

# Supporting QoS Routing in Mobile Ad Hoc Networks using Probabilistic Locality and Load Balancing \*

Ehab S. Elmallah  
U. of Alberta  
Computing Science  
Edmonton, T6G 2H1,  
Canada

Hossam S. Hassanein  
Queen's University  
Comp. and Info. Science  
Kingston, K7L 3N6,  
Canada

Hosam M. AboElFotouh  
Kuwait University  
Math. and Comp. Science  
Kuwait, P.O. Box 5969

*Abstract*— Harnessing the time varying topological aspect of mobile ad hoc networks so as to support Quality of Service (QoS) measures is a challenging problem. In this paper we develop and investigate the use of a simple spatial probabilistic locality model to enhance the performance of current on-demand routing algorithms. To explore the applicability of our approach, we examine two basic routing problems. The first problem calls for evaluating the likelihood that a given source-destination route exists, given that each mobile host on the route can be in any position of its locality set. The second problem calls for choosing the most probable route between a given source-destination pair that avoids traffic bottlenecks. In each case, we formalize a suitable problem, and devise an efficient solution strategy.

## I. INTRODUCTION

Recent progress in mobile wireless networking has initiated intensive research towards seamless integration between mobile users and the Internet. Mobility, however, poses a great challenge in the architecture of both mobile cellular and mobile ad hoc networks. Relatively speaking, however, the situation for mobile cellular networks is conceptually simpler than the situation for mobile ad hoc networks: in a cellular network a mobile host reaches a base station either directly, or perhaps through an intermediate mobile foreign agent; moreover, the base stations can predict the location of the host. In contrast, in a wireless ad hoc network, mobile hosts move freely with no supporting network backbone. Presently, much progress on mobility prediction in cellular networks has been achieved. The prediction algorithms may take into account such detailed parameters as the past locations of the user, his speed, directionality, and pausing times. The availability of such detailed models has provided support for developing recent proposals on signaling and reservation protocols that can accommodate various classes of services in cellular networks (see, for example, [1], [12], [15], [16],

and [17]). In contrast, in ad hoc networking it is not clear what level of detail a useful mobility model might have, how to estimate the parameters of the model, and how to fruitfully apply such information to enhance the performance of current sophisticated routing algorithms. In this regard, the current literature on mobility modeling for cellular networks offers a little reuse.

In this paper, we approach the mobility modeling problem by adopting a simple and intuitive behavioural model whereby each user moves among a number of geographic locations (e.g., a residence, a workplace, a playground, etc.) spending some time (with a known probability) in each location before moving to another location. The collection of such locations form the user's *locality set*. That is, we consider environments where the users exhibit a probabilistically predictable spatial behavior. In doing so, we deviate from using purely random high mobility (zero pause) models that are commonly used in creating stressful test environments. Our goal is to investigate the opportunities offered by such a simple model to enhance the performance of existing dynamic routing algorithms. At this preliminary stage, however, the emphasis is on identifying problems that appear most useful to tackle, and on devising possible solutions. To achieve this goal, a number of practical considerations that arise is estimating the model parameters, and how the parameters are exchanged between the mobile users are not fully addressed.

Our main contributions are two algorithms: function `evalRoute()` computes the likelihood that a given end-to-end path exists (cf. Section 3), and function `selectRoute()` computes an approximate solution to an optimization problem that calls for selecting a most probable route that balances the nodal loads (cf. Section 4). We now discuss possible opportunities for using our findings as optional optimization steps in existing dynamic routing protocols. In particular, we consider the class of *on-demand* routing algorithms [4, 9, 10, 13, 14, 19] that has been shown to be effective in ad hoc networks. Of the proposed routing algorithms in this class, *Dynamic Source Routing* (DSR) [9], and

\*This research is supported by NSERC Canada and Communications and Information Technology Ontario.

*Ad-hoc On-Demand Distance Vector Routing (AODV)* [14] have received particular attention, and subjected to detailed comparative analysis [6, 11]. To make our points concrete, we reproduce (and establish close ties with) some of the important findings of the recent comparative study of [6].

Firstly, we note that a key feature in the strength of dynamic routing protocols is their ability to limit the flooding process inherent in path discovery when a route reconstruction is required. DSR, for example, stores the routes it has “learned” in a *route cache*, where more than one route for each destination can be stored. If an intermediate node on a route detects a broken connection to the next-hop, it searches its cache for an alternate route. The detailed simulation study of [6] shows that DSR has a low *normalized routing load* (defined as the ratio of the routing packets transmitted per data packet delivered at the destination). The study attributes this feature to the high hit ratio to DSR’s routing cache, which results in lower use of the route discovery process.

The above finding suggests that using route cache in dynamic routing is a clear value proposition that can be enhanced further by “smart” management of the cache’s content. In an environment where spatial locality can be estimated probabilistically, one can hope of extracting information that allow the ranking of routes in the cache according to their existence likelihood. In our work, this ranking capability can be provided using function `evalRoute()`. Additionally, if the cache is empty for a particular source-destination request, one may resort to generating new routes that have plausible success probability, and add them to the cache. This latter aspect is addressed along with the next point.

Secondly, it has been long believed that the performance of ad hoc networks routing protocols is enhanced with reduced nodal mobility. Nevertheless, the detailed simulation studies of [6], and [8] have shown that, to the contrary, the end-to-end delays for both DSR and AODV increases with very low mobility. In [6], this rather unexpected behaviour has been attributed to a high level of network congestion and multiple access interferences at certain regions of the ad hoc network, combined with the fact that neither protocol has any mechanism for *load balancing* (i.e., choosing routes in such a way that the data traffic can be more evenly distributed in the network). The phenomenon is less visible with higher mobility.

We remark that both findings suggest that it is worthwhile considering optimization problems that call for generating routes that have plausible success probability, and (simultaneously) have the potential of avoiding bottlenecks, and achieving a reasonable balance between mobile loads. Our work supports this feature through function `selectRoute()`.

The rest of the paper is organized as follows. Section 2 gives a formal account of the probabilistic locality model.

Section 3 discusses the complexity of evaluating the likelihood that a given end-to-end path exists, and introduces function `evalRoute()`. Section 4 formalizes a multi-criteria optimization problem that calls for selecting a most probable route that balances the nodal loads, and describes function `selectRoute()`. Finally, Section 5 draws some conclusions.

## II. THE NODAL LOCALITY MODEL

We start by drawing some remarks on the existing literature. First, a commonly used model to study random graphs starts with a fixed set of  $n$  nodes, and then assumes that edges are chosen randomly, and *independently* from each other, with some known probability (see, for example, [2]). This model has been used in the early work on wireless networks. However, it falls short of capturing the dependency between links when a mobile host moves around. Modifying the conventional random graph model to account for joint probabilities offers no use of the existing results. In addition, the modified model is less intuitive, with parameters that are hard to estimate.

Second, in a remarkable paper by [5], some basic results of Kolmogorov complexity, in conjunction with results from random graphs, have been used to show a probabilistic guarantee on the convergence of a distributed algorithm for building a backbone in a mobile network. The technique circumvents the link dependency problem of conventional random graph techniques, however, is not suitable to achieve our goals.

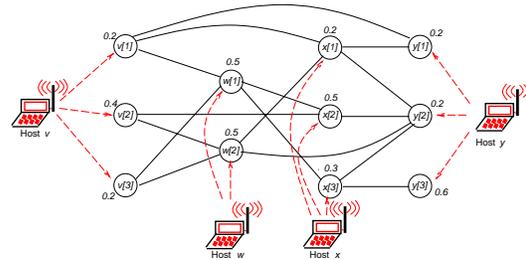


Fig. 1 An illustration of the probabilistic locality model.

Here, we adopt a simple model in which any arbitrary host  $x$  moves around between a set  $\{x[1], x[2], \dots, x[\ell]\}$  of geographic locations ( $\ell$  may differ from one host to another), pausing at each location for some arbitrary interval of time. We call such a set the *locality set* of  $x$ , and denote it  $P_{os}(x)$ . Figure 1 is used as a running example, where  $|P_{os}(v)| = |P_{os}(x)| = |P_{os}(y)| = 3$ , and  $|P_{os}(w)| = 2$ . In this preliminary work, we don’t assume knowledge of any pausing time distribution. We assume, however, that one can estimate the probability  $p_x[i]$  that host  $x$  is at location  $x[i]$  (a cooperating host may choose to announce its expected spatial behaviour over a certain period of time). That is,

$\sum_{x[i] \in \text{Pos}(x)} p_x[i] \leq 1$  (the ' $\leq$ ' accounts for partial knowledge of the host positions).

The connectivity relation is denoted  $L$ , and is defined as follows. If mobile  $x[i]$  can reach mobile  $y[j]$  directly, then  $L$  contains the (unordered pair)  $(x[i], y[j])$ . Equivalently, we write  $L(x[i], y[j]) = 1$  (or, 0 otherwise). If we denote the set of all hosts by  $B$ , and let  $\text{Pos}(B) = \bigcup_{x \in B} \text{Pos}(x)$ ,  $p(B) = \bigcup_{x \in B} \{p_x[1], p_x[2], \dots\}$  then the ad hoc network can be represented by the sequence  $(B, \text{Pos}(B), p(B), L)$ .

### III. ROUTE RANKING

We now address the first problem: at some point the mobile station discovers that the current route is no longer valid, and tries to find an alternate route while avoiding executing a path discovery phase. In this section we show that if the mobile has some older routes, it can rank them according to their likelihood. In the next section, we show how the model can be utilized to proactively generate optimized routes for the mobile host to try before flooding. In its simplest form, the problem can be stated as follows.

**[P1]** Given a source mobile  $s$ , and a destination mobile  $d$  at specific locations, and an  $(s, d)$ -route (a simple path)  $R$ , compute  $\text{Prob}[R]$ .

Two remarks are now in order. Firstly, the exact computation of  $\text{Prob}[R]$  depends not only on  $R$ , but also on the way the communication protocol handles *overheard* packets (that is, packets not intended to the node) at each of the receiving nodes on  $R$ . To illustrate, consider for example, the  $(s, d)$ -route  $R = (s, v, w, x, y, d)$  of Figure 2.a, where the dashed lines correspond to connections that give rise to overhearing over the route  $R$ .

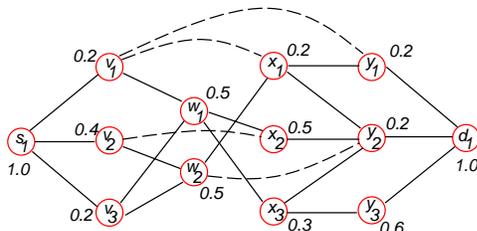


Fig. 2 (a) overhearing packets on a route

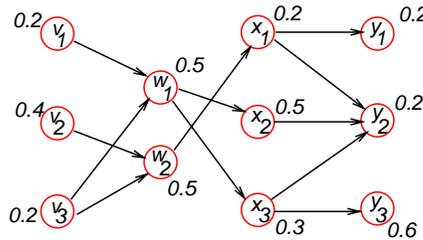


Fig. 2 (b) a  $(v, y)$ -route with source and destination location uncertainties

In dealing with overheard packets, one may distinguish two extreme possibilities: either the overheard packets at each

mobile node are dropped, or else, utilized to form more efficient routes. Exploiting overheard packets in the latter way is used, for example, in the *promiscuous listening* optimization feature of the DSR protocol. In our preliminary investigation presented below, we adopt the viewpoint that the communication protocol indeed exploits such route refinement possibilities so that the routes stored in the cache result in negligible overhearing. When applied to Figure 2.a, the assumption implies that non of the dashed lines exists in  $R$ , if  $R$  is a route stored in the cache at the connection breaking time.

As a second remark, the route evaluation process may be invoked at a time when the source and/or the destination are either engaged in (or anticipate) moving frequently between a certain subset of their respective locality sets. Figure 2.b illustrates a  $(v, y)$ -route  $R = (v[1, 2, 3], w[1, 2], x[1, 2, 3], y[1, 2, 3])$ , where for example, the source node  $v$  anticipates moving between 3 positions. The sought evaluation process should consider such source/destination uncertainties along with similar uncertainties for the intermediate nodes. Considering the above new aspect leads to the following refined problem formulation:

**[P2]** As in **[P1]**, except that each of  $s$  and  $d$  can be in any one position of a prescribed locality set  $\text{Loc}(s) \subseteq \text{Pos}(s)$ , and  $\text{Loc}(d) \subseteq \text{Pos}(d)$ .

We now devise a solution to **[P2]** that scales down to solve **[P1]** without a significant loss of efficiency. For each mobile  $x$  in  $R$ , let  $\text{Loc}(x)$  denote a subset of  $\text{Pos}(x)$  that best represent the possible locations of  $x$  that contribute to computing  $\text{Prob}[R]$ . With each  $x \in R$ , we associate an aggregate probability vector  $c_x = (c_x[i] : 1 \leq i \leq |\text{Loc}(x)|)$ . Roughly speaking,  $c_x[i]$  is the probability that mobile  $x$  reaches the destination  $d$ , given that  $x$  is at the  $i$ th indexed position  $x[i]$  of its locality set  $\text{Loc}(x)$ . Function `evalRoute()` below traverses the route in reverse order (i.e., from the destination to the source) while computing the aggregate probabilities, and finally returns the required value of  $\text{Prob}[R]$ .

To illustrate the algorithm, we use the  $(v, y)$ -route of Figure 2.b. The resulting aggregate vectors after step 1 (the initialization), and step 2.3 (removing a node), are shown below. The returned value  $\text{Prob}[R] = 0.092$ .

<u>initialization</u>	<u>removing <math>y</math></u>
$c_v = [0, 0, 0]$	$c_v = [0, 0, 0]$
$c_w = [0, 0]$	$c_w = [0, 0]$
$c_x = [0, 0, 0]$	$c_x = [0.4, 0.2, 0.8]$
$c_y = [1, 1, 1]$	
<u>removing <math>x</math></u>	<u>removing <math>w</math></u>
$c_v = [0, 0, 0]$	$c_v = [0.17, 0.04, 0.21]$
$c_w = [0.34, 0.08]$	

```

function evalRoute (R, Loc)
#input:   an (s, d)-route R, and a local-
         ity set Loc(x) for each mobile
         x ∈ R
#output:  Prob[R]
1. for each mobile x ∈ R, and possible
   location index i, set:
   c_x[i] = { 1 if x is the destination
             node
             0 otherwise
2. while (R has at least two nodes) {
  2.1 Let y be the mobile farthest
      from s on R, and let x be
      y's unique neighbor on R
      # compute c_x as follows:
  2.2 for (x[i] in Loc(x))
      c_x[i] = ∑_{y[j] ∈ Loc(y)} p_y[j] c_y[j] L(x[i], y[j]);
  2.3 remove y from R;
}
3. return (∑_{s[i] ∈ Loc(s)} p_s[i] c_s[i]);

```

To prove correctness, one may verify that at the end of each iteration, if  $x \in R$ , and  $x[i] \in \text{Loc}(x)$  then the path segments accounted for in  $c_x[i]$  are: (a) topologically distinct, (b) correspond to statistically independent set of events, and (c) exhaustive.

We conclude this section by mentioning a limiting result (while omitting the proof to conserve space): as we generalize the role of the above algorithm from assessing the likelihood of a simple route to assessing the likelihood of an arbitrary chosen backbone subnetwork, the complexity of the algorithm increases dramatically. A simple NP-completeness reduction can be devised to show that computing  $\text{Prob}[R]$  is NP-complete, if we consider a backbone subnetwork  $R$  where each two mobile nodes are connected, and each node has a 3-position locality set.

#### IV. A LOAD BALANCING ALGORITHM

As mentioned in Section 1, part of the critical findings in [6] motivates the development of routing algorithms that avoid traffic bottlenecks, specially in very low mobility environments. As of the present time, we are not aware of a research work that is devoted to routing with load balancing on ad hoc networks. In this section, we pursue this direction further, in the context of using a probabilistic locality model, and taking into account the heterogeneous nature of the mobile hosts with respect to their computational and communication resources. Our approach is based on developing an optimization framework in search of routes that can be used in an advisory manner for a higher-level layer of the routing protocol. Section 4.1 introduces the key optimization parameters of concern and formalizes the problem, and Section 4.2 discusses a solution strategy.

#### A. An Optimization Framework

We consider a simple 2-parameter framework where: (a) each host  $x$  monitors its current traffic load, denoted  $u(x)$ , measured by the number of packets received and transmitted per second (for simplicity, many recent simulation studies assume fixed-length packets of 64-byte each), and (b) each host sets the maximum acceptable traffic rate it can handle, denoted  $u_{\max}(x)$ . Additionally, we assume that the mobile hosts exchange such information, so that each host has an approximate knowledge of the current load, and the maximum acceptable rate of other relevant hosts in the network.

With the above parameters, several route optimization problems can be formalized. Of the existing possibilities, we formalize the following problem. First, given an  $(s, d)$ -route (a simple path)  $R$ , define the *critical utilization ratio* of  $R$ , denoted  $ur_{\text{critical}}(R)$  as  $\max_{x \in R} (\frac{u(x)}{u_{\max}(x)})$ . That is, the hosts associated with the critical ratio are the most loaded hosts on the route  $R$ . Our problem calls for selecting the *most probable*  $(s, d)$ -route  $R$  that has the lowest possible critical utilization ratio. That is, our routing algorithm favours a route  $R$  satisfying:

$$\begin{aligned}
 \text{[P3] } & \text{maximize } \text{Prob}[R] \\
 & \text{subject to} \\
 & ur_{\text{critical}}(R) = \\
 & \min_{R' \text{ is an } (s,d)\text{-route}} (ur_{\text{critical}}(R')).
 \end{aligned}$$

We now draw some remarks about the above parameters. Firstly, compared to other QoS measures (e.g., *packet delivery fraction*, *average end-to-end delay*, and *normalized routing load*), the traffic load is conceptually easier to assess by a mobile node. Nevertheless, due to the broadcast nature of the MAC protocol, the traffic load  $u(x)$  is affected by routes passing by the node's itself, as well as its one-hop neighbours. Such unpredictable interference suggests the use of a sampling technique and a prediction method to approximate the traffic load of a mobile host at any time. For instance, one may use the exponential-weighted moving average method, commonly used in measurement-based call admission control schemes (see, for example, [3] and [7]) to approximate the host's traffic load.

Secondly, we remark that while it is obvious that  $u(x)$  should never exceed  $u_{\max}(x)$ , the existing routing algorithms cannot guarantee this; the advisory routes obtained by our devised algorithm tries to relieve the problem. Furthermore, by lowering the maximum acceptable traffic load, each mobile host can accommodate temporary overload conditions, and low battery power conditions.

Thirdly, we remark that while accounting for the interferences between neighbours is done implicitly in reality, as well as in detailed simulation experiments; other simulation models that abstract away from the MAC layer can still approximate such interferences by incorporating simple update operations. For example, if  $x$ 's current load is  $u(x)$ , then a

new end-to-end connection  $R$  affects  $u(x)$  as follows: if  $x$  is either the source, or the destination, of  $R$  then  $x$ 's new load is  $u(x) + 1$ ; else if  $x$  is an intermediate node on  $R$ , then its new load is  $u(x) + 2$ ; moreover, each connection carried by each neighbor of  $x$  adds 1 unit to  $u(x)$ .

### B. The Algorithm

Evidently, a solution strategy based on generating and examining all possible  $(s, d)$ -routes falls short of being efficient even for networks of relatively small sizes (e.g., 20 hosts). A better strategy, that is central to our discussion, proceeds as follows:

- First, sort the utilization ratio vector  $(\frac{u(x)}{u_{\max}(x)} : x \in B)$  in ascending order (i.e., from the least loaded to the most loaded hosts), and compact the vector by removing duplicate numerical entries. Call the resulting sorted sequence  $ur_{sorted}$ , and denote its length by  $|ur_{sorted}|$ .
- Second, for each numerical value  $ur_{check}$  in  $ur_{sorted}$  (taken in order, from the lowest to the highest), form a subnetwork  $N'$  that includes only the mobiles  $\{x : \frac{u(x)}{u_{\max}(x)} \leq ur_{check}\}$ . Subsequently, for each subnetwork  $N'$ , find the most probable  $(s, d)$ -route  $R$ , if one exists. Exit upon finding the first  $R$  with  $\text{Prob}[R] > 0$  ( $R$  is the required solution). Otherwise (i.e., if no such  $R$  exists), then  $d$  is not reachable from  $s$  in the original network.

The above search algorithm finds a solution to the 2-criteria optimization problem in time proportional to the product of the length  $|ur_{sorted}|$ , and the time required to find the most probable path between mobiles  $s$  and  $d$ . Unfortunately, we don't know of any polynomial time solution to solve the latter problem. To cope with the apparent intractability of finding the most probable  $(s, d)$  simple path, we propose a weaker notion of route probability that is amenable to algorithmic solution. For this purpose,

**[D1]** define an  $(s, d)$ -strand between a source  $s$  and a destination  $d$  to be a routing path where each mobile host is at some specific location of its locality set. For example,  $(v[1], w[1], x[3], y[3])$  corresponds a strand along the route  $(v, w, x, y)$  of Figure 2.b.

Clearly, if  $R'$  is any strand of a route  $R$  then  $\text{Prob}[R'] \leq \text{Prob}[R]$ , and hence a high  $\text{Prob}[R']$  can be taken as rough indicator of a high  $\text{Prob}[R]$ . Consequently, if finding the most probable  $(s, d)$ -strand can be solved efficiently, then a first approximation attempt is to use the  $(s, d)$ -route associated with the most probable  $(s, d)$ -strand. Indeed the following theorem shows that the existing rich body of shortest distance algorithms (end-to-end, and all-pairs) can be used to solve the most probable strand problem. As implied by the proof given below, the shortest distance algorithm runs on a

graph that is moderately larger than the original ad hoc network. In particular, if we let  $B' = \bigcup_{x \in B} \text{Pos}(x)$  to be the set of all positions of all mobiles in the ad hoc network, then the shortest path algorithm runs on a graph with  $2|B'|$  nodes, and  $|B'| + 2|L|$  links.

**Theorem 1.** Finding the most most probable strand can be done in polynomial time.

*Proof.*

Given a probabilistic locality model  $(B, \text{Pos}(B), p(B), L)$ , and two mobiles  $s$  and  $d$ , we construct a directed graph  $G$  as follows. Each mobile position  $x[i] \in \text{Pos}(x)$  is associated with two nodes of  $G$ :  $x^{in}[i]$ , and  $x^{out}[i]$  connected by an arc  $(x^{in}, x^{out})$  of distance  $-\log(p_x[i])$ . Moreover, for every possible link  $(x[i], y[j]) \in L$  of the ad hoc network, add two arcs  $(x^{out}[i], y^{in}[j])$  and  $(y^{out}[j], x^{in}[i])$ , each of zero distance, to the directed graph  $G$ .

One may then verify that for any two mobiles  $x$  and  $y$  in  $B$ : (i) if  $R'$  is a  $(x[i], y[j])$ -strand in the ad hoc network, then it is associated with a directed  $(x^{in}[i], y^{out}[j])$ -path in  $G$ , and (ii) if  $R'$  exists with probability  $\text{Prob}[R']$ , then the corresponding directed path in  $G$  has distance  $-\sum_{p_x[i] \in R'} \log(x[i]) = -\log(\text{Prob}[R'])$ . This completes the proof.  $\blacksquare$

```
function selectRoute (ADHOC, s, d)
#input:   a probabilistic locality model
         ADHOC of an ad hoc network,
         and two mobiles s and d
#output:  a route that approximates a
         solution to problem [P3]

1. initialize: (R, Prob[R]) = ( $\phi$ , 0);
2. construct the sorted list  $ur_{sorted}$ ,
   as outlined above
3. foreach ( $u_{check} \in ur_{sorted}$  in
           ascending order) {
3.1 ignore the mobiles  $x$  with
     $\frac{u(x)}{u_{\max}(x)} > u_{check}$ ;
3.2 foreach ( $(s[i], d[j]) \in \text{Loc}(s) \times \text{Loc}(d)$ ) {
3.2.1 find the most probable
     $(s[i], d[j])$ -strand; denote
    its associated route by  $R$ ;
3.2.2 call evalRoute( $R$ )
3.2.3 keep the route with the
    highest probability
}
3.3 if ( $R \neq \phi$ ) break;
}
4. return (R);
```

We now consider an avenue for enhancing to the above route selection algorithm. The strategy is to generate a number of  $(s, d)$ -strands, and examine their associated  $(s, d)$ -

routes to select the best route. More specifically, we consider the set  $\text{Pos}(s) \times \text{Pos}(d)$  of all possible positions that  $s$  and  $d$  can take. For each possible pair of positions, say  $s[i]$  and  $d[j]$ , we compute the most probable  $(s[i], d[j])$ -strand; the corresponding route  $R$  becomes a candidate route for selection. We now use function `evalRoute()` to compute  $\text{Prob}[R]$  for each candidate route. Function `selectRoute()` below outlines our proposed solution for problem [P3]. This concludes the route selection algorithm.

## V. CONCLUDING REMARKS

Our main focus in the paper has been on investigating basic algorithmic issues that arise in exploiting user behaviour to support QoS routing in ad hoc networks. We note that the field of assessing the behaviour of mobile users has only recently started to receive attention (see, e.g., [18]).

To this end, we formalized a simple probabilistic model that captures the user's spatial behaviour, and used the model to develop two efficient algorithms: function `evalRoute()` estimates the likelihood the all segments of a multi-hop route coexist simultaneously, and function `selectRoute()` selects a highly probable route that avoids traffic bottlenecks. The resulting algorithms can be used as optimization features that may be used by a host protocol. Simulation studies are currently underway to assess the impact of using the locality model, and the devised algorithms on some on-demand routing algorithms.

## References

- [1] N. Berier, M. Karsten, J. Schmitt, and R. Steinmetz. Mechanisms for QoS signaling in a mobile internet environment. In *MoMuC2000*, October 2000.
- [2] B. Bollobás. *Modern Graph Theory*. Springer-Verlag, 1998.
- [3] L. Breslau and S. Jamin. Comments on the performance of measurement-based admission control algorithms. In *IEEE INFOCOM*, March 2000.
- [4] T.-W. Chen and M. Gerla. Global state routing: a new routing scheme for ad hoc wireless networks. In *Proceedings of IEEE ICC '98*, 1998.
- [5] I. Chlamtac and A. Faragó. A new approach to the design and analysis of peer-to-peer mobile networks. *Wireless Networks*, (5):149–156, 1999.
- [6] S. R. Das, C. E. Perkins, and E. M. Royer. Performance comparison of two on-demand routing protocols for ad-hoc networks. In *IEEE INFOCOM*, March 2000.
- [7] S. Floyd. Comments on measurement-based admissions control for controlled-load services. Technical report, Lawrence Berkeley National Laboratory, July 1996.
- [8] P. Johansson, T. Larsson, N. Hedman, and B. Mielczarek. Routing protocols for mobile ad-hoc networks - a comparative performance analysis. In *Proceedings of the 5th International Conference on Mobile Computing and Networking (ACM MOBICOM)*, pages 195–206, August 1999.
- [9] D. Johnson and D. Maltz. Dynamic source routing in ad hoc wireless networks. In T. Imielinski and H. F. Korth, editors, *Mobile Computing*, chapter 5. Kluwer Academic Publishers, 1996.
- [10] Y.-B. Ko and N. H. Vaidya. Location-aided routing (LAR) in mobile ad hoc networks. In *ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*, October 1998.
- [11] S.-J. Lee, M. Gerla, and C.-K. Toh. A simulation study of table-driven and on-demand routing protocols for mobile ad hoc networks. *IEEE Network*, pages 48–54, July/August 1999.
- [12] I. Mahadevan and K. M. Sivalingam. Architecture and experimental results for quality of service in mobile networks using RSVP and CBQ. *Wireless Networks*, 6(3):221–234, 2000.
- [13] G. Pei, M. Gerla, and T.-W. Chen. Fisheye state routing: a routing scheme for ad hoc wireless networks. In *IEEE International Conference on Communications (ICC)*, volume 1, pages 70–74, 2000.
- [14] C. E. Perkins and E. M. Royer. Ad hoc on demand distance vector routing. In *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, pages 90–100, February 1999.
- [15] S. Singh. Quality of service guarantees in mobile computing. *Journal of Computer Communications*, 19, 1996.
- [16] A. K. Talukdar, B. R. Badrinath, and A. Acharya. MRSVP: A resource reservation protocol for an integrated network with mobile hosts. Technical Report DCS-TR-337, Department of Computer Science, Rutgers University, 1998.
- [17] A. K. Talukdar, B. R. Badrinath, and A. Acharya. Integrated services networks with mobile hosts: Architecture and performance. *Wireless networks*, 5:111–124, 1999.
- [18] D. Tang and M. Baker. Analysis of a local-area wireless network. In *Proceedings of Mobicom 2000*, August 2000.
- [19] C.-K. Toh. Associativity-based routing for ad hoc mobile networks. *Wireless Personal Communications*, 4(2):103–139, March 1997.