balance lifetime of different nodes in the network system, hence prolong the lifetime of the whole system. This feature is very useful for sensor networks where recharging the battery is not an option.

PANDA algorithms can also be integrated with other optimization techniques, such as LAR [7] and Self Pruning [35]. LAR can reduce routing overhead since it only searches new routes in a limited sub-region. Using the idea of Self Pruning, in PANDA algorithms, a node in a "bad" class will give up relaying the route-request message if multiple duplicates are heard from other nodes in a "good" class, because it knows its message will be discarded by further neighbors anyway. This technique can reduce routing overhead since it suppresses unnecessary rebroadcast. To evaluate the performance gain in term of searching for better end-to-end routes by solely PANDA approach, we choose not to integrate other optimization techniques in the PANDA algorithms.

The PANDA algorithms are fully distributed in the sense that there is no intercommunications among the neighboring nodes except that they get the location and velocity information from the previous-hop node. Upon receiving a route-request message from the same previous-hop node, all the neighboring nodes run the same algorithm locally and independently. PANDA algorithms are designed in such a manner that, while competing for being chosen as the next-hop node, neighboring nodes cooperate in a way such that good candidates always go earlier than bad ones. Compared to the RRD approach, this feature naturally leads to the discovery of better end-to-end routes in terms of the desired performance and QoS metrics. By "better", we mean that the discovered route has a smaller hop count and end-to-end delay (as in PANDA-LO and PANDA-LV), or a smaller transmission power consumption (as in PANDA-TP), or it can help prolong the system lifetime (as in PANDA-RB).

The novelty of PANDA approach is in its simplicity and versatile capabilities of accommodating different purposes in the route discovery process. By smartly calculating the rebroadcast delays for different neighboring nodes, PANDA algorithms can be used to achieve a variety of routing goals such as shortest hop routing, energy efficient routing, and load balancing routing.

4 PANDA Designs

In this section we will discuss the detailed designs of different PANDA algorithms. We first present PANDA-LO (Location Only) algorithm and PANDA-LV (Location & Velocity) algorithm, both of which are employed to find a route with the smallest number of hops and lowest end-to-end delay. To show PANDA's capability in searching routes based on other constraints, we will also discuss how to apply the PANDA approach in searching a power-conserving route in sensor networks. For this purpose, we develop PANDA-TP (Transmission Power) algorithm. We also discuss PANDA-RB (Residual Battery) algorithm, which is aimed to balance the battery utilization and prolong the network's lifetime.

As in PANDA-LO and PANDA-LV, each individual node in a MANET is required to know its geographical location and velocity information. This is easy to achieved if each node is equipped with a GPS device. The GPS receiver can provide current readings on its location and velocity. In cases where GPS devices are not applicable, the network can use localization techniques to allow individual nodes to obtain their location estimates. Based on two consecutive location estimates and the observation interval, each node can also derive its approximate moving speed and direction. As in PANDA-TP (for sensor networks), a sensor node is required to know its location. In most cases, it is not feasible to require each individual sensor node has GPS capability. Various localization techniques have been developed to obtain location information without GPS capability in individual nodes (for example, see [31, 32, 33, 34]).

In the following designs, each PANDA algorithm is targeted at optimizing one routing goal. In some applications, it would be desired to achieve optimization among two or more routing goals. In such cases, some form of combination of our PANDA algorithms can be used to achieve optimization among different metrics. The implementation of such combination is straight forward if the required goals are not conflicting with each other. In other cases where the required multiple goals are conflicting, for example, smaller hop count (as in PANDA-LO and PANDA-LV) and less transmission power (as in PANDA-TP), some compromising rules need to be used to make trade-off among these conflicting goals.

4.1 PANDA-LO

End-to-end delay is accumulated in the course of multihop transmissions in MANETs. It is essential to reduce the hop count of the discovered routes in order to reduce end-to-end delay and save network resources such as bandwidth and battery source. PANDA-LO is an approach designed to discover routes with small hop count. In this approach, when determining the re-broadcast delay, we only consider the distance between two nodes without estimating the link lifetime. The basic idea of PANDA-LO is that the farther away a neighboring node is from the upstream node, the shorter rebroadcast delay it will use. Thus, a route-request packet always attempts to make a big jump in each hop of rebroadcasting. Intuitively, a shorter path in term of hop count will be found from source to destination using this approach.

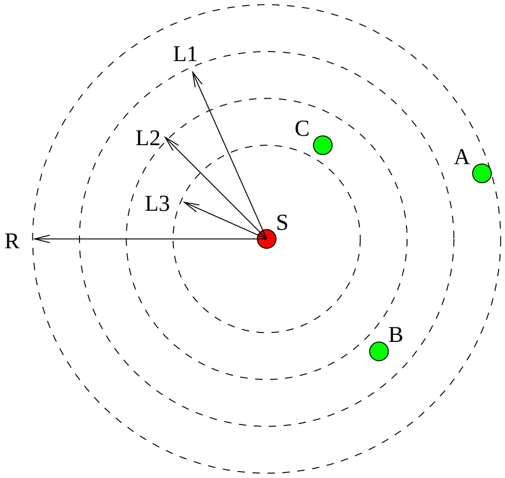


Figure 2. An example of “PANDA-LO” approach

Consider the example shown in Figure 2. When nodes A, B and C receive a route-request packet from node S, node A, which is farthest away from node S, will identify itself as the best next-hop candidate and use the shortest rebroadcast delay, while node C, which is closest to node S, will identify itself as a bad candidate and will wait until node A and B are done (without being aware of their existence though).

The calculation of link distance is based on the location of the current intermediate node and that of the previous-hop node. Take node A for example. When node A receives the route-request from node S, node A will calculate its distance to the upstream node S as follows:

$$\| SA \| = \sqrt{(X_s - X_a)^2 + (Y_s - Y_a)^2}, \tag{1}$$

where (X_s, Y_s) and (X_a, Y_a) are S and A’s locations, respectively. Having the distance information, node A can use PANDA-LO to determine its rebroadcast delay accordingly.

A possible implementation of “PANDA-LO” approach is shown in Algorithm 1. We choose appropriate threshold values for L_1, L_2, and L_3 such that $L_1 > L_2 > L_3$. From our experience in the simulations, the selection of values for these parameters are related to specific network scenarios such as transmission range, node density distribution and node moving speed. Considering that we assume a uniformly randomly distributed network, a natural way to choose these values is using evenly spaced steps. This algorithm classifies neighboring nodes into four classes which will determine the use of different rebroadcast delays. In Algorithm 1, t_1 is the base time of delay in milliseconds, and the function $uniform(0,\, t_1)$ will return a random value uniformly distributed between 0 and t_1. As our design goal, a node in a better class of next-hop candidates will use shorter rebroadcast delay. In particular, neighboring nodes in Class 1 differ a random time uniformly distributed in range of $(t_1,\, 2 * t_1)$, and neighboring nodes in Class 2 differ a random time uniformly distributed in range of$(2 * t_1,\, 3 * t_1)$, and so on. By choosing different values for t_1 we can vary the ranges of the delay times for different classes. Note that the delay times of different classes do not overlap each other, which is intended to guarantee that better candidates always go first. However, due to the randomness incurred by $uniform(0,\, t_1)$, candidates within each single class may go before each other randomly.

Algorithm 1 Determining Rebroadcast Delay in "PANDA-LO"

```
at node A
if $\| SA \| > L_1$
    delay = $t_1 + uniform(0,\, t_1)$        //this is Class 1
else if $\| SA \| > L_2$
    delay = $2 * t_1 + uniform(0,\, t_1)$ //this is Class 2
else if $\| SA \| > L_3$
    delay = $3 * t_1 + uniform(0,\, t_1)$ //this is Class 3
else
    delay = $4 * t_1 + uniform(0,\, t_1)$ //this is Class 4
```

Algorithm 1 is to sort out the neighboring nodes based on their hop distances, and assign different rebroadcast delays, which embody different routing preferences. Please note that, while it would select some specific direction(s) to go first, PANDA-LO does not stop other possible search directions. In other words, PANDA-LO keeps the advantage of flooding based route discovery while incorporating its new functionality in QoS-aware routing. This desirable feature is maintained in our other designs of PANDA algorithms.

We want to point out that it is possible that the furthest neighboring node could be out of the transmission range of the upstream node pretty soon if they are moving apart. So PANDA-LO may lead to fragile paths because it does not consider the link lifetime in the process of route discovery.

As discussed earlier, all neighboring nodes determine their rebroadcast delays independently without being aware of the existence (or nonexistence) of one another. Consider a rare case at a certain hop in which there are only a few neighboring nodes, all of which fall into the worst category - Class 4. These neighboring nodes will wait for a random delay time in the range of $(4*t_1,\ 5*t_1)$. Thus they waste some amount of time in waiting and hence incur unnecessary delay in the route discovery process. However, this route discovery delay only affects packets currently waiting in buffer for transmission, which accounts for a small portion of total traffic over a long period. Additionally, if the mobile nodes are uniformly distributed in a specific area, occurrence of such a case is rare.

4.2 PANDA-LV

Now let us discuss the PANDA-LV approach which uses both location and velocity information to determine the rebroadcast delay. By estimating the link lifetime and choosing neighboring nodes with stable links as the next hops in route discovery, we expect to find longer-lived as well as relatively shorter path from a source to a destination.

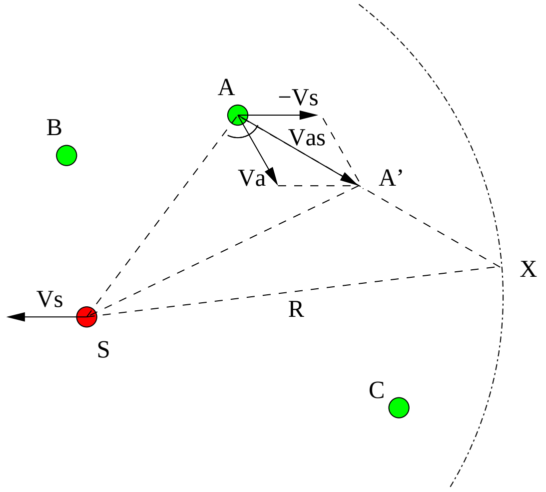


Figure 3. An example of “PANDA-LV” calculation

Consider the example shown in Figure 3. An upstream node S broadcasts (or rebroadcasts) a route-request packet, and its downstream nodes are A, B and C. How will the downstream node determine if it is a good candidate or not? Let's consider node A for example. First, node A will calculate its distance to the upstream node S by using equation (1). Node A will also estimate the lifetime of the link between nodes A and S based on the distance and relative velocity. Assume the wireless transmission range is R. In Figure 3, assume V_s and V_a are S and A's velocity respectively. Let $V_{a,s}$ be the relative velocity of node A to node S, θ be the angle $\angle SAA'$ (or $\angle SAX$) , and $\| AX \|$ be the distance that node A can move before it is out of the transmission range of node S(assuming S and A would not change their moving speeds and directions during this period). Based on the cosine theorem, the formulas we use to calculate the estimated link lifetime are as follows:

$$\| V_{a,s} \| = \| V_a - V_s \| = \| AA' \| \tag{2}$$

$$cos(\theta) = \frac{\| SA \|^2 + \| AA' \|^2 - \| SA' \|^2}{2 * \| SA \| * \| AA' \|} \tag{3}$$

$$\| AX \| = \sqrt{\| SX \|^2 - \| SA \|^2 + \| SA \|^2 * cos^2(\theta)} + \| SA \| * cos(\theta) \tag{4}$$

$$LIFETIME_{a,s} = \frac{\| AX \|}{\| V_{a,s} \|} \tag{5}$$

Note that $LIFETIME_{a,s}$ is the estimated lifetime of the link between node A and S. Intuitively, the longer the lifetime of each link along the path, the longer-lived the route is as a whole.

Having the distance and link lifetime information, node A can run Algorithm 2 to determine its qualification and set its rebroadcast delay accordingly. In Algorithm 2, L_1 and L_2 are two threshold values for distance, and T_1, T_2, and T_3 are threshold values for the estimated link lifetime. We choose appropriate values for these thresholds, which satisfy $L_1 > L_2$ and $T_1 > T_2 > T_3$, such that Class 1 is better than Class 2, which is in turn better than Class 3, and so on. Class 1 is farthest away($L_1 > L_2$) from the upstream node S and the link lifetime is the longest($T_1 > T_2 > T_3$), while Class 2 has the same distance as Class 1 but a shorter link lifetime, and Class 3 has even shorter(but still fairly good) distance and link lifetime. All other nodes falls into Class 4, which represents the worst candidates as the next-hop nodes. As in Algorithm 1, good candidates use shorter rebroadcast delay.

Since PANDA-LV attempts to consider both hop distance and link lifetime together, there is

Algorithm 2 Determining Rebroadcast Delay in "PANDA-LV"

```
at node A
if $\| SA \| > L_1$ && $LIFETIME_{a,s} > T_1$
    delay = $t_1 + uniform(0, t_1)$        //this is Class 1
else if $\| SA \| > L_1$ && $LIFETIME_{a,s} > T_2$
    delay = $2 * t_1 + uniform(0, t_1)$ //this is Class 2
else if $\| SA \| > L_2$ && $LIFETIME_{a,s} > T_3$
    delay = $3 * t_1 + uniform(0, t_1)$ //this is Class 3
else
    delay = $4 * t_1 + uniform(0, t_1)$ //this is Class 4
```

a need to choose a balanced combination of these two metrics.This implies a subtle situation in choosing appropriate values for L_1, L_2 and T_1, T_2, T_3. Additionally, these choices may be related to network density and spatial distribution, some form of tuning is necessary to obtain network specific threshold values.

We would like to point out that the classification demonstrated in both Algorithms 1 and 2 just embodies the idea of discriminating next-hop nodes as good or bad candidates. Neither the four classes are necessary, nor are they typical. We can certainly choose other threshold values and use finer or coarser granularity classes. The four-class differentiation shown in this paper is just an example and should be taken in that spirit.

4.3 PANDA-TP

In some cases such as wireless sensor networks, power conservation is more important than reduction of end-to-end delay. Wireless sensor nodes have limited battery source which cannot be replaced or recharged in most situations. Thus it is desirable to discover power conserving end-to-end routes such that the lifetime of the whole network can be prolonged as much as possible. These networks normally have high node density and very low mobility. To achieve the goal of power conservation, it would be desirable to break a big single hop into several small hops such that each small hop needs very small transmission power and the overall power consumption along the path is much smaller than a big single hop, as demonstrated in the following example.

Let us consider the example shown in Figure 4. Node S can send data to node D directly in one single hop of distance R, or in three small hops of distance R/3 via intermediate nodes A and

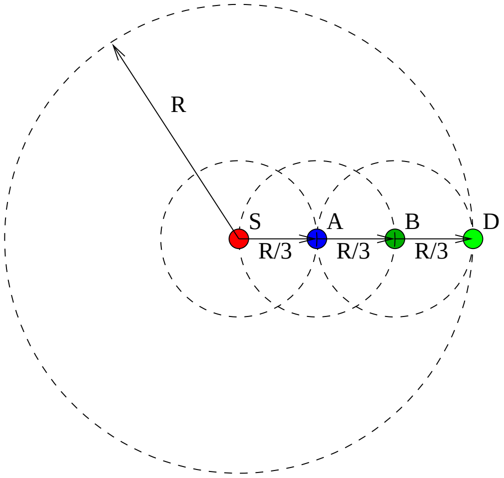


Figure 4. Transmission power: single hop v.s. multihop

B. We assume each node requires the same minimal receiving power P_{RXmin} for correct packet reception. We also assume the propagation loss L is a simple function of distance R as follows [9]:

$$L = c * R^{\alpha}, \tag{6}$$

where c and α are constants. So, the required transmission power for a single hop over distance R and that for a small hop over distance R/3 are, respectively:

$$P_{TX(S\,to\,D\,directly)} = P_{RXmin} * L = P_{RXmin} * c * R^{\alpha} \tag{7}$$

$$P_{TX(S\,to\,A)} = P_{TX(A\,to\,B)} = P_{TX(B\,to\,D)} = P_{RXmin} * c * (R/3)^{\alpha} \tag{8}$$

We obtain the total transmission power along the path and the ratio of power consumption as:

$$P_{TX(S\,to\,D\,via\,A\,and\,B)} = 3 * P_{RXmin} * c * (R/3)^{\alpha} \tag{9}$$

$$Ratio = \frac{P_{TX(S\,to\,D\,via\,A\,and\,B)}}{P_{TX(S\,to\,D\,directly)}} = \frac{3 * P_{RXmin} * c * (R/3)^{\alpha}}{P_{RXmin*c*R^{\alpha}}} = \frac{1}{3^{(\alpha-1)}} \tag{10}$$

Note that the propagation constant α is often assigned a value between 2 and 4 in practice, which makes the power consumption ratio small. For a given distance, as the number of hops increases, the power consumption ratio decreases. Thus, in the route discovery phase, it is desirable to choose close neighboring nodes as next-hop candidates. Using Figure 2, we can derive PANDA-TP algorithm, which is similar to PANDA-LO algorithm but works the other way around.

Algorithm 3 Determining Rebroadcast Delay in "PANDA-TP"

```
at node A
if $\| SA \| < L_3$
    delay = $t_1 + uniform(0, t_1)$        //this is Class 1
else if $\| SA \| < L_2$
    delay = $2 * t_1 + uniform(0, t_1)$ //this is Class 2
else if $\| SA \| < L_1$
    delay = $4 * t_1 + uniform(0, t_1)$ //this is Class 3
else
    delay = $6 * t_1 + uniform(0, t_1)$ //this is Class 4
```

PANDA-TP algorithm is shown in Algorithm 3. Referring to Figure 2, L_1, L_2 and L_3 are distance threshold values that satisfy the relation: $L_3 < L_2 < L_1$. Similar to PANDA-LO shown in Algorithm 1, t_1 is the base delay time and $uniform(0, t_1)$ will return a random value uniformly distributed between 0 and t_1. In this PANDA-TP algorithm, neighboring nodes are also classified into four classes. Unlike PANDA-LO where farther neighboring nodes use shorter delay, PANDA-TP allows closer neighboring nodes to go first. In particular, close neighboring nodes will be classified as good candidates and use short rebroadcast delay, while neighboring nodes far away will be classified as bad candidates and wait for longer delay. So in PANDA-TP scheme, each hop attempts to make a relatively small jump, and thus the total power consumption of the route is hopefully small.

What makes PANDA-TP different from the RRD approach is the way to deal with duplicate route-request messages and the way to determine rebroadcast delay. In the RRD approach, each intermediate node has a request-cache table and will ignore a replicated route-request packet if it has been heard recently. In PANDA-TP approach, the overall transmission power of the partial route that has been traversed so far is carried with the route-request packet. If the duplicate route-request packet is from a path with less power consumption, the intermediate node will still rebroadcast it again. As shown in Algorithm 3, the rebroadcast delay is determined on the basis of hop distance. By this way, PANDA-TP attempts to explore the network for paths with less power consumption at the cost of increased routing messages. In order to reduce the increase of routing overhead, PANDA-TP can choose to ignore duplicate messages with marginal effect (i.e., the reduction in power consumption is too small).

To make PANDA-TP work as expected and reduce the number of routing messages, it is especially important to choose appropriate delay values for different classes. Take Class 1 and Class 2 for example. When the source node broadcasts a route-request message, all nodes in the four classes hear this message and applies different delays before relaying it. Nodes in Class 1 will at most wait for $2 * t_1$ time in the worst case. To make sure that the relayed message (i.e., duplicate message with less transmission power consumption) via Class 1 will reach nodes in Class 2 before they relay the first copy (which has larger power consumption), nodes in Class 2 have to use a delay which is larger than $2 * t_1$. Similarly, duplicate messages via Class 1 need two small hops to reach Class 3, which incurs $4 * t_1$ delay in the worst case, so Class 3 has to wait for a time greater than $4 * t_1$.

4.4 PANDA-RB

PANDA-RB is a variant of PANDA approach that does not use location information. Instead, this method is concerned with the current intermediate node's residual battery capacity.

PANDA-RB attempts to balance the traffic among nodes in the network such that different nodes experience similar battery drain rates. Upon receiving a route-request message, an intermediate node first checks its residual battery capacity. An intermediate node with high battery capacity will consider itself as a good candidate and apply a short rebroadcast delay. Since at each hop PANDA-RB attempts to choose a node with high battery capacity as next hop, the end-to-end route discovered at the end of the routing process may consist of nodes with longer battery life. The algorithm of PANDA-RB is shown in Algorithm 4. Note that P_1, P_2, P_3 are percentage threshold values which satisfy $1.0 > P_1 > P_2 > P_3 > 0$.

In DSR, the destination node may receive and reply to multiple route-request initiated by the same source node. Since at each hop "bad" intermediate nodes have longer delay, we expect that

Algorithm 4 Determining Rebroadcast Delay in "PANDA-RB"

```
at node A
$RBP = \frac{\text{residual battery}}{\text{full charge battery}}$
if RBP > $P_1$
    delay = $t_1 + uniform(0, t_1)$        //this is Class 1
else if RBP > $P_2$
    delay = $2 * t_1 + uniform(0, t_1)$ //this is Class 2
else if RBP > $P_3$
    delay = $3 * t_1 + uniform(0, t_1)$ //this is Class 3
else
    delay = $4 * t_1 + uniform(0, t_1)$ //this is Class 4
```

the messages that experience one or more "bad" intermediate nodes should come later than those that come along a route with nodes having high residual battery power. To avoid sending back routes with low battery life, we can limit the number of replies the destination node will send within a certain amount of time. In particular, the destination node will only reply to the first few duplicate route-request messages.

Another technique to improve PANDA-RB could be "selective ignoring," which allows an intermediate node with some probability not to participate in a new route discovery process if its residual battery percentage (RBP) is below some critical threshold level.

5 Simulations and Results

In this section we evaluate the performance of PANDA approaches through simulations. We use the *ns-2* simulator [36] to simulate PANDA-LO and PANDA-LV algorithms.

Using a customized simulator, we evaluate the capability of PANDA-TP scheme in term of finding power conserving end-to-end routes in wireless sensor networks. We focus on the power consumption of the routes discovered by PANDA-TP and by the RRD approach.

We also use *ns-2* to evaluate the performance of PANDA-RB algorithm by comparing it with RRD approach. Simulation setup for PANDA-RB and the results are presented in Section 5.3.

5.1 PANDA-LO and PANDA-LV

The Monarch Group's mobility extension [37] to the *ns-2* simulator provides detailed implementation of IEEE 802.11 radio and MAC specifications. In order to compare the results of the PANDA approaches and the RRD approach, we utilize the codebase of DSR in the *ns-2* simulator and integrate PANDA-LO and PANDA-LV algorithms into DSR. Although our discussion and simulation of PANDA-LO and PANDA-LV is based on DSR, these PANDA algorithms are applicable to other flooding based routing protocols for MANETs, such as AODV. We have integrated PANDA into AODV and the simulation results are quite similar to that of DSR. To avoid repetition, we show the results based on DSR scheme only. In any case, the proposed approach is independent of the underlying routing algorithm.

We mainly consider the path optimality ratio and end-to-end delay, which are the major design goal of our routing protocol. Meanwhile, we also measure other metrics, such as the number of route errors, the number of routing messages and packet delivery ratio, to show that PANDA algorithms do not sacrifice other aspects when achieving its major goal. Instead, in most cases, PANDA can improve these metrics, more or less, compared to the RRD approach.

Here the path optimality ratio is defined as:

$$\text{path optimality ratio} = \frac{\text{hop count of the actual path}}{\text{hop count of the optimal path}}. \tag{11}$$

According to the above definition, we use the result of the centralized routing scheme as the benchmark for both PANDA and RRD. The best-case centralized routing would provide the optimal path with shortest hop count. Using the path optimality ratio, we are not only able to compare the performance between PANDA and RRD, but also get an idea of how good PANDA's performance is when compared to the centralized optimal scheme.

5.1.1 Simulation Setup

The simulation area is 1500×300 square meters. A node's speed is uniformly distributed in the range of (0, 20) meters per second, and its wireless transmission range is 250 meters. The nodes move according to the Random Waypoint model [16], and the communication pattern is peer-to-peer communications, as is provided by the Monarch Group's mobility extension [37] to *ns-2*. To analyze PANDA's performance under low and high node densities, we use 50 nodes and 100 nodes, respectively. In both cases, there are 30 constant bit rate (CBR) connections, each of which randomly starts during the first 180 seconds and has a packet rate of 4 packets per second. Each simulation runs for 500 seconds of simulation time. Initially, nodes are randomly distributed in the simulation area. After the simulation starts, each node will stay at its initial location for *pause-time* seconds, and then randomly chooses a destination location within the simulation area and starts moving towards the destination with a speed randomly chosen between (0, 20) meters per second. After it arrives at the destination, it stays there for *pause-time* seconds and then chooses a new destination and new speed and moves again. The parameter *pause-time* reflects the degree of mobility of a MANET. For different mobility, we use different *pause-times* of 0, 30, 60, 150, 300, and 500 seconds. When *pause-time* is 0 seconds, it means that all nodes are moving all the time and the MANET has a high degree of mobility. When *pause-time* is 500 seconds, it means that all nodes are stationary during the simulation period. For each *pause-time* (i.e., each point of the curves), we run multiple rounds of simulations using different moving patterns and then obtain the average values.

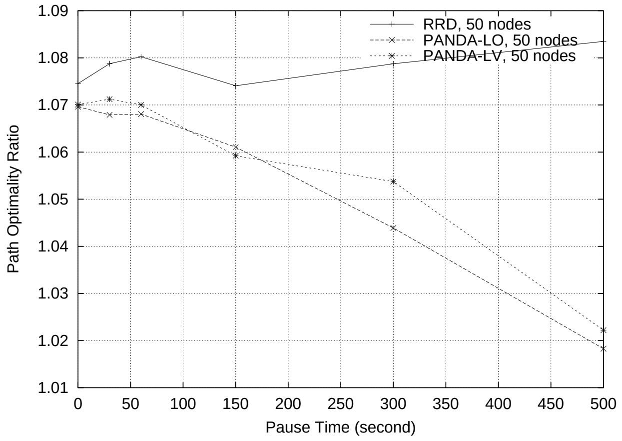
As discussed in earlier sections, appropriate choices of the threshold values in Algorithm 1 is needed to get good performance in specific network scenarios. As proposed in [16], RRD uses a random delay to overcome the broadcast storm problem. As stated by the authors [16], the best choice for this random delay is between [0, 10] ms. Similarly, the function $uniform(0, t_1)$ in PANDA algorithms is mainly used for this purpose. Since the neighboring nodes have been divided into four smaller classes, we set $t_1 = 5ms$ in all these simulations. In other words, in PANDA-LO and PANDA-LV, the random delay range for individual classes are [5, 10] ms, [10, 15] ms, [15, 20] ms, and [20, 25] ms, respectively. For the distance thresholds, we choose $L_1 = 200m$, $L_2 = 150m$, and $L_3 = 100m$. For PANDA-LV algorithm, we choose $T_1 = 80s$, $T_2 = 40s$, and $T_3 = 30s$.

5.1.2 Simulation Results

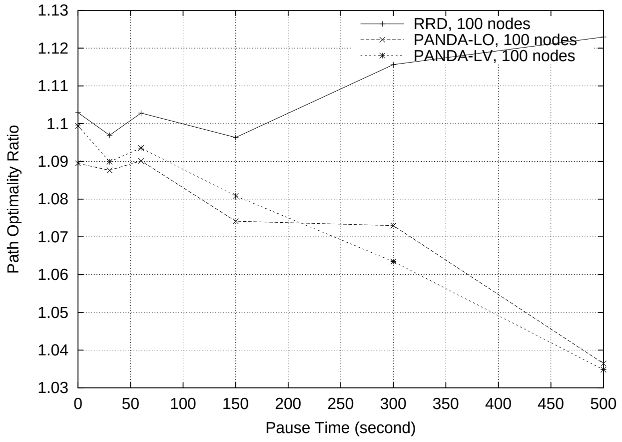
First, let us observe the path optimality ratio shown in Figure 5. Recall that the path optimality ratio is defined as the actual path length over the shortest path length. So the lower the ratio, the better is the path. In both 50 node and 100 node networks, both PANDA-LO and PANDA-LV achieve better path optimality than RRD. This is because PANDA algorithms attempt to make a longer jump in each hop of rebroadcast in the process of route discovery, which naturally leads to shorter end-to-end route in term of the number of hops. Since PANDA algorithms rely on local decisions without any global knowledge, it is a greedy heuristic in nature and cannot guarantee the discovery of the optimal path. This is reflected in the path optimality, which is always greater than 1.0. However, when the network becomes more stable (as the pausetime goes up to 500s in our simulations), the path optimality ratio of PANDA is very close to 1.0, which means that PANDA achieves very good end-to-end performance even though it is based on per hop decisions with local information only.

In Figure 5, we can also observe that the path optimality ratio for both PANDA-LO and PANDA-LV drops as the *pause-time* increases, while the path optimality of RRD increases a little bit instead. That is to say, the improvement in path optimality by PANDA becomes greater as the *pause-time* becomes larger. This phenomena can be explained in this way: as the *pause-time* increases, the degree of mobility decreases and the network topology is more stationary. So once a route is discovered between a pair of source and destination, the route will be used for quite a long time because no route breakage is likely to occur. Since RRD is likely to discover longer routes than PANDA, a more static network topology means that a larger number of packets will have to go through longer routes in RRD. This is the reason why PANDA will perform even better than RRD in a static MANET.

Before we move ahead to other simulation results, we would like to point out that, as can be observed in Figure 5 and next few figures, the behaviors of the curves between pausetime 0~60 seconds seems not so consistent with those from 60~500 seconds. We believe these dynamics are due to the mobility model we use in our simulations. More details about the instability of the Random Waypoint mobility model can be found in [25].



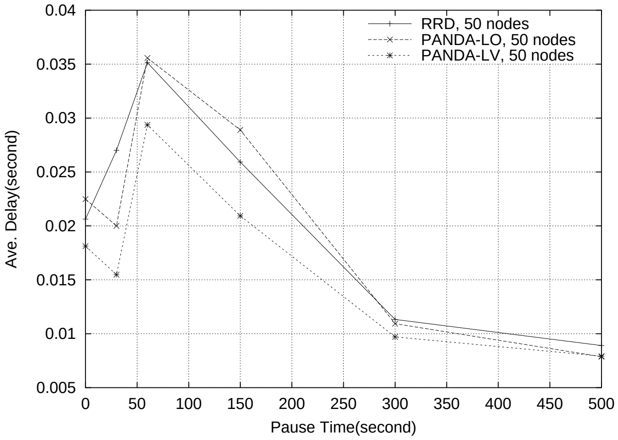
(a) 50 nodes



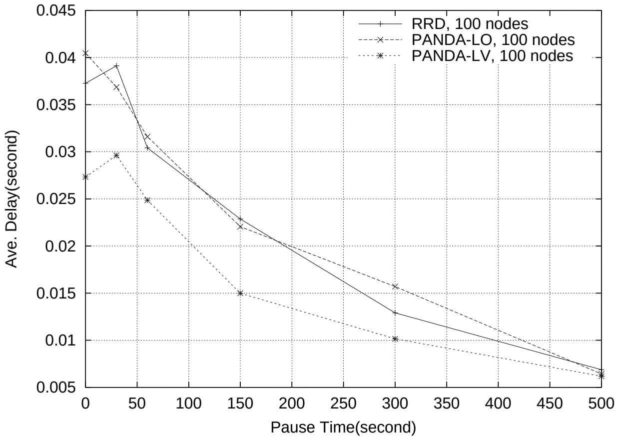
(b) 100 nodes

Figure 5. Comparison of path optimality ratio

Now, let us compare the end-to-end delays shown in Figure 6. In both 50 nodes and 100 nodes networks, PANDA-LV has lower end-to-end delay than RRD, while PANDA-LO does not show this improvement. This is due to the fact that PANDA-LO may lead to fragile routes without considering the link lifetime. According to the implementation of DSR, when a packet meets a route error, the intermediate node will try to rescue the packet with a locally cached route. If no such route is found in the local cache, the packet has to wait for a new route to be discovered. Hence fragile route in PANDA-LO will certainly increase the end-to-end delivery delay even though it has better path optimality as shown in 5. On the contrary, PANDA-LV approach can discover routes that are shorter in term of hop count, and longer-lived in term of link lifetime. Since the path has smaller number of hops, the packets will face less queuing delay waiting for wireless channel, comparing to that in RRD. Since the path is longer-lived, fewer route breakages will occur and thus data packets will face less buffering delay waiting for new routes. So PANDA-LV can achieve better end-to-end delay than RRD.



(a) 50 nodes



(b) 100 nodes

Figure 6. End-to-end delays

While shortest hop count and end-to-end delay are the major goal of PANDA-LO and PANDA-LV, we also want to see how they perform in terms of other metrics such as route error message, routing overhead and end-to-end delivery ratio. It would be desirable that PANDA doesnot sacrifice these metrics while attempting to discover a new route with shortest hop count. As shown in the next three figures (Figures 7, 8, and 9), our simulation results justify this great feature of PANDA.

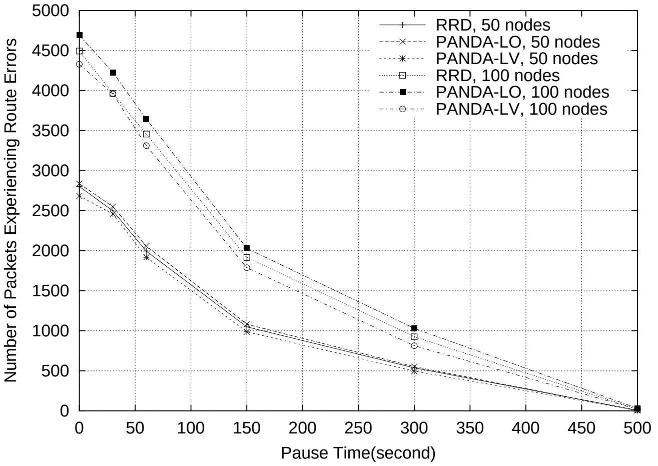


Figure 7. Route errors

Figure 7 shows the number of packets that experience route errors when they travel from source to destination. As implemented in DSR, when an intermediate node attempts to transmit a packet but ends up with getting a transmission error report from the data link layer, a route-error packet will be sent back to the original sender. In our simulation, the same packet will be recounted if it undergoes two or more route errors. We observe that PANDA-LV has a smaller number of route errors than RRD does, while PANDA-LO has a relatively larger number of route errors. This shows that PANDA-LO may lead to fragile paths because it does not consider the link lifetime in route discovery.

Figure 8 shows the routing overhead comparison. As our design, PANDA routing packets carry information about location and velocity of the previous hop node, it surely incurs a limited number of bytes of extra overhead per packet. It is well known that, in contention based multiple access wireless networks, it is more expensive to get a chance of transmission than transmitting a few more extra bytes in a packet. So we consider the measurement of the number of transmissions of routing messages. Every transmission of a route-request or route-reply message at each hop is counted once. We observe that routing overhead reduces as pausetime increases in all cases, which is natural since the mobility degree decreases. In both 50 nodes and 100 nodes networks,

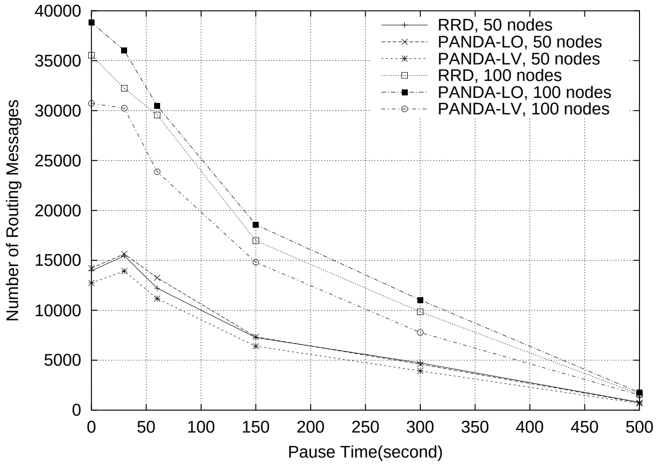


Figure 8. Routing overhead

PANDA-LV can reduce the routing overhead to some extend. This justified the analysis that PANDA-LV can find more stable route because it considers the link lifetime. On the contrary, PANDA-LO may incur more routing overhead since it may lead to fragile routes. We also observe that the tendency of the curves in Figure 8 is matched well with that in Figure 7, which shows the close relation between route errors and routing overhead. Whenever there is a route error, a new routing message will be initiated if there is no outstanding route-request message for this route breakage.

As shown in Figure 9, we observe that both PANDA-LO and PANDA-LV approaches can achieve almost the same packet delivery ratio as RRD. Here packet delivery ratio is defined as the number of received packets over the number of sent packets. An interesting observation is that, on the 50 node network scenario, PANDA-LO does not show much difference against RRD on route errors as shown in Figure 7. Accordingly, PANDA-LO performs as good as RRD on the 50 node network in Figure 9. However, on the 100 node network setup, PANDA-LO has a little increase in route errors, and accordingly, PANDA-LO has a little decrease in delievery ratio on the 100 node network. Overall, PANDA approaches achieve almost as good delivery ratio as RRD while it can improve the path optimality ratio (shown in Figure 5).

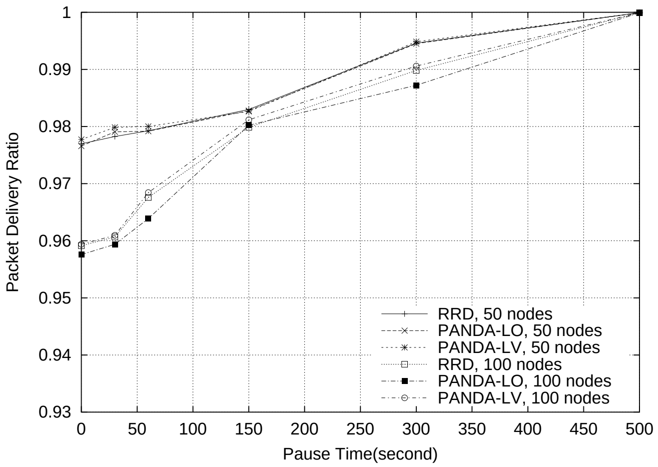


Figure 9. Delivery ratio

5.2 PANDA-TP

In this simulation we compare the performance of PANDA-TP and the Random Rebroadcast Delay (RRD) approach. We assume that sensor nodes can dynamically control their transmission power, which is not supported in the *ns-2* simulator. In PANDA-TP, we assume that when doing flooding based route discovery, an intermediate node will rebroadcast a duplicate route-request packet as long as it is from a path with less transmission power consumption. We also assume that the destination node does not respond to the first route-request packet immediately. Instead, it will wait for a small amount of time for multiple incoming route-request packets, and then choose the route with minimal transmission power consumption. Considering our simulation goal and the ease of implementation, we wrote our own discrete event simulation program, instead of modifying the *ns-2* simulator, to compare the performance of PANDA-TP and the RRD approach.

5.2.1 Simulation Setup

We only consider a static network topology. The simulation area is 1500×300 square meters. For different node density, we use 20, 40, 60, 80, 100 nodes. For each node density, we run multiple simulations with different connection numbers and obtain average results. For both RRD and PANDA-TP approaches, we assume that the wireless nodes can dynamically control their transmission range. In the route discovery phase, however, the nodes will use a fixed transmission range of 250 meters for broadcasting route-request packets. Once the route is discovered, an *en route* node will dynamically change its transmission range based on the link distance to the next-hop node.

5.2.2 Simulation Results

We consider two metrics in evaluating PANDA-TP's performance against RRD approach. The first one is the path energy ratio, which is defined as the power consumption of the route discovered by PANDA-TP over that of the route found by the RRD approach. The second metric is the routing overhead ratio, which is defined as the number of routing packets in PANDA-TP approach over the same parameter in the RRD approach.

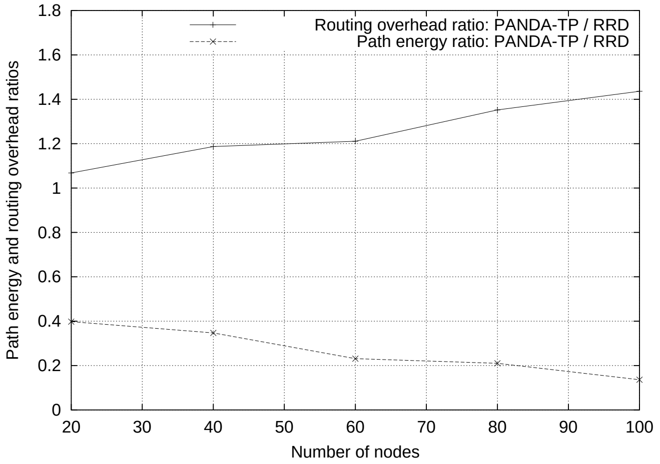


Figure 10. Energy conserving route discovery

As shown in Figure 10, the path energy ratio is as low as only 15%~40%, which translates to a huge saving of energy in sensor networks. We observe that as the number of nodes increases, the path energy ratio decreases. This means PANDA-TP can save even more power under high node density. The cost we pay is in term of routing overhead, which is about 10%~43% more than the RRD approach. This extra overhead increases as the network density increases. We argue that this cost is worth because the route discovery process is seldom executed in these static networks. Once the routes are found, they will be used to transfer data packets over a long period. So by greatly reducing the power consumption of data paths, we can prolong the overall system lifetime, even though we need to pay more in the route discovery phase.

5.3 PANDA-RB

We use ns-2 to evaluate PANDA-RB approach by comparing the performance with RRD approach. The goal of the PANDA-RB approach is to maximize the residual battery life of nodes in the network. This feature is highly desirable in cases such as sensor networks where recharging the battery is not an option. PANDA-RB is effective when the traffic pattern is non-uniform and there are one or more congested regions in the network.

5.3.1 Simulation Setup

To display the potential and effectiveness of PANDA-RB, we evaluate its performance in a non-uniform traffic scenario as shown in Figure 11. 100 mobile nodes are randomly placed in the network area of 1200x1200 square meters. There are 6 constant bit rate (CBR) connections in total, 3 of which take place between nodes in areas A and C, and the rest 3 between areas B and D. The source and destination nodes' movements are confined within each of these areas, respectively. Each connection randomly starts within the first 180 seconds and has a packet rate of 6 packets per second. Each mobile node has a transmission range of 250 meters. We adopt the energy model as implemented in the NS-2.1b8 version. In this model, each node has an initial energy level and a given energy usage on every packet it receives and transmits. The default value for the transmitting power as well as the receiving power is 281.8 mW. The energy consumption during the idle time is not considered in this model. In our simulations, we set the initial energy as 60.0 joules per node. The Random Waypoint mobility model is adopted and the speed range is between (0, 10) meters per second. Each run of simulation will last 1000 seconds. For different mobility degree, we use different pause-time of 0, 100, 200, 400, 600, 800, and 1000 seconds. For the threshold values in Algorithm 4, we choose $P_1 = 0.80$, $P_2 = 0.50$, $P_3 = 0.20$. For each pause-time, we run multiple rounds of simulations using different moving patterns and then obtain the average results.

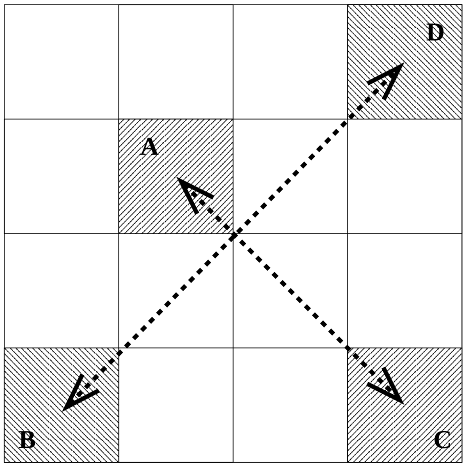


Figure 11. PANDA-RB v.s. RRD: Simulation setup

5.3.2 Simulation Results

We consider the minimal nodal lifetime, and the number of node the die during the simulation time. The minimal nodal lifetime is defined as the duration from the beginning of the simulation to the first time a node runs out of energy.

First, let us observe the minimal nodal lifetime shown in Figure 12. Under all the pause-times, PANDA-RB gives much longer lifetime than RRD approach. This justifies that PANDA-RB can balance the traffic load among different nodes and prolong the individual node's lifetime and hence the entire system lifetime.

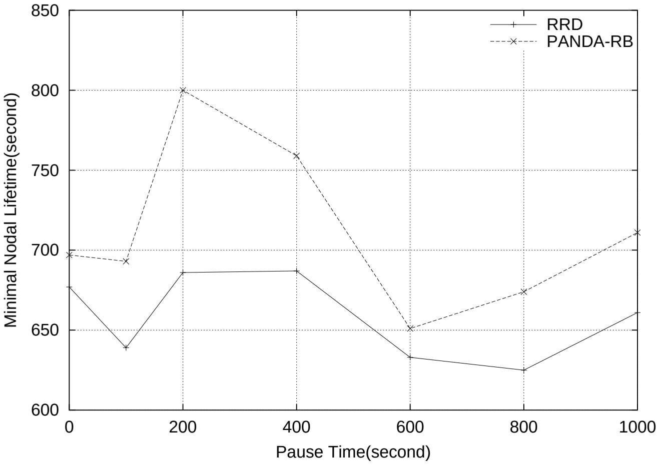


Figure 12. Minimal nodal lifetime

Second, Figure 13 shows the average number of nodes that die during the simulations. This can used to compare how fast the network loses mobile nodes due to battery outage. As can be seen, in all cases of different pause-times, PANDA-RD has smaller number of nodes that die compared to RRD approach. This again shows that PANDA-RB outperforms RRD approach in balancing battery utilization to prolong nodal lifetimes.

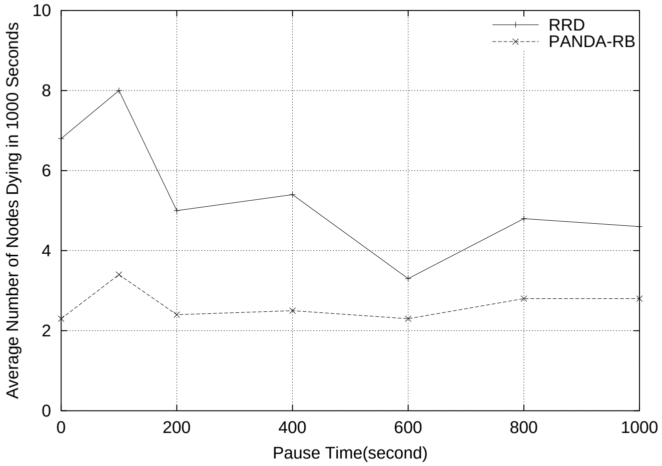


Figure 13. Number of dying nodes

6 Related Works

Flooding based techniques are used by a number of routing protocols in MANETs, such as DSR [4], AODV [5], LAR [7] and DREAM [8]. As discussed earlier, to avoid the problem of broadcast storm, DSR and AODV introduce a uniformly distributed random delay before rebroadcasting route-request packets and responding to a broadcast packet. To reduce flooding overhead, an intermediate node will drop duplicate route-request packets heard within a short period. Due to this reason, as described in [10], multiple routes discovered at the end of a route discovery process are nearly identical, except for some limited diversity in the last few hops preceding the destination. The authors of [10] proposed to use "diversity injection" technique to introduce diversity into the collection of route replies in query-based routing protocols. The basic idea is that intermediate nodes cache the partial routes found in the duplicate, to-be-dropped route-request packets. When later relaying a route-reply packet back to the route requester, the intermediate nodes will inject diversity into the route-reply packet by replacing the partial route from the local cache. To address the problem of high per packet overhead in source routing protocols such as DSR, "implicit source routing" technique was proposed in [11].

A variety of techniques have been developed to reduce the flooding overhead in on demand protocols. DSR aggressively utilizes route cache to reduce the routing overhead [12]. ZRP [6] is a hybrid protocol, where intra-zone routing is done with a proactive approach and inter-zone routing is done with an on-demand approach. For routing within a local neighborhood (intra-zone routing), ZRP uses a simple, timer-based Link State protocol. For destinations that are located beyond the local zone (inter-zone routing), ZRP uses a query-reply mechanism to discover route on demand. ZRP's inter-zone routing is dependent on a service called *bordercasting*, which allows the route-request packets to be directed to the current intermediate node's peripheral nodes. By utilizing *bordercasting* and appropriate query control mechanisms [14], ZRP can reduce the routing overhead compared to purely proactive link state or purely on-demand route discovery. Both LAR [7] and DREAM [8] attempt to utilize geographical location information to reduce the flooding overhead. They assume that mobile nodes can learn their locations via means such as Global Positioning System (GPS). Based on the location of an intermediate node and that of the destination node, packets (route-request packets in LAR, data packets in DREAM) can be broadcast into a restricted region instead of the whole network, hence reducing the routing overhead. More recently, a gossiping-based approach [13] was proposed to reduce the flooding overhead in ad hoc routing protocols, where each node forwards a route-request packet with some probability.

Location-aided rebroadcast delay has been investigated in [20] and [21], in which distance-based and location-based schemes are proposed to address the problem of broadcast storm in a mobile ad hoc network. A more comprehensive investigation on broadcast techniques in MANETs can be found in [22]. In cases where GPS is not available, signal strength can be used to determine the relative distance between nodes. In the work [23] this approach is adopted to achieve efficient broadcasting in ad hoc networks. Our proposal of PANDA is similar to [20, 21] in the sense that we also attempt to utilize location information to determine rebroadcast delay. But we use location information in a different manner. In our PANDA approaches, location information is used to determine if an intermediate is a good or bad candidate for the next-hop node. Additionally, we utilize velocity information to estimate the link lifetime, which can lead to the discovery of longer-lived end-to-end routes. Finally, the design goal of PANDA is different from that in [20, 21], whose goal is to reduce redundant rebroadcast messages while enjoying high reachability. In contrast, the goal of PANDA is to discover better routes in terms of desired QoS metrics such as smallest number of hops, lowest end-to-end delay, minimum transmission power, etc.

A number of power aware routing protocols have been proposed for wireless ad hoc networks. Like our PANDA-TP, PARO [26] assumes that the nodes can dynamically adjust their transmission range. PARO depends on *redirecting* technique to generate a path with a larger number of short-distance hops. According to PARO, an intermediate node will redirect the traffic of a direct communication between two other nodes via itself by inserting itself into the path whenever it determines that doing so will save overall transmission power consumption. Other protocols, such as [27], take residual battery capacity into consideration and attempt to avoid routes where many intermediate nodes are close to battery exhaustion. Similarly, the authors of [28] argued that always routing traffic through the minimal power path may drain out the batteries of certain nodes along the path, which in turn may disable further information delivery even if there are many nodes with plenty of energy. Aiming at maximizing the system lifetime as a whole, they proposed a set of algorithms which balance the energy consumption rates among the nodes in proportion to their residual energy. More recently, the work in [29] proposed a novel approach for power consumption calculation in the routing process, which takes link error rate and thus retransmission power consumption into consideration when attempting to minimize the overall energy of the path.

PANDA-TP shares the same goal as PARO in finding routes with multiple shorter-distance hops. Unlike PARO, PANDA-TP utilizes location information to determine rebroadcast delay in the route discovery process, which is targeted to choose close neighboring nodes as the next hop and hence reduce the overall transmission power of the path. Complementary to PANDA-TP, PANDA-RB chooses to consider battery lifetime in route discovery, where intermediate nodes with high residual battery capacity are given priority. A recent work similar to PANDA-RB can be found in [30], which uses a continuous function to determine the rebroadcast delay inversely proportional to the current node's residual battery capacity. Like other PANDA algorithms, PANDA-RB utilizes a categorization technique instead.

7 Conclusions

On-demand routing protocols in MANETs, such as DSR and AODV, often utilize flooding based techniques to discover new routes when needed. To avoid the problem of wireless broadcast storm, the random rebroadcast delay (RRD) approach is used in both DSR and AODV. This RRD approach, however, may lead to the "next-hop racing" phenomena. In this paper we have proposed a Positional Attribute based Next-hop Determination Approach (PANDA) to address the problem of "next-hop racing". In the route discovery process, PANDA attempts to utilize positional and energy information to determine the rebroadcast delay. Aiming at finding better end-to-end paths, PANDA was designed in such a manner that good candidates for the next hop will always go first and thus "win" in the route discovery process.

Through simulation studies, we evaluated the performance of PANDA algorithms. PANDA-LO algorithm considers only relative distance when deciding rebroadcast delay, while PANDA-LV takes into consideration both distance and link lifetime. Both PANDA-LO and PANDA-LV can achieve better path optimality than the RRD approach, while enjoying the same high end-to-end delivery ratio. Since PANDA-LO algorithm does not consider link lifetime, it may lead to fragile routes and thus does not improve the overall end-to-end delay. On the contrary, PANDA-LV algorithm attempts to choose shorter hops as well as longer-lived routes in the route discovery process. So PANDA-LV algorithm can improve both path optimality and end-to-end delay.

The PANDA approach can also be applied in searching routes in terms of other constraints such as transmission power consumption and residual battery capacity. Motivated by the fact that multihop routes have lower power consumption than a big single hop transmission, we designed PANDA-TP algorithm for static or low-mobility wireless networks such as sensor networks. Our simulation showed that at a moderate cost of increased routing overhead, PANDA-TP can lead to the discovery of data paths with as less as only 15%~40% transmission power consumption compared to the RRD approach. Since route discovery process is executed occasionally in static network topology, we can get high performance gain in term of power conservation with PANDA-TP approach. In addition, we developed PANDA-RB, which is a complementary mechanism to PANDA-TP. PANDA-RB takes into account the current intermediate node's residual battery capacity in route discovery. Simulation results show that PANDA-RB can prolong individual node's and hence the network's lifetime compared to the RRD approach.

Since PANDA algorithms rely on local decisions without any global knowledge, they are greedy heuristics in nature and cannot guarantee the discovery of the optimal end-to-end path. Keeping this in mind, we would like to point out that PANDA can achieve very good end-to-end performance, even compared to the centralized routing scheme.

In summary, the PANDA approach can be considered as a generic framework for improving the performance, quality, and energy conservation of routing algorithms in ad hoc networks.

Acknowledgment

This research was supported in part by the National Science Foundation through the grants CCR-0296070 and ANI-0296034.

References

[1] E.M. Royer and C.-K. Toh. A Review of Current Routing Protocols for Ad-hoc Mobile Wireless Networks. IEEE Personal Communications, Apr. 1999, pp.46-55.

[2] C. E. Perkins and P. Bhagwat, Highly Dynamic Destination-Sequenced Distance Vector Routing (DSDV) for Mobile Computers. Proc. ACM SIGCOMM'94, London, UK, Sep. 1994, pp. 234-244.

[3] V.D. Park and M.S. Corson. A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks. Proc. of IEEE INFOCOM '97, Kobe, Japan, April 1997.

[4] D.B. Johnson, D.A. Maltz, and J. Broch. DSR: The Dynamic Source Routing Protocol for Multi-Hop Wireless Ad Hoc Networks. Ad Hoc Networking, edited by C. E. Perkins, Addison-Wesley, 2001.

[5] C. E. Perkins and E. M. Royer. Ad hoc On-Demand Distance Vector Routing. Ad Hoc Networking, edited by C. E. Perkins, Addison-Wesley, 2001.

[6] Z. Haas and M. Pearlman. The Zone Routing Protocol: A Hybrid Framework for Routing in Ad Hoc Networks. Ad Hoc Networking, edited by C. E. Perkins, Addison-Wesley, 2001.

[7] Y-B Ko and N. H. Vaidya. Location-Aided Routing(LAR) in Mobile Ad Hoc Networks. ACM/Baltzer Wireless Networks (WINET) journal, Vol.6-4, 2000.

[8] S. Basagni, I. Chlamtac, V.R. Syrotiuk, and B.A. Woodward. A Distance Routing Effect Algorithm for Mobility (DREAM). Proc. of ACM MOBICOM'98, pp. 76-84.

[9] T.S. Rappaport. Wireless Communications Principles and Practice, Prentice Hall, 1996, pp.102-104.

[10] M. Pearlman and Z. Haas. Improving the Performance of Query-based Routing Protocols Through "Diversity Injection". Proc. of WCNC'99, New Orleans, LA, Sept. 1999.

[11] Y.-C. Hu and D.B. Johnson. Implicit Source Routes for On-Demand Ad Hoc Network Routing. Proc. of ACM MOBIHOC'01, pp.1-10, Long Beach, CA, Oct. 2001.

[12] Y.-C. Hu and D.B. Johnson. Caching Strategies in On-Demand Routing Protocols for Wireless Ad Hoc Networks. Proc. of ACM MOBICOM 2000, Boston, MA, August 2000.

[13] Z. Haas, J. Halpern, and L. Li. Gossip-based Ad Hoc Routing. Proc. of IEEE INFOCOM 2002.

[14] Z. Haas and M. Pearlman. The Performance of Query Control Schemes For the Zone Routing Protocol. IEEE/ACM Transactions on Networking (TON), Volume 9, Issue 4, August 2001, pp.427-438.

[15] B. Karp and H. T. Kung. GPSR: Greedy Perimeter Stateless Routing for Wireless Networks. Proc. of ACM MOBICOM 2000, pp.243-254.

[16] J. Broch, D. A. Maltz, D. B. Johnson, Y-C. Hu, and J. Jetcheva. A Performance Comparison of Multi-hop Wireless Ad-hoc Networking Routing Protocols. Proc. ACM MOBICOM '98, pp. 85-97, October 1998.

[17] P. Jain, A. Puri, and R. Sengupta. Geographical Routing Using Partial Information for Wireless Ad Hoc Networks. IEEE Personal Communications, 2001.

[18] C.-K. Toh. Associativity Based Routing For Ad Hoc Mobile Networks. Wireless Personal Communications Journal, Special Issue on Mobile Networking and Computing Systems, Vol. 4, No. 2, pp.103-139, March 1997.

[19] S. Marti, T. Giuli, K. Lai, and M. Baker. Mitigating Routing Misbehavior in Mobile Ad hoc Networks. Proc. IEEE INFOCOM 2001

[20] S-Y Ni, Y-C Tseng, Y-S Chen, and J-P Sheu. The Broadcast Storm Problem in a Mobile ad hoc Network. Proc. ACM MOBICOM 1999, pp. 151-162.

[21] M. Sun, W. Feng, and T. H. Lai. Location Aided Broadcast in Wireless Ad Hoc Networks. Proc. IEEE GLOBECOM 2001, pp. 2842-2846, San Antonio, TX, November 2001.

[22] B. Williams and T. Camp. Comparison of Broadcasting Techniques for Mobile Ad Hoc Networks. Proc. of MOBIHOC 2002, EPFL, Lausanne, Switzerland.

[23] X. Chen, M. Faloutsos, and S. Krishnamurthy. Distance ADaptive (DAD) Broadcasting for Ad Hoc Networks. MILCOM 2002.

[24] M. Mauve, J. Widmer, and H. Hartenstein. A Survey on Position-based Routing in Mobile Ad hoc Networks. IEEE Network Magazine, 15(6):30–39, November 2001.

[25] J. Yoon, M. Liu, and B. Noble. Random Waypoint Considered Harmful. In Proc. IEEE INFOCOM, vol 2, pp 1312-1321, April 2003, San Francisco, CA.

[26] J. Gomez, A.T. Campbell, M. Naghshineh, and C. Bisdikian. Conserving Transmission Power in Wireless Ad Hoc Networks. Proc. 9th International Conference on Network Protocols (ICNP 2001), Riverside, California, November 11 - 14, 2001.

[27] S. Singh, M. Woo, and C. Raghavendra. Power-Aware Routing in Mobile Ad-hoc Networks. Proc. of ACM MOBICOM, Oct. 1998.

[28] J. Chang and L. Tassiulas. Energy Conserving Routing in Wireless Ad Hoc Networks. Proc. IEEE INFOCOM 2000, pp. 22-31.

[29] S. Banerjee and A. Misra. Minimum Energy Paths for Reliable Communication in Multi-hop Wireless Networks. Proc. ACM MOBIHOC 2002, pp.146-156.

[30] W. Yu and J. Lee. DSR-based Energy-aware Routing Protocols in Ad hoc Networks. Proc. of ICWN 2002.

[31] N. Bulusu, J. Heidemann, and D. Estrin. GPS-less low cost outdoor localization for very small devices. IEEE Personal Communications Magazine, vol. 7, no. 5, pp. 28-34, Oct. 2000.

[32] A. Savvides, C. Han, and M.B. Strivastava. Dynamic finegrained localization in ad hoc networks of sensors. In Proc. of ACM MOBICOM, pp. 166-179, 2001.

[33] D. Niculescu, and B. Nath. Ad hoc positioning system using AoA. In Proc. of IEEE INFOCOM 2003, San Francisco, CA.

[34] X. Cheng, A. Thaeler, G. Xue, and D. Chen. TPS: a time-based positioning scheme for outdoor wireless sensor networks. In Proc. of IEEE INFOCOM, 2004, Hongkong, China (to appear).

[35] J. Wu and F. Dai. Broadcasting in Ad Hoc Networks Based on Self-Pruning. Proc. of INFOCOM 2003.

[36] The Network Simulator *ns-2*, available online at http://www.isi.edu/nsnam/ns/.

[37] The Monarch Group at Rice University, project website http://www.monarch.cs.rice.edu/.