An Application of Parameter Estimation to Route Discovery By On-Demand Routing Protocols

John Sucec and Ivan Marsic
Center for Advanced Information Processing
Rutgers University
jsucec/marsic @caip.rutgers.edu

Abstract[†]

To discover a route to a peer node, an on-demand routing protocol may initiate a flood-search procedure known as route discovery. By selecting the correct query radius, the number of packet transmissions required for route discovery can be minimized.

This paper presents methods to estimate the geographic radius (R_G) and the number of currently active pairs of communicating nodes (P) in a mobile ad hoc network. The methods are entirely distributed and incur little communication overhead. Network nodes can apply the estimated parameters to predict the probability mass function (PMF) of route discovery hop distance. An accurate prediction of the PMF aids the selection of an appropriate query radius for the route discovery process.

A computationally lightweight procedure to select an appropriate query radius, based only on an estimate of P, is also proposed. Simulation results show that this procedure facilitates a sensible trade-off between route request packet overhead and route reply delay.

1. Introduction

Network protocols based on reactive routing techniques have received considerable attention with regard to their applicability to mobile ad hoc networks (MANETs). Among the routing protocols that are based at least in part on reactive routing are the Ad hoc On-Demand Distance Vector (AODV) routing protocol [1,15] and the Dynamic Source Routing (DSR) protocol [2,16]. These particular protocols are reactive, or "on-demand", in nature because each proactively acquires little or no data regarding the network topology. Instead, when a node originates a packet to a destination for which no forwarding path is known, the originating node initiates a *flood-search* route discovery procedure.

Ignoring the implementation details of any specific flood-search on-demand routing protocol, a generalized description of the route discovery procedure is as follows. The procedure consists of an originating node, or source node (s), disseminating a route request (RREQ) packet that effectively queries recipient nodes for a route to the target node (t). (Assuming a shared media, broadcast transmission environment, the RREQ packet can be propagated by a node to each of its neighbors via a single broadcast transmission.) If the RREQ arrives either at t or at a node with a valid path to t in its route cache, a route reply (RREP) message is sent back to s. The RREP packet is returned along the reverse of the path traversed by the received RREQ packet until it arrives at s.

Now, the issue of substantial interest in this paper is setting the query radius (R_Q), measured in hops from s, to which the RREQ should be propagated. At most, R_Q may need to be set to D, the network diameter. Under this condition, the route discovery procedure is equivalent to network-wide broadcast (NWB) and may result in up to $|V|$-1 transmissions of the RREQ packet.[1] That is, every node save t itself propagates the packet. At the optimistic end of the spectrum, s may presume that with high likelihood a neighbor of itself has a route to t in its cache. If this is indeed the case, then a single transmission of the RREQ packet by s is sufficient. DSR specifies such a non-propagating query in its implementation and resorts to NWB if a RREP is not received by s within some maximum allowable non-propagating query duration [2].

In addition to setting R_Q to either 1 hop (a non-propagating or local query) or D hops (a NWB query), R_Q may be set to anything in between. A number of heuristics can be employed to determine R_Q. Among them include the expanding ring approach as detailed in [1] in application to AODV. In the expanding ring

---

[†] This work was supported in part by US Army CECOM Contract Number DAAB07-00-D-G505.

[1] Where, V is the set of nodes in a given network, E is the set of bi-directional links connecting nodes in V and $G = (V,E)$ is the undirected graph formed by V and E. $|S| \equiv$ Cardinality of the set S.

approach, if the allowable duration for the existing query is exceeded, R_Q is incremented by some amount and a new query is initialized. To be consistent with terminology of [1,2], the process of incrementally increasing the query radius and allowable query duration until the query scope reaches either t or a node with a route to t in its route cache, is referred to here as an *expanding ring search* (ERS). In particular, the ERS sequence of R_Q values of $\{1,2,D\}$ is considered in detail, herein. One of the performance measures employed here is the *expected number of RREQ packet transmissions* (ψ) required by the route discovery process. The performance of the $\{1,2,D\}$ R_Q sequence is compared with that of the $\{1,D\}$ R_Q sequence.

Although an ERS procedure that defers NWB for as long as possible may minimize the ψ metric under certain network conditions, it may also incur an unacceptable expected route discovery delay (τ). This occurs when each incremental increase in R_Q yields little additional likelihood of route discovery over the previous value of R_Q. Under such circumstances, it is desirable for the route discovery procedure to employ a larger R_Q increment or resort immediately to NWB in the event that the initial non-propagating query fails. This implies the need for a predictive method to select an appropriate value of R_Q. It is also desirable that the predictive method used to determine the R_Q increment or to determine whether the ERS process should be blocked altogether, incurs little additional control message overhead. Such a predictive method, that is also *distributed* and *asynchronous*, is detailed in Section 4.

The rest of the paper is organized as follows. Section 2 presents the framework upon which the simulation work described herein is based and briefly summarizes some related work. Section 3 reports simulation results that demonstrate the savings $\{1,2,D\}$ ERS can provide over a procedure that allows only $R_Q \in \{1,D\}$. Section 5 reports simulation results and assesses the performance of the predictive method described in Section 4. Section 6 proposes a simplified prediction method that is of greater practical value for the purpose of query scope selection. Lastly, Section 7 summarizes the key points of this paper.

2. Framework

2.1. Network Environment

The underlying network and link assumptions for this paper are as follows. The network topology is represented by a *connected* and *undirected* graph $G = (V,E)$. Every node is equipped with a transceiver whose transmission range is given by R_{TX}, in meters. All nodes within the transmission range of a node v will be able to hear transmissions by v. Likewise, v will be able to hear the transmissions of all nodes lying not more than R_{TX} from it. On the other hand, nodes lying more than R_{TX} from one another are assumed to be unable to communicate directly with one another and at least one intermediary node is required to forward packets between such pairs. The MAC protocol on all interfaces is some variant of CSMA/CA. It is assumed that receivers operating in promiscuous mode can tap the frames transmitted on the CSMA/CA media.

The network layer protocol can be any network protocol that supports both next-hop routing and source routing. Additionally, the network layer datagram header must provide a means to indicate the path length between s and t. This can be provided implicitly when source routing is in effect. When next-hop routing is in effect, a header extension or extra field is needed in the datagram header to indicate explicitly the path length. Transmission of path length information is needed to estimate the number of pairs of communicating nodes, as discussed in Section 4.

Each network node has a single CSMA/CA network interface card (NIC) and, therefore, each node can be uniquely identified by the address associated with that NIC. Further, it is assumed that each node is cognizant of $|V|$ and knows the node ID (i.e., the network address) of all network nodes. This information would possibly be pre-configured in network nodes and if a new node joins the network then a single NWB is performed for it to announce itself. (The new node would also have to query a neighbor to obtain node count and node ID information of the network nodes.) Presumably, the frequency of join (or departure) events will be sufficiently low so that overhead associated with tracking network node count and node identities does not contribute significantly to network traffic.

Lastly, it is assumed that the routing protocol includes a variant of the Hello protocol that allows each node to periodically announce itself to its neighbors and, therefore, discover its neighbors. Each Hello message contains not more than $3\times\lceil \log_2|V| \rceil$ additional bits to communicate data related to parameter estimation, as discussed in Section 4.

2.2. Routing Paradigms

Although AODV and DSR are cited in this paper, the specifics of their implementations are not essential for the purposes here. Instead, AODV and DSR are referenced primarily as representatives of two important classes of routing protocols: next-hop table-lookup routing and source routing, respectively. A third class of routing protocols is geographic routing which has received attention recently [3,4]. The geography-based routing paradigm will not be considered here.

The reason why next-hop routing and source routing deserve separate consideration in terms of route discovery is because their potentials for passively acquiring topology information via tapping of incident packets can be quite different. Whereas a next-hop routed packet contains only the source and destination addresses in the datagram header, a source routed packet contains also the addresses of the intermediary nodes of the source to destination path specified in the datagram header. Thus, when nodes operate their NICs in promiscuous mode, as assumed in the simulation results of [5,6,7], a node tapping the network layer headers of packets transmitted by its neighbors will potentially learn significantly more topology information if the network employs a source routing protocol than it would if a next-hop routing protocol was employed. Since such passively acquired topology data can be used in the route discovery process, source routing protocols have a greater potential for efficient route discovery than next-hop routing protocols. This statement is not intended to reflect on the overall goodness of one routing paradigm versus the other.

2.3. Simulation Assumptions

For the simulations reported in Sections 3, 5 and 6, network nodes are situated randomly throughout a network area of fixed size in accordance with a two-dimensional uniform probability distribution. To assess the probability of route discovery, it is assumed that the network topology remains fixed over the duration of any given route discovery procedure. This topology *snapshot* assumption is valid under the condition where node mobility is sufficiently modest such that the network link state represented by E is unlikely to change over the duration of a single route discovery event. This is a reasonable assumption to make. Otherwise, if the ratio of node speed to R_{TX} while route discovery is taking place is high enough to cause significant changes to E, then any routing information conveyed by a RREP is likely to be obsolete by the time it reaches s. The snapshot model, therefore, relaxes the simulator requirements, as mobility effects need not be considered for the purposes of assessing route discovery overhead when E is constant.

Three other simulation assumptions concern the communication sessions between pairs of nodes. One, mentioned already in Section 2.1, is that each packet contains the path length, of the route used between s and t, in its datagram header. Another is that each communication session consists of unicast communication(s) between a unique (unordered) pair of nodes whose datagrams are forwarded over a single, least-hop, bi-directional network path (i.e., alternate paths not used). Lastly, for each communication session, packets are originated at a steady rate so that neighbors of nodes along an active path are able to use the overheard packet transmission to refresh their route caches.

Avoiding large inter-packet times can be facilitated without overhead if the communication session is for a CBR application. In the case of a bursty communication session, s can originate "heartbeat" packets to notify tapping nodes along an active path that the topology data they have cached about the route is still fresh. Implied in the consideration here of the communication sessions, is the notion of avoiding stale route cache entries. That is, the nature of the communication sessions ensure that topology data obtained via packet tapping is valid, provided it is *purged* from the route cache when the communication session, from which it was learned, is either terminated or resorts to a different path. Purging route cache entries when the associated communication session is considered no longer active is consistent with the *Link-Static-X* cache purging approach analyzed in [7], where X is the expiration timeout period.

2.4. Earlier Work

A considerable number of simulation studies have been conducted to assess the performance of various MANET routing protocols that have on-demand characteristics. Among these include [5,8,9,10]. Further, algorithms and heuristics have been proposed for alleviating the effect of network-wide broadcast in MANETs. Among these include [11,12,17]. However, none of this earlier work has assessed the performance of an ERS-like approach to reducing routing protocol overhead or considered distributed parameter estimation techniques to predict actual network conditions.

Reference [13] proposes a gradually expanding request zone in order to reduce the likelihood of NWB events. Unlike [13], however, the methods of this paper do not presuppose the presence of GPS enabled nodes or the availability of GPS data in the route discovery process.

Heuristics for query containment that exploit knowledge of a previously known valid path to a target node t are proposed in [14]. The underlying premise of these techniques is that many of the nodes lying on a previously valid path are likely to be still useful for constructing a new path to t and yields a localization of the query process when node mobility has not drastically disrupted the earlier known topology. This approach appears to be very promising for the purpose of *route maintenance*. However, in cases where no path to t is previously known or when path information becomes stale as a result of node mobility, such query localization techniques are ineffective. The methods of this paper, on the other hand, do not presuppose knowledge of topology data specifically related to finding a route to a particular target node. Thus, this paper addresses query containment issues beyond the scope of [14].

3. ERS Performance

To assess the benefit of an ERS approach to route discovery, RREQ packet overhead for when $R_Q \in \{1,2,D\}$ (two ERS increments or *2-ERS*, 0→1→2 hops) is compared with that generated for when $R_Q \in \{1,D\}$ (single ERS increment or *1-ERS*, 0→1 hop). Table 1 indicates the network scenarios under which the *2-ERS* and *1-ERS* cases were tested (for both the next-hop routing and source routing paradigms). For each scenario, the network area is a circle with a geographic radius indicated in column 4. For each scenario, and all other simulations reported herein, R_{TX} = 250m. All other network conditions are as detailed in Sections 2.1 and 2.3.

Table 1

Scenario	No. Nodes	No. Trials	Radius
1	50	200	500m
2	100	100	700m
3	200	50	1000m

Fig. 1-3 report the performance of the *2-ERS* and *1-ERS* approaches. In each scenario, the *number* (P) of active communication sessions (i.e., communication paths) was varied from 0 to $|V|$ and this corresponds to the x-axis of the plots. Only trials where route discovery occurred at $R_Q \geq 1$ hop were counted toward the evaluation of ψ.

It is interesting to note that next-hop routing outperforms source routing, slightly, for the scenario of a 50-node network (Fig. 1). This is because for such a relatively small network node count, a cached route for *t* is very often available at *s* itself (i.e., zero hops from *s*). This is particularly true for source routing where the combination of complete path information in datagram headers and the relatively small network diameter means that *s* will, with high probability, have a cached route for *t* if *t* lies on any active communication path. Consequently, the likelihood that cached information is available at a node situated one hop away from *s*, but not at *s* itself, *may* be smaller for a source routing implementation than for a next-hop routing implementation, in small networks. This is true for Scenario 1. That is, next-hop routing has a larger incremental gain in the probability of cached routing information when expanding the search from zero hops to one hop, for Scenario 1. Since only trials where route discovery occurred at least one hop away from *s* were considered in Scenarios 1-3, next-hop routing *appears* to have a performance advantage over source routing in Fig. 1. However, for the 100-node and 200-node scenarios, where the probability of useful routing information being cached at *s* is much smaller, the benefit of a source specified packet-forwarding path in datagram headers is clearly evidenced in Fig 2 and 3, respectively.

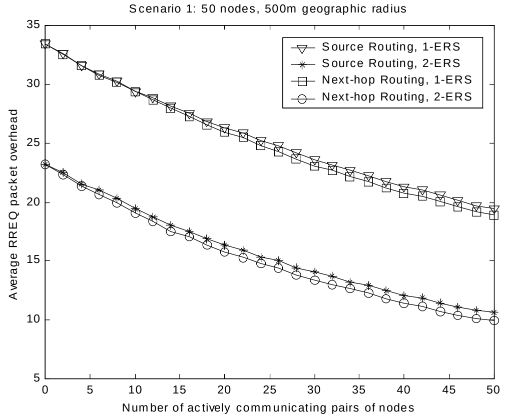

Figure 1. 50 nodes, ψ versus P.

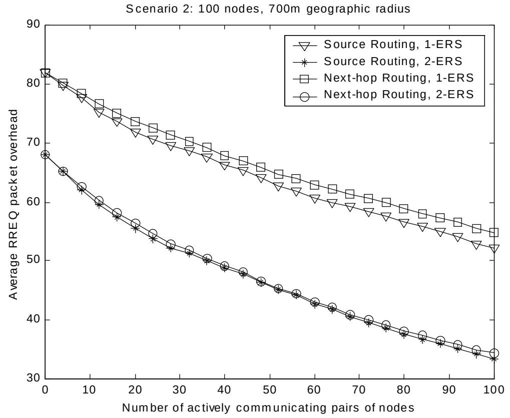

Figure 2. 100 nodes, ψ versus P.

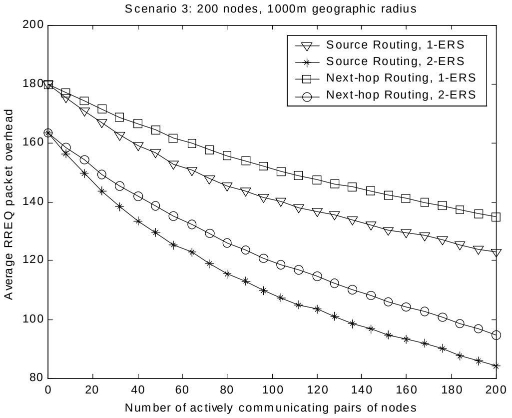

Figure 3. 200 nodes, ψ versus P.

4. Predictive Method

The simulation results reported in Section 3 demonstrate the advantage of *2-ERS* over *1-ERS* in terms ψ. However, the reduction in RREQ packet overhead is not enjoyed for free. In trials where a 2-hop query radius fails to yield a RREP for a path to t, R_Q must be set to D and a NWB is initiated just as it would have been had no ERS or *1-ERS* been employed. This is undesirable for two reasons. (i) It results in 1 + *deg(s)* additional RREQ packet transmissions that are avoided in the *1-ERS* approach (where *deg(s)* is the number of neighbors of s). (ii) It incurs additional route discovery delay as s has waited for a 2-hop query timer to expire before initiating the NWB. The performance trade-off between ψ and τ is also noted in [14].

Thus, it is desirable that s employs some predictive method to determine when it can "safely" apply *2-ERS*. The techniques proposed are based on the observation that under the condition of a uniformly random node distribution, the probability of route discovery at various hop counts from s is a function of two network parameters (in addition to $|V|$ and R_{TX}, of course). One is the geographic radius (R_G) of the network measured in meters. The other is P, the number of currently active communication sessions in the network.

Combining the estimates of R_G and P with its knowledge of $|V|$ and R_{TX}, s then computes locally a network simulation consisting of Monte Carlo trials of a random network, similar to the simulations used to generate the figures of Section 3. Here, however, the simulation is used to predict the expected probability mass function (PMF) for route discovery at various hop counts from itself. To reduce simulation overhead, PMFs for likely values of R_G and P may be *computed in advance* and saved in a database dedicated to the selection of R_Q.

Letting H be a discrete random variable denoting the number of hops away from s at which a route to t is discovered, the PMF of H, therefore, is given by $f_H(h)$, $h \in \{0,1,\ldots,D\}$. Therefore, the simulation consisting of Monte Carlo trials that is performed at s is a prediction of $f_H(h)$. Having predicted $f_H(h)$, s can then decide whether it is prudent to perform *2-ERS* by checking whether the likelihood of a successful 2-hop query exceeds some minimum probabilistic threshold:

$$\frac{f_H(2)}{\sum_{h=2}^{h=D} f_H(h)} > \Lambda \qquad (1)$$

If the quotient on the left-hand side of (1) exceeds Λ then s presumes that the likelihood of a successful 2-hop query is sufficiently high and performs *2-ERS*. Otherwise, s resorts immediately to NWB in the event that the non-propagating 1-hop query fails.

Now, a means for s to estimate R_G and P based solely on locally acquired information is described. The estimate of R_G is straightforward. First, each network node v counts the number of neighbors it has, *deg(v)*. This is obtained easily from the Hello protocol running at each node. The local estimate of R_G, ρ, is given by:

$$\rho = \sqrt{\frac{|V|}{1+\deg(v)}} \cdot R_{TX} \qquad (2)$$

The estimate of P is more complex. First, v counts the number of currently active paths of which it has learned, P', either from tapping of packets transmitted by its neighbors or because v itself lies on the path. Now the following are defined:

- $\Pi \equiv$ The set of paths known to v.
- $\Pi_k \equiv$ A path belonging to the set Π where $k \in \{1,2,\ldots,P'\}$.

Each Π_k is denoted by an ordered list of nodes $\{v_1,v_2,\ldots,v_K\}$ where v_1 and v_K correspond to the pair of nodes communicating via path Π_k and $K = |\Pi_k|$. For each path belonging to Π, v computes the fraction of nodes lying on the path that belong to the set of local nodes $L = \{v\} \cup N(v)$, where $N(v)$ is the set of nodes lying exactly one hop from v. The local estimate of P, π, is then computed as follows:

$$\pi = \frac{|V|}{1+\deg(v)} \cdot \sum_{k=1}^{P'} \frac{\left|(\{v\} \cup N(v)) \cap \Pi_k\right|}{|\Pi_k|} \qquad (3)$$

To facilitate a more robust estimate of R_G and P, network nodes exchange their degree information and local estimate of P, π, with each of their neighbors via the periodic messaging of the Hello protocol. Each node v then applies the average of the degree of all nodes in L to the computation of ρ', as given by (4).

$$\rho' = \sqrt{\frac{|V|}{1+\frac{1}{|L|}\cdot\left(\deg(v)+\sum_{u\in N(v)}\deg(u)\right)}} \cdot R_{TX} \qquad (4)$$

Similarly, nodes combine their own estimate π with the most recently received estimates from its neighbors to obtain a new estimate π', as given by (5).

$$\pi' = \frac{1}{1+\deg(v)} \cdot \left(\pi + \sum_{u\in N(v)} \pi_u\right) \qquad (5)$$

Where π_u corresponds to the estimate of P computed at the neighboring node $u \in N(v)$. These revised estimates,

ρ' and π', are used as the geographic radius and path count parameters, respectively, in each node's locally computed route discovery simulation.

In order for the local estimates of ρ and π to be communicated via the periodic Hello message, two additional Hello message fields are required. The field for *deg(v)* can be accommodated by a $\lceil \log_2|V| \rceil$ bit field, as no node will have can have degree greater than $|V|-1$ in a $|V|$ node network. Similarly, the field for π can be accommodated by a $\lceil 2\times\log_2|V|-1 \rceil$ bit field as there can not be more than $|V|\times(|V|-1)\div 2$ pairs of communicating nodes in a $|V|$ node network. Thus, the additional overhead per Hello message required to support the exchange of local estimates of ρ and π is easily upper bounded by $3\times\lceil \log_2|V| \rceil$ bits. This figure represents modest per message overhead.

The calculation of ρ' and π' and subsequent network simulation at a node *v* to predict $f_H(h)$ is collectively referred to hereafter as the *prediction agent* (*PA*). It is envisioned that the *PA* or its simplified version, proposed in Section 6, will be incorporated by the routing protocol to aid in determining the extent to which ERS should be employed in the route discovery process.

5. Performance of Predictive Method

5.1. Simulation Results

The simulation scenarios used to evaluate the *PA* of Section 4 correspond to Scenarios 1 and 2 outlined in Section 3. However, the performance metric employed in this section, to be defined shortly, is different. The following definitions are relevant to the metric:

- E_+ ≡ Event that occurs when (1) is satisfied for some threshold $\Lambda = \Lambda_+$.
- E_- ≡ Event that occurs when (1) is *not* satisfied for some threshold $\Lambda = \Lambda_-$.
- $\mathbf{E_C}$ ≡ Event that the *PA* at *v* correctly decides that either E_+ or E_- has occurred.
- $\mathbf{E_{C,+}}$ ≡ Event that the *PA* at *v* correctly decides that E_+ has occurred.
- $\mathbf{E_{C,-}}$ ≡ Event that the *PA* at *v* correctly decides that E_- has occurred.
- ε_+ ≡ Efficiency gain threshold for E_+ to occur.
- ε_- ≡ Efficiency gain threshold for E_- to occur.
- Λ_+ ≡ Optimistic approximation for the probability that a savings of at least ε_+ will occur, on average, when *2-ERS* is used rather than *1-ERS*.
- Λ_- ≡ Optimistic approximation for the probability that a savings less than ε_- will occur, on average, when *2-ERS* is used rather than *1-ERS*. ($\Lambda_- \le \Lambda_+$)
- Λ_{prob} ≡ Probability threshold for the *PA*: Use *2-ERS* when (1) is satisfied for the threshold Λ_{prob}. Otherwise, use *1-ERS*.

Thresholds Λ_+ and Λ_- are set as given by (6). Details for the derivation of (6) are available in the Appendix.

$$\Lambda_+ = \frac{\deg(v)+|V|-(1-\varepsilon_+)\cdot(|V|-1)}{|V|-1} \qquad (6a)$$

$$\Lambda_- = \frac{\deg(v)+|V|-(1-\varepsilon_-)\cdot(|V|-1)}{|V|-1} \qquad (6b)$$

The thresholds specified in (6) are optimistic because implicit in the relationships given between these thresholds and ε_+ and ε_- is the assumption that a NWB event results in $|V|-1$ RREQ packet transmissions. This assumption exaggerates the negative effect of NWB because typically many RREQ packet transmissions are suppressed. That is, when a copy of the packet arrives at a node *u* for which a route to *t* is known, *u* will transmit a RREP packet rather than propagate the RREQ. Although the actual values for Λ_+ and Λ_- do effect the performance of the *PA*, the fact that the equations of (6) happen to be optimistic does not invalidate the assessment of how accurately the *PA* decides whether E_+ or E_- has occurred.

The threshold Λ_{prob} is set as follows:

$$\Lambda_{\text{prob}} = \Lambda_-^{\mu} \qquad (0 < \mu < 1) \qquad (7)$$

The decision to make Λ_{prob} a power of Λ_- rather than a linear function of Λ_+ and Λ_- was based on a conjecture that Λ_{prob} should be set higher (i.e., more pessimistically) at nodes with small degree. This is to compensate for the fact that the threshold set by (6b) will be set higher at nodes with large degree than at nodes with small degree.

In Table 2, N_+, N_-, $N_{C,+}$, $N_{C,-}$ and N_C correspond to the number of times the E_+, E_-, $\mathbf{E_{C,+}}$, $\mathbf{E_{C,-}}$ and $\mathbf{E_C}$ events occurred, respectively, for the simulation scenario. For Scenario 1 (50 nodes), a total of 160 trials were performed. 40 trials were performed for Scenario 2 (100 nodes). In each scenario, *P* was varied from 0 to $|V|$.

Table 2

Scen-ario	Routing Scheme	N_+	N_-	$N_{C,+}$	$N_{C,-}$	N_C
1	Source	774	818	460	473	933
1	Next-hop	803	681	524	385	909
2	Source	280	153	248	96	344
2	Next-hop	275	112	247	92	339

Table 3 reports the parameter settings employed in each case. The performance gain (*g*) is also reported, where *g* is given by:

$$g = 100 \cdot \frac{N_C - \max\left(N_+, N_-\right)}{\max\left(N_+, N_-\right)}\% \qquad (8)$$

Eq. (8) provides an indication of the performance improvement yielded by the *PA* over what would have been experienced had *v* simply either always blocked *2-ERS* or always used *2-ERS*.

Table 3

Scenario	Routing Scheme	$\varepsilon+$	$\varepsilon-$	μ	*g*
1	Source	0.6	0.2	0.7	14.1%
1	Next-hop	0.6	0.2	0.7	13.2%
2	Source	0.4	0.15	0.75	22.9%
2	Next-hop	0.4	0.15	0.75	23.3%

5.2. Discussion of Results

As indicated in Tables 2 and 3, the *PA* provides a degree of control over the performance tradeoff between ψ and τ. Further, for the thresholds ε_+ and ε_- set to determine the E_+ and E_- events, respectively, applying parameter estimation to predict route discovery probabilities yielded a moderate performance gain.

An observation of the *PA*, not evident in the scenarios reported, is that it is sensitive to the settings of ε_+, ε_- and μ. This is a mixed blessing. On one hand, an improperly specified parameter can result in worst performance than simply following a default decision. On the other hand, when parameters are set correctly, a node *v* has great flexibility in its decision making process when E_+ and E_- events are not equally weighted. For example, if E_- events are weighted substantially more then E_+ events, ε_+, ε_- and μ would be set higher than they would if the two events were equally weighted. (Such an implementation would facilitate a more frequent blocking of *2-ERS*.) Nevertheless, devising a means for nodes to intelligently set ε_+, ε_- and μ, based on limited knowledge of the network environment, represents an open issue.

Another issue with the *PA* is that it requires a considerable amount of CPU time. That is, once a node estimates R_G and *P*, it then performs a number of Monte Carlo trials, which are averaged to yield a prediction of $f_H(h)$. For the results reported herein, 20 Monte Carlo trials were run locally at a node *v*.

Table 4 shows the incidence of E_+ and E_- events for some of the values of *P* under consideration in the simulation of Scenario 1, for the source routing case. Clearly, E_+ events occur considerably more frequently for large values of *P* than when *P* is small. The opposite is true for E_- events. This is consistent with the results of Fig. 1. Thus, an alternative predictive method is to simply apply the estimate π' of the parameter *P* to provide a node with a sense of whether it should employ or block *2-ERS*. For example, a node may use the following simple decision criteria: Employ *2-ERS* if π' exceeds some threshold and block, otherwise. Such a decision methodology would also be consistent with the prevailing network traffic conditions. That is, if π' suggests that *P* is large, then this also suggests that perhaps the network traffic volume is relatively heavy, as well. This inference is correct when all communication sessions, in operation over the network lifetime, generate traffic at approximately the same rate. Under conditions when network traffic volume is heavy, incurring a larger τ may be acceptable if it helps to reduce ψ, and therefore, reduce the likelihood of congestion events. A *2-ERS* decision method based on the value of π' represents a simple means to facilitate this trade-off.

Table 4

P	E_+	E_-	*P*	E_+	E_-
0	17	46	26	30	27
4	18	41	30	31	29
8	16	41	34	30	26
12	18	35	38	39	26
16	19	32	42	49	28
20	21	32	46	52	26
24	27	27	50	54	24

6. A Simplified Prediction Method

6.1 Simplified Procedure (*simple-PA*)

As indicated by Table 4, an increase in E_+ tracks closely the increase in communication sessions *P*. Conversely, an increase in E_- corresponds closely with a decrease in *P*. This suggests that a *simplified prediction agent* based on an estimate of *P* may be a viable means to facilitate a trade-off between ψ and τ. The simplified prediction agent is denoted here as the *simple-PA*.

An estimate for *P* is performed as described in Section using (5) to compute π'. The *simple-PA* then compares π' with a path count threshold (Λ_P) as follows:

$$\pi' > \Lambda_P \qquad (9)$$

If (9) is satisfied, a node *s* will perform *2-ERS*. Otherwise, it will block *2-ERS* and perform a NWB upon failure of a non-propagating query to yield a RREP message in a timely manner. Eq. (9) effectively replaces the threshold test given by (7).

6.2 Evaluation of *simple-PA*

The simulation assessment for evaluating the performance of the *simple-PA* is nearly identical to that performed for the *PA* as detailed in Section 5.1. Specifically, the definitions for the event types, thresholds, and expressions (6) and (8) all apply here, as well. Eq. (7), of course, is replaced by (9).

However, because the *simple-PA* is far less computationally costly than the *PA* of Section 4 is, it was possible to consider all three of the scenarios given by Table 1. Further, more trials were performed for each scenario. Table 5 summarizes the performance of the *simple-PA* for a particular set of ε_+, ε_-, and Λ_P.

Table 5

Scenario	**Routing Scheme**	ε_+	ε_-	Λ_p	***g***
1	Source	0.6	0.2	20	26.1%
1	Next-hop	0.6	0.2	20	21.1%
2	Source	0.35	0.25	50	24.1%
2	Next-hop	0.35	0.25	50	41.5%
3	Source	0.3	0.1	100	26.4%
3	Next-hop	0.3	0.1	100	69.1%

It is significant to note that the *simple-PA* actually outperforms the *PA* of Section IV. This indicates that distributed prediction of the route discovery hop distance PMF, $f_H(h)$, is a difficult objective to achieve, in practice. It suggests further that the optimal choice, between *1-ERS* and *2-ERS*, depends heavily on the parameter, *P*.

6.3 Packet Transmission Overhead and Delay

Figures 4-7 report the performance of the *simple-PA*, in terms of ψ and *hop delay*, for the source routing cases. The simulation scenarios are the same as considered in Section 6.2. Results for the next-hop routing cases were similar. Here, hop delay (η) is defined as the total number of hops that must be searched in the route discovery procedure before a RREP is transmitted back to *s*. As an example, a case is considered where the initial RREP is generated by a node situated 5 hops away from *s*. If *1-ERS* is in effect, η will be 5 (hops searched during NWB) + 1 (due to failed 1-hop query) = 6 hops. If *2-ERS* is in effect, η will be 5 (hops searched during NWB) + 1 (due to failed 1-hop query) + 2 (due to failed 2-hop query) = 8 hops. 800 trials were run for the 50-node case, 250 trials for the 100-node case and 100 trials for the 200-node case. Presuming τ is proportional to η, facilitating a trade-off between ψ and η is equivalent to facilitating a trade-off between ψ and τ.

The top portions of Fig. 4 (50 nodes) and Fig. 5 (200 nodes), and Fig. 6 (100 nodes) in its entirety, show that the *simple-PA* responds effectively to increasing *P* to reduce ψ. Focusing on Fig. 6, as an example, the savings in terms of ψ that would be achieved by *2-ERS* versus *1-ERS* are less than 25% when *P* is in the range of 0 to 28 pairs. Under these conditions, the *simple-PA* follows closely the performance curve of *1-ERS*, indicating that the *simple-PA* has blocked *2-ERS* in nearly all trials. In the range of 32 to 72 communicating pairs, the *simple-PA* elects with increasing frequency to apply *2-ERS*. In this range, the savings of *2-ERS* versus *1-ERS* vary from 25.1% to 32.6%. This represents a "transition interval" as the efficiency gain afforded by *2-ERS* is between the efficiency gain thresholds $\varepsilon_- = 0.25$ and $\varepsilon_+ = 0.35$. Over the range of 76 to 100 communicating pairs, the *simple-PA* employs *2-ERS* nearly all the time. Here, the savings afforded by *2-ERS* are in excess of 33%. Lastly, not only does the percent reduction in ψ grow with increasing *P*, but the absolute count in packet savings also grows as *P* increased from 28 or fewer pairs, to the transition interval of 32 to 72 pairs and then to 76 or more pairs. That is, the fractional reduction in ψ here is *not* simply due to the packet overheads associated with *1-ERS* and *2-ERS* both being driven towards zero with increasing *P*.

The bottom portions of Fig. 4 and Fig. 5, and Fig. 7 (100 nodes) in its entirety, show the control over η that is achieved by the *simple-PA*. Focusing on Fig. 7, as an example, a savings in η of more than one hop is afforded by the *simple-PA* versus *2-ERS* when *P* is in the range 0 to 12 pairs. This significance of this is as follows. Supposing the query waiting time for a RREP to be 30ms per hop (as an example), then the *simple-PA* has successfully avoided incurring an additional 30ms delay, on average, in route discovery that would have occurred had the route discovery process defaulted to *2-ERS*.

As discussed already in relation to Fig. 6, the *simple-PA* suppressed *2-ERS* in nearly every trial over the range of 0 to 28 communicating pairs. For large *P* (i.e., 76 to 100 communicating pairs), the *simple-PA* applies *2-ERS* in nearly every trial, but here, the penalty, in terms of η, of employing *2-ERS* as opposed to *1-ERS* is less than 0.63 hops. (The penalty due to the *simple-PA* is less than 0.58 hops.) The savings in terms of ψ provided by *2-ERS* over this range, justifies its use under these conditions. The *simple-PA* exploits this to achieve savings of more than 17.4 packet transmissions over this range.

Lastly, the transition interval of Fig. 7 is punctuated by a kink in the hop delay curve corresponding to the *simple-PA* performance. That is, with increasing *P*, the *simple-PA* will transition from applying *1-ERS* to applying *2-ERS*. Despite the kink in the η curve of the *simple-PA*, the performance of the *simple-PA*, in terms of η, ranges only from 0.06 to 0.57 hops worse than that achieved by *1-ERS*, over the transition interval. This moderate penalty is compensated for by progressively better performance in terms of ψ, as shown for the transition interval of Fig. 6.

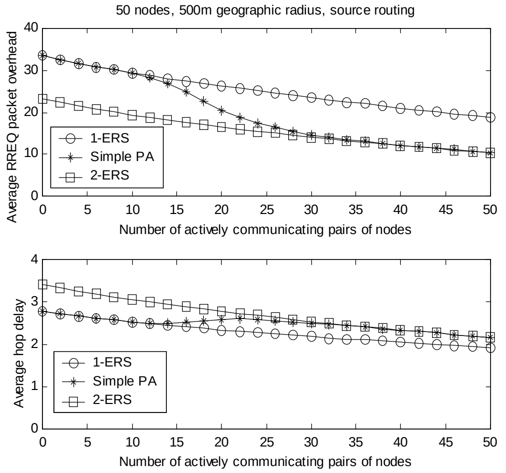

Figure 4. 50 nodes, ψ (top) and η (lower) versus *P*.

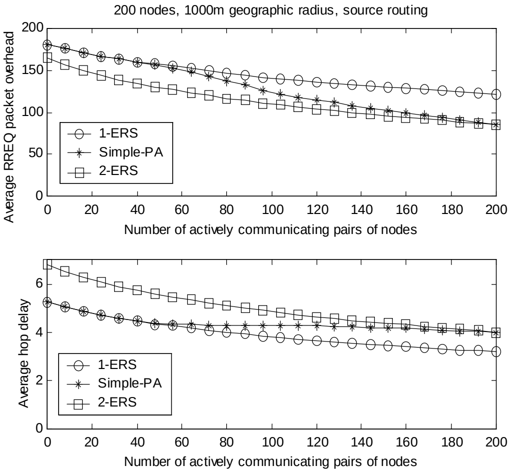

Figure 5. 200 nodes, ψ (top) and η (lower) versus *P*.

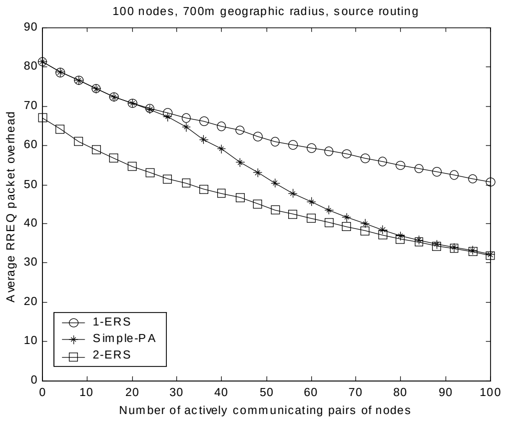

Figure 6. 100 nodes, ψ versus *P*.

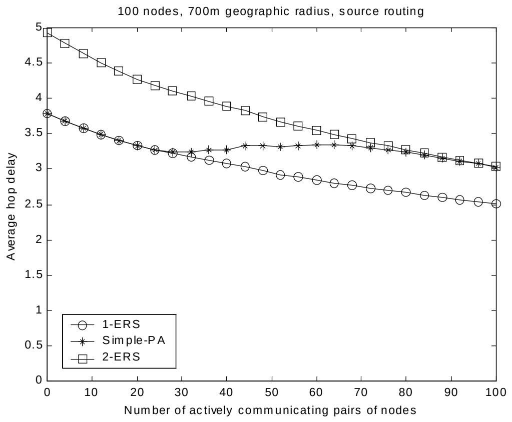

Figure 7. 100 nodes, η versus *P*.

7. Conclusions

ERS represents a potentially powerful means to reduce route discovery packet overhead in on-demand routing protocols. The results reported here indicate that the savings due to employing *2-ERS* rather than *1-ERS* can be up to 45%, on average. Exploiting the benefits of ERS without incurring excessive route discovery delay, however, is an issue. It is suggested here that distributed parameter estimation may offer a solution.

It has been shown that by estimating the geographic radius (R_G) and the number of communicating pairs of nodes (*P*), a degree of control over the performance trade-off between RREQ packet overhead (ψ) and route discovery latency (τ) can be achieved by incorporating them within the context of a prediction agent (*PA*). The *PA* proposed here incorporates the estimates of R_G and *P* to predict the route discovery hop distance PMF ($f_H(h)$). The simulation results reported herein indicate that the *PA* is responsive to the trade-off between ψ and τ. Further, the estimates of R_G and *P* are computed in a completely distributed fashion, require only local topology knowledge and incur little communication overhead.

There are a number of issues, however, confronting the current implementation of the *PA*. First, its performance is very sensitive to the setting of thresholds (ε_+, ε_-, Λ_+ and Λ_-) and μ. Second, Monte Carlo simulations to predict $f_H(h)$ can be very costly in terms of computation time and possibly also in terms of energy consumption (although, these processor costs may be offset by pre-computing a PMF database). Third, and most important, accurate prediction of $f_H(h)$ is difficult to achieve. More robust prediction of $f_H(h)$ is currently the most pressing need to be met in order for the potential usefulness of the *PA* to be fully exploited for the purpose of query scope selection.

A simplified prediction agent (*simple-PA*), also proposed here, represents a practical solution to facilitate the trade-off between ψ and τ. It is shown here in simulation results that the *simple-PA* advantageously applies *2-ERS* with increasing likelihood as its expected cost in terms of hop delay (η) decreases, as desired. In fact, for the simulation cases under consideration, the *simple-PA* actually outperformed the PMF-based *PA*.

The *simple-PA* is a lightweight procedure in terms of both communication overhead and computation overhead. It is, therefore, suitable for the MANET environment.

Appendix

Here a derivation of (6) is provided. Λ_2 is computed as follows.

$$\Lambda_2 = \frac{f_H(2)}{\sum_{h=2}^{D} f_H(h)} \quad \text{(A-1)}$$

Further, ψ_1 and ψ_2 are defined as the expected *additional* cost of *1-ERS* and *2-ERS*, respectively, when an initial 1-hop query fails. Under the pessimistic assumption that a NWB of a RREQ incurs $|V|-1$ packet transmissions, ψ_1 and ψ_2 are computed as follows:

$$\psi_1 = |V| - 1 \quad \text{(A-2a)}$$

$$\psi_2 = \deg(v) + |V| - \Lambda_2 \cdot (|V| - 1) \quad \text{(A-2b)}$$

2-ERS provides, on average, a fractional reduction in the cost of NWB (as incurred by *1-ERS*) of at least ε when the following inequality holds:

$$1 - \varepsilon \geq \frac{\psi_2}{\psi_1} = \frac{\deg(v) + |V| - \Lambda_2 \cdot (|V| - 1)}{|V| - 1} \quad \text{(A-3)}$$

Letting (A-3) be an equality and solving for the value of the threshold Λ_2 in terms of ε, yields the threshold expressions given by (6a) and (6b) when ε is set to ε_+ and ε_-, respectively.

Acknowledgements

The anonymous reviewers provided numerous insightful comments and helpful suggestions. As a result, their reviews helped considerably to improve this paper.

References

[1] C. E. Perkins, E. M. Royer and S. R. Das, "Ad hoc on-demand distance vector (AODV) routing," *IETF Internet Draft*, draft-ietf-manet-aodv-07.txt, 24 November 2000.

[2] D. B. Johnson, D. A. Maltz, Y.-C. Hu and J. G. Jetcheva, "The dynamic source routing protocol for mobile ad hoc networks," *IETF Internet Draft*, draft-ietf-manet-dsr-04.txt, 17 November 2000.

[3] J. Li, J. Janotti, D. De Couto, D. Karger and R. Morris, "A scalable location service for geographic ad-hoc routing," *Proc. ACM/IEEE MobiCom*, August 2000, pp. 120-130.

[4] B. Karp and H. T. Kung, "GPSR: Greedy perimeter stateless routing for wireless networks," *Proc. ACM/IEEE MobiCom*, August 2000, pp.243-254.

[5] J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu and J. Jetcheva, "A performance comparison of multi-hop wireless ad hoc network routing protocols," *Proc. ACM/IEEE MobiCom*, October 1998, pp. 85-97.

[6] D. A. Maltz, J. Broch, J. Jetcheva and D. B. Johnson, "The effects of on-demand behavior in routing protocols for multihop wireless ad hoc networks," *IEEE J. Select. Areas Commun.*, Vol. 17, No. 8, August 1999, pp. 1439-1453.

[7] Y.-C. Hu and D. B. Johnson, "Caching strategies in on-demand routing protocols for wireless networks," *Proc. ACM/IEEE MobiCom*, August 2000, pp. 231-242.

[8] P. Johansson, T. Larsson, N. Hedman, B. Mielczarek and M. Degermark, "Scenario-based performance analysis of routing protocols for mobile ad-hoc networks," *Proc. ACM/IEEE MobiCom*, August 1999, pp. 195-206.

[9] A. Iwata, C.-C. Chiang, G. Pei, M. Gerla and T.-W. Chen, "Scalable routing strategies for ad hoc wireless networks," *IEEE J. Select. Areas Commun.*, Vol. 17, No. 8, August 1999, pp. 1369-1379.

[10] S. R. Das, C. E. Perkins and E. M. Royer, "Performance comparison of two on-demand routing protocols for ad hoc networks," *Proc. IEEE Infocom*, April 2000, pp. 3-12.

[11] 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/IEEE MobiCom*, August 1999, pp. 151-162.

[12] W. Peng and X.-C. Lu, "On the reduction of broadcast redundancy in mobile ad hoc networks," *Proc. MobiHoc Workshop*, August 2000, pp.129-130.

[13] Y.-B. Ko and N. Vaidya, "Location-aided routing (LAR) in mobile ad hoc networks," *Proc. ACM/IEEE MobiCom*, October 1998, pp. 66-75.

[14] R. Castaneda and S. R. Das, "Query localization techniques for on-demand routing protocols in ad hoc networks," *Proc. ACM/IEEE MobiCom*, August 1999, pp. 186-194.

[15] C. E. Perkins and E. M. Royer, "Ad-hoc on-demand distance vector routing," *Proc. IEEE WMCSA '99*, February 1999.

[16] D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks," *Mobile Computing*, Editors T. Imielinski and H. Korth, Ch. 5, pp. 153-181, Kluwer Academic Publishers, 1996.

[17] J. Sucec and I. Marsic, "An efficient distributed network-wide broadcast algorithm for mobile ad hoc networks," *CAIP Technical Report*, TR-248, July 2000.