Hindawi Publishing Corporation
Journal of Computer Networks and Communications
Volume 2012, Article ID 234824, 10 pages
doi:10.1155/2012/234824

Research Article

Adaptive Probabilistic Proactive Routing for Dense MANETs

Abdelfettah Belghith,[1] Mohamed Amine Abid,[1] and Adel Ben Mnaouer[2]

[1] *HANA Research Group, Ecole Nationale des Sciences de l'Informatique (ENSI), University of Manouba, La Manouba 2010, Tunisia*
[2] *College of Computer Engineering and IT, Dar Al Utoum, Al Mizan, Al Falah, Riyad 13314, Saudi Arabia*

Correspondence should be addressed to Abdelfettah Belghith, abdelfattah.belghith@ensi.rnu.tn

Received 4 October 2011; Accepted 13 December 2011

Academic Editor: Liansheng Tan

Conventional proactive routing protocols, due to their inherent nature based on shortest paths, select longer links which are amenable to rapid breakages as nodes move around. In this paper, we propose a novel adaptive probabilistic approach to handle routing information in dense mobile ad hoc networks in a way to improve the proactive routing pertinence as a function of network dynamics. We first propose a new proactive routing framework based on probabilistic decisions and a generic model to compute the existence probabilities of nodes and links. Then, we present a distributed algorithm to collect the cartography of the network. This cartography is used to instantiate the existence probabilities. Conducted simulations show that our proposal yields substantially better routing validity. Nonetheless, it amounts to much longer routes. We proposed then a bounding technique to adapt and overcome this side effect and defined two probabilistic proactive routing variants. Conducted simulations show that our proposed bounded probabilistic proactive routing schemes outperform conventional routing protocols and yield up to 66 percent increase in throughput.

1. Introduction

Mobile ad hoc networks (MANETs) are spontaneous networks that do not require any infrastructure for their operations. The task of routing packets from a source to a destination is the sole responsibility of all participating nodes and is distributed among them, where a node can serve as a traffic source, a destination, or a relaying router. All nodes should cooperate, under normal conditions, to fulfill such a requirement. In these networks, nodes and links can appear and disappear spontaneously as a consequence of several facts such as the behavior of users, the depletion of energy resources, but more inherently and subtly the underlying random mobility of the different nodes. These aspects imply a dynamic and randomly evolving topology in both time and space making the routing a real challenging task.

A host of routing protocols and algorithms were proposed, though, only very few of them are actually standardized. The standardized routing protocols are classified into reactive and proactive protocols. Reactive protocols, such as DSR [1] and AODV [2], calculate routes only when needed, and as such they are supposed to generate low signaling overhead. Proactive protocols, like the Optimized Link State Routing protocol (OLSR [3]) and the Destination Sequenced Distance Vector routing protocol (DSDV [4]), establish paths for all known source-destination pairs in advance by periodically exchanging topological information, and, as a result, they are stipulated to generate more control traffic than reactive protocols. Routing overhead, nevertheless, depends on many factors such as topology, number of nodes (i.e., density), number of hops, degree and type of mobility, number of flows, and the rate at which traffic streams are established within the network.

Numerous simulation studies were conducted on different scenarios to evaluate the performance of both proactive and reactive routing protocols [5–7]. Nevertheless, due to the large number of relevant and complex events that can happen in mobile wireless ad hoc networks and their effects on the performance of the underlying protocols, the results do not necessarily agree as to which family of protocols yields better performances and lower control traffic overhead.

In this paper, we restrict our attention to proactive routing where a periodic exchange of topological messages provides each node with a certain image of the network at the beginning of each routing period. Once the routing table is updated, it will be maintained and used during the entire current routing period. As a result and as the elapsed time since the start of the routing period gets farther, the topological information collected at the beginning of the routing period becomes inaccurate leading patently to invalid paths. In such mobile networks, there is certainly a need to consider paths composed of nodes and links having large remaining residual lives. The question naturally arises as to how should we select and decide on these links and routes. Finding the most stable path must anticipate or predict topological changes. We propose a novel probabilistic approach that would improve the routing pertinence of proactive routing protocols as a function of the network dynamics. Conducted simulations confirm that our proposed probabilistic scheme improves the accuracy of proactive routing and hence yields much larger throughput.

The paper is organized as follows. In Section 2, we present some of the relevant research work done in the field. In Section 3, we define our probabilistic general framework for selecting stable routes. Section 4 presents a Markovian model which provides a simple, yet an effective mean for the computation of the probabilities of existence of network links and nodes. In Section 5, we present our network cartography collecting algorithm that is required to instantiate the existence probabilities of network links and nodes. In Section 6, we investigate, through extensive simulations, the suitability of the proposed probabilistic framework and its instantiation. We finally summarize our work in Section 7.

2. Related Work

Conventional protocols use in general the hop-count metric to compute shortest paths towards destinations. However, shortest paths are not always reliable especially in case of dynamic networks. Finding stable routes is rather the main concern for dynamic multihop ad hoc networks. Several works established already that choosing routes based on positions, battery level, and so forth of the nodes would make selected paths more resilient to topological changes. A new routing approach has then come out considering the route stability or resiliency as a fundamental routing metric. Stability-based routing is a new emergent approach which allows routing to withstand the network dynamics. It aims essentially at choosing routes which are more stable in time and hence more resilient to dynamic changes in the network topology.

The stability of a path relies on the stability of its composing links. Authors in [8] proposed to classify links based on the mobility behavior of their end point nodes. Links between stationary or very slowly moving nodes are considered as stationary links. Links which exist only for a short period of time are handled as transient links. Newly formed links are also considered to be transient as they are more likely to break down. Routing should use then stationary links whenever this is possible. A ticket-based probing procedure is proposed to find stable routes. Classification of links is previously adapted in [9] where authors used the strength of the received signal from each neighboring node to determine whether the associated link is either weak or strong. Routing is then made through paths maximizing the received signal strength.

Associativity-based routing (ABR) [10] used a new metric called associativity which defines the stability of the link between two given nodes. ABR considers that the longer the two nodes have being neighbors, the longer they would stay connected. To express its associativity, each node broadcasts periodically a Beacon to indicate its presence. Upon the reception of a Beacon, a counter associated to the generating node is updated. The counter is reset to zero if the associated node is no longer accessible. The optimal route towards a destination is the one maximizing its cumulative associativity metric. Further stability-based routing techniques can be found in [11–18]. In contrast to all of these research proposals, our present work thrives to select the appropriate links composing a given path based on an adaptive procedure that calibrates the stability of chosen links and consequently yields shorter paths.

Furthermore, few probabilistic techniques were proposed although they seem to better cope with the unpredictable behavior of ad hoc topologies. In [19], a probabilistic technique is proposed to estimate the residual lifetime of routes. Routing is then made through the ones with maximum residual lifetime. Lifetime of a route is computed as a function of the existence probability of each link which is derived from the distribution governed by the underlying mobility model. In contrast, our present work focuses rather on the existence probability of complete routes. Mobility is represented through a generic and simple behavioral analytical Markovian model based on the network cartography collected at the start of each routing period. This cartography is then used to instantiate the existence probabilities. Our probabilistic proposal yields substantially better routing validity, but it amounts to much longer routes. A simple route bounding technique is then proposed to overcome this side effect.

3. Probabilistic Framework for the Selection of Stable Routes

Throughout the paper, we consider that links are symmetric. We model a mobile multihop ad hoc network by an undirected complete graph $G = (V, E)$, where V is the set of vertices and E represents the set of undirected edges. We stress here the fact that the graph is complete but this does not mean that all nodes are mutually within transmission range of each other. Links in $G = (V, E)$ will be partitioned into two disjoints groups as defined and explained in the following paragraph.

We define two states Up (U) and $Down$ (D) for the links and nodes of $G = (V, E)$. A vertex (i.e., a node) in V is said to be in state U at time t if it is *actively* connected to the network; notice that a node can be connected but not active when, for instance, it is forced to be in a dozing state by a power saving mechanism. A vertex is said to be in state D at time t if it is not active independently of being connected or not. An edge in E is said to be in state U when its end point vertices are within propagation range of each other independently of the states in which they are; otherwise, the edge is said to be in state D. An edge is said to be in state U means only that its end points are within propagation range independently of whether they are active or not. We note here that E is indeed partitioned into two groups of edges: a group of U edges and a group of D edges. The actual topology of the network at time t is then provided by the subgraph of $G = (V, E)$ that contains only the U nodes and their corresponding U edges. This subgraph may surely not contain some U edges exactly those not having both of their end point vertices in the U state.

3.1. Probability of Link Existence. In ad hoc networks, a link exists whenever its two end point nodes exist (i.e., they are in the U state) yet they are within transmission range (i.e., the link is in the U state). Let x and y be the end vertices of edge e, the probability of existence of e at time t is then given by

$$P(e,t) = P_U(x,t) \cdot P_U(y,t) \cdot P_U(e,t), \quad (1)$$

where $P_U(x,t)$ is the probability that node x is in state U at time t, $P_U(y,t)$ is the probability that node y is in state U at time t and $P_U(e,t)$ is the probability that nodes x and y are within transmission range at time t. We note here that we are tacitly assuming the independence between the different nodes and links. This might not be the case in real-life networks. However, later we shall focus solely on the dynamics of the links and assume that all nodes in the network are kept permanently active. In this case, there is no need of such an assumption as links are independent.

3.2. Probability of Path Existence. We define a path (s, d) as a sequence of n undirected links from a source s to a destination d. A path exists if all its nodes and its links are in state U. Let $S_v = \{n_0 = s, n_1, \ldots, n_i, \ldots, n_n = d\}$ be the sequence of $n + 1$ vertices and $S_e = \{e_1, e_2, \ldots, e_i, \ldots, e_n\}$ the sequence of n edges composing path (s, d). The probability of existence of the path from a source s to a destination d at time t is then given by

$$P((s,d)) = \prod_{n_i \in S_v} P_U(n_i,t) \prod_{e_i \in S_e} P_U(e_i,t). \quad (2)$$

Here again, we assumed a complete independence among links, among nodes, and between links and nodes. Our objective is to find the optimal path between any two given nodes. The optimal path is the one having the greatest probability of existence among all possible paths. Let $T_{s,d}$ be the set of all possible paths from source s to destination d. We call $T_{\text{opt}}\ (s,d)$ the most stable path (i.e., having the greatest probability) from a source s to a destination d, that is

$$T_{\text{opt}}(s,d) : \arg \max_{T \in T_{s,d}} P(T), \quad (3)$$

and, consequently

$$T_{\text{opt}}(s,d) : \arg \max_{T \in T_{s,d}} \ln P(T), \quad (4)$$

using now (2), we obtain

$$T_{\text{opt}}(s,d) : \arg \max_{T \in T_{s,d}} \left(\ln \left(\prod_{n_i \in T} P_U(n_i,t) \prod_{e_i \in T} P_U(e_i,t) \right) \right). \quad (5)$$

which leads to

$$T_{\text{opt}}(s,d) : \arg \min_{T \in T_{s,d}} \left(\sum_{n_i \in T} - \ln P_U(n_i,t) + \sum_{e_i \in T} - \ln P_U(e_i,t) \right). \quad (6)$$

The solution to (6) may be readily provided by any shortest path algorithm executed on the corresponding valued graph where every edge e has a weight $-\ln P_U(e,t)$ and every vertex has a weight $-\ln P_U(n,t)$ (see for instance [20]).

The above represents a generic framework to compute probabilistic optimal paths. It remains to devise how the probability for a node or a link to be in the U or D state is calculated. In the following section, we present a novel method to compute these probabilities.

4. Markovian Model for the Existence of Network Elements

According to the dynamics of the network, nodes and links of our complete graph $G = (V, E)$ switch between state U and state D in a completely independent manner (by our independence hypothesis stated above). At any instant, the actual network is the one composed of the U vertices and the corresponding U edges. To model this dynamic behavior, we propose to view each node and each link of the complete graph $G = (V, E)$ as a Markovian two state automaton.

4.1. Probability of Existence of Vertices and Edges. Let x be a vertex or an edge of $G = (V, E)$. We model state changes of x by a 2-state continuous time Markov chain where the residence time in state U, respectively, in state D, is exponentially distributed with parameter λ, respectively, with parameter μ. Let $P_U(x,t)$ denote the transient probability of element x being in state U at time t. Let also $P_D(x,t)$ denote the transient probability of element x being in state D at time t. Our objective is to obtain the transient probability $P_U(x,t)$. The solution to this transient probability should then obey to the following differential equations [21]:

$$P_U(x,t) + P_D(x,t) = 1,$$
$$\frac{dP_U(x,t)}{dt} = \mu P_D(x,t) - \lambda P_U(x,t) \cdot \quad (7)$$

The solution of these equations is of the form

$$P_U(x,t) = \frac{\mu}{\lambda + \mu} + B \exp^{-(\lambda+\mu)t}. \quad (8)$$

Let us see the time axis divided into intervals representing the routing periods. At the beginning of a routing period, that is at time $t = 0$, we assume that the state of each and every element of the complete graph is known. This indeed necessitates the complete knowledge of the network cartography. For now, we suppose that a certain oracle is there to give us this cartography, we will develop on that later on. As a result, we readily have

$$P_U(x, at\ t = 0) = 1, \quad \text{if } x \text{ is } U \text{ at } t = 0,$$
$$= 0, \quad \text{if } x \text{ is } D \text{ at } t = 0, \quad (9)$$

if we define

$$\delta = \frac{\mu}{\lambda + \mu}, \tag{10}$$

then, the final solution for $P_U(x,t)$ will be

$$\begin{aligned} P_U(x,t) &= \delta + (1-\delta)\exp^{-(\lambda+\mu)t}, \quad \text{if } x \text{ is } U \text{ at } t = 0, \\ &= \delta - \delta\exp^{-(\lambda+\mu)t}, \quad \text{if } x \text{ is } D \text{ at } t = 0. \end{aligned} \tag{11}$$

The steady state probabilities of having element x in state U denoted by $P_U(x)$ or in state D denoted by $P_D(x)$ are readily given by

$$P_U(x) = \delta, \qquad P_D(x) = 1 - \delta. \tag{12}$$

In the rest of the paper, we shall focus solely on links' dynamics and assume that all nodes are kept permanently active. We are then restricting the network dynamics to be solely driven by the mobility of the nodes. Our aim is then to investigate the impact of node mobility on proactive routing tables and at the same time demonstrate the efficiency of our probabilistic model. Recall that there is no need here to consider the aforementioned independence assumption.

4.2. Computing Links Parameters. Link stability depends on the actual distance separating its end points and their relative mutual mobility. Let R denote the adopted transmission range. Let d_e denote the distance separating two given nodes. These two nodes are capable of communication only if d_e is less than or equal to R. However, the shorter is d_e, the more stable is the corresponding link. Such a stability, as a function of time, depends necessarily on their relative speeds and the adopted mobility model. For a given predefined mobility, this amounts to consider λ and μ of any link as functions of the distance separating their corresponding end points. The question here is what functions of d_e should we use for λ and μ. To answer this question, we rely on the tacit observation that the transient state of a Markov process decays rather very rapidly to reach its steady state [21]. Consequently, we can discuss our functions by considering and relying on the steady state probabilities given by (12).

Consider a link that is known to be in state U at the beginning of the routing period. That is d_e for this link is not larger than R at the start of the routing period. Its transient probability of being Up starts equal to one and then will decay rapidly to δ its steady state probability. This steady state probability should tend to one as the distance d_e gets smaller; yet it should tend to one half as d_e approaches R.

Now, let us consider a link in state D at the beginning of the routing period, that is, its d_e is larger than R. Its transient probability of being in state D starts equal to one (its transient probability of being in state U starts equal zero) and then will decay rapidly to $(1-\delta)$ its steady state probability. The later should tend to one as d_e gets farther and should equal one half as d_e approaches R.

The previous discussion amounts to consider λ as an increasing function (resp., μ as a decreasing function) of d_e/R. The rate at which λ increases (resp., μ decreases) as a function of d_e/R is of utmost concern since it reflects the degree at which shorter hops are preferred in the selection process. The larger such a rate of increase of λ (resp., such a rate of decrease of μ), the more we select shorter hops. Extensive experiments and tests dictate the usefulness of the following equations where we adopt the value $k = 4$:

$$\lambda = \left(\frac{d_e}{R}\right)^k, \qquad \mu = \left(\frac{R}{d_e}\right)^k. \tag{13}$$

5. Cartography-Augmented DSDV

Now, we turn to our earlier assumption about the oracle that provides us, at the start of each routing period, with the actual real network cartography which gives us the different distances separating the different nodes. Our aim here is then to design a distributed algorithm that collects the cartography of the network at the beginning of each routing period. The network cartography can be collected by any proactive routing algorithm by appropriately integrating the cartography information in its signaling messages. More specifically, we here propose to augment the Destination Sequenced Distance Vector routing DSDV protocol [4] that uses the distributed BellmanFord [22, 23] algorithm to calculate paths. It requires that each node in the network periodically generates and sends a Hello message to inform its neighbors of any detected topology changes.

First of all, we intend to get a correct and valid cartography, and, therefore, we will not tolerate delayed routing information. As such, we propose to distinguish between control (i.e., Hello) messages and data packets. Hello messages are to be transmitted as soon as possible before any other data packets. As such, received or locally generated Hellos are put at the head of the IP sending queue in front of any awaiting data packets. Secondly, we assume that each node is capable of knowing its own geographical location. Recent availability of small and inexpensive low-power GPS receivers and approaches for inducing relative coordinates based on signal strength provides a justification for such an assumption [24]. For instance, the APS protocol [25] is a distributed hop by hop positioning algorithm that approximates the absolute positions of all nodes given that only a small fraction of nodes possess a self-positioning capability. The APS algorithm works as an extension to a distance vector proactive routing protocol, and, as such, it can be assumed for this current work. Consequently, before forwarding a Hello or sending its locally generated Hello message, a node includes its own perceived position. Note in particular that a forwarded hello contains both its originating node's position and its forwarding node's position.

The cartography capable augmented DSDV protocol works as follow.

(1) It is through Hello messages that a node can maintain its routing table up to date. The routing table includes entries to already heard destinations. Each entry includes a creation date (i.e., the period's number), the entry local start instant (recall that an entry is deleted after a maximum duration period

(Max_duration_period)), the current perceived position of the destination node, and the current perceived position of the next hop node.

(2) Each node generates locally a Hello message every Hellomsgperiod_DSDV period. The Hello message contains the following fields.

(i) Source address: the address of the node that originated this Hello.

(ii) Current period number: works as a sequence number since synchronization is supposedly maintained throughout the network.

(iii) Next hop address: the next hop to reach the source address, this is just the address of the node that sent this Hello.

(iv) Number of hops: representing the cost.

(v) Source position: the source node coordinates at the Hello generation instant.

(vi) Next hop position: the sender node coordinates at the forwarding instant.

(3) Upon receiving a Hello message, a node first consults the sender's relative fields (they necessarily contain new information). If no entry exists in the routing table to reach the node pointed by the next hop address field, then an entry is added in the table. Otherwise, if the entry is relative to a previous period, then the entry is updated accordingly. A Hello is automatically created to advertise this new entry to other nodes in the network.

(4) Then, the node checks whether it is the originator of this Hello, in which case it just discards the message. If the Hello is relative to an older period, it will be discarded too. Otherwise, the node tests the usefulness of the received hello: if its routing table contains already an entry for the *Source Address* (which represents the destination to be reached), the creation time of the entry and the sequence number included in the Hello are compared. The entry is updated if it is older, or when the two sequence numbers are equal, but the message proposes a better cost. A new entry is automatically created in the routing table, if the message offers a path to a destination not yet known.

(5) If the received message is useful, the node forwards it to its neighbors. It puts its address in the *Next hop Address* field, puts its position in the corresponding field, increments the *Number of hops*, and then broadcasts it to its neighbors.

From the above discussion, we readily observe that our cartography is solely based on the underlying routing protocol in effect; namely, here the DSDV protocol. More interestingly, we do not require any additional control traffic to build the network cartography. The exact control traffic of DSDV is sufficient, but where each Hello message is augmented to include the originator and the forwarder positions.

5.1. Cartography Validity Definition. The network cartography is built by every node through the dissemination of geographic information integrated in the routing announcements as described above. What we need, for our current purpose, is a collected cartography that reflects, the best possible, the actual real network cartography at the start of the routing period. To evaluate the correctness of the collected cartography, we compared it against the real actual cartography of the network. Note that the actual instantaneous cartography of the network can be extracted from the simulator but it cannot be known in practice. This is of no concern to us here since all what we are searching for is to be confident that the collected cartography does represent adequately the real actual cartography at the start of the routing period.

Consider a target node N. When N advertised itself (i.e., sent its own generated Hello), it was at position (x_0, y_0). In the routing table of a node A that had already heard N's Hello, a new entry for N was created, showing (x_0, y_0) as N's coordinates. Since node N is mobile, its position varies as a function of time, and, consequently, it will be at position (x_t, y_t) at time t during the same current routing period. We say that N's position, as indicated by A's current cartography, is valid as long as the distance between the recorded position (x_0, y_0) and the actual current position (x_t, y_t) is less than a tolerated predefined value denoted by d. That is $\sqrt{(x_t - x_0)^2 + (y_t - y_0)^2} \leq d$. The validity of the cartography, as perceived by any given node, say node A, represents the percentage of nodes having valid positions among all nodes. d is a tuning parameter whose value is relative to the transmission range used and is in general a small fraction of this range.

5.2. Simulation Set up. To ascertain the validity of the cartography as a function of the mobility, the traffic load, and the elapsed time since the start of the current routing period, we conducted an extensive set of simulations. We have considered a simulation area of 400 m by 400 m with 120 mobile nodes using the Random Way Point mobility model [17]. We used a transmission range of 100 m, a tolerance d of 10 meters that is a tolerance equals to one tenth of the used transmission range, a network capacity of 11 Mbps, and a maximum MAC retransmission count equals to 7. We used a priority IP module at the network layer to enforce that Hellos are treated before any awaiting data packet. The priority queue maximum size is 100 packets. Furthermore, this queue is handled such as a locally generated IP data packet is only accepted if less than 70 packets (data and Hellos) are present in the queue; otherwise, it is rejected at the IP level. Hello messages are only rejected (dropped) if the queue is completely full. This enforces a further layer for the priority handling of the Hello messages as they are the responsible for the cartography dissemination. The routing updating period is set to 10 seconds. Finally, the *Max_duration_period*, representing the life time of an entry in the routing table, is set to 15 seconds. All required modifications are ported on the OMNET++ network simulator.

5.3. Observations. Figure 1 portrays the validity of the cartography as a function of the elapsed time since the start of the routing period and for different node speeds. Recall that we are using a priority IP handling, and, therefore, the network load has a very little impact on the validity of the cartography. For a null node speed (no mobility), we get a validity of one hundred percent. For speeds higher or equal to 1 m/sec, the validity of the cartography gets at its maximum around instant 1.5 sec which is the time required to get the maximum of Hellos throughout the network. Consequently, if we launch the routing updating process around 1.5 seconds before the start of the period, we get the maximum cartography validity just at the start of the routing period. Moreover, as we see on Figure 1, the validity reaches one hundred percent for all considered speeds but 10 m/sec. For the later, the validity is sufficiently high and equals nearly ninety seven percent.

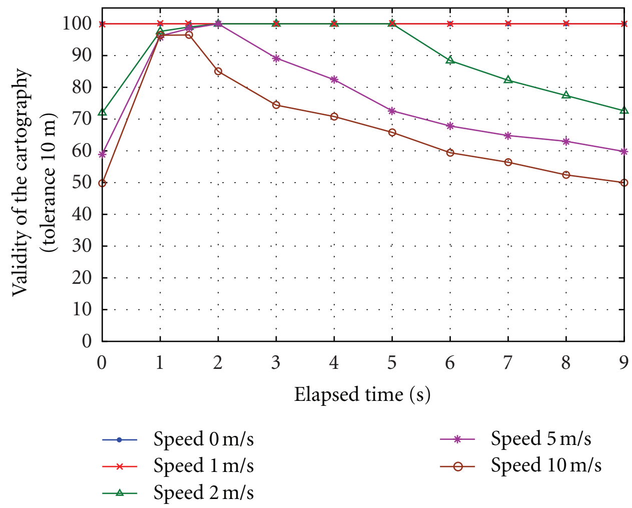

Figure 1: Cartography validity as a function of the elapsed time.

6. Suitability of the Proposed Probabilistic Model

To experiment our new probabilistic proposal, we conducted a set of simulations using the OMNET++ simulator where we integrated the proposed probabilistic framework, the cartography collecting algorithm, and the Priority IP handling. Our simulation set up is as defined previously; that is a simulation area of 400 m × 400 m, 120 mobile nodes, a transmission range of 100 meters, a tolerance of 10 meters, and the Random Way Point mobility model [17]. The time axis is seen divided into consecutive routing periods each of length T seconds. For the conventional proactive routing; namely, the DSDV, each node generates a Hello message just at the start of the routing period. However, for the probabilistic routing protocol and its derivatives which will be introduced next, each node generates a Hello message one second and a half before the end of the current period. As such, the cartography with the maximum validity is readily available just at the start of the next routing period. The initial topology of 120 nodes is chosen randomly, and the period size T is fixed to 10 seconds.

Before we can proceed further, we require here to recall that probabilistic routing inherently thrives to select shorter hops as the elapsed time from the start of the routing period gets farther. As a result, probabilistic routes are much more resilient to breakages caused by the mobility of nodes, but they are much longer than the routes used by the common proactive routing. The upper curve in Figure 2 represents the average route length in number of hops provided by the probabilistic routing, for a null speed and a null traffic load, as a function of the elapsed time since the start of the routing period. At the start of the routing period that is at instant 0, the network collected cartography provides the distances between the nodes, and, consequently, we readily get the initial probabilities for all links as given by (9). At time instant 1 second, we reach the steady state and the link probabilities are now rather governed by (12). As such, the probabilistic route average length starts around 2.8 which is the same average route length as given by both the conventional proactive routing and the real actual topology. From instant one second, the probabilistic average route length stabilizes at around 11 hops which is a rather very large value.

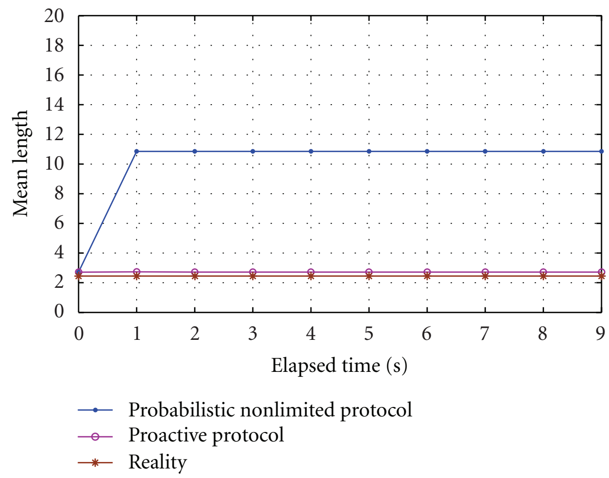

Figure 2: Average route length as a function of the elapsed time, $V = 0$ m/sec.

We clearly observe that probabilistic routing amounts to much longer routes than both the real optimal routes given by the actual network and those delivered by the conventional proactive routing. This is quite expected since the crux of the probabilistic routing is its preference and greed for short stable links. Conventional proactive routing, however, amounts to a slightly larger average route length than the real optimal case which can be clearly observed on Figure 3. Indeed, some Hello messages may arrive through longer paths due to collisions. This will not affect the cartography validity which remains at its optimum validity of one hundred percent but do slightly affect the proactive routing tables.

This very large average route length hinders any practical use of our probabilistic routing unless some definite action is undertaken to impose on it an upper limit. To mitigate the

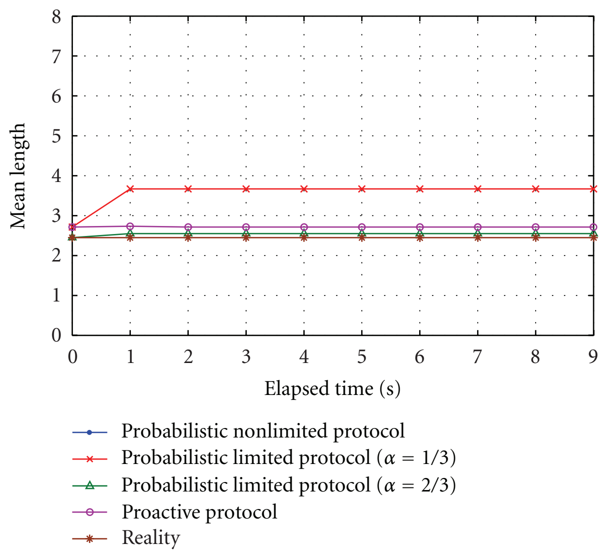

Figure 3: Average route length for $\alpha = 1/3$ and $\alpha = 2/3$, $V = 0$ m/sec.

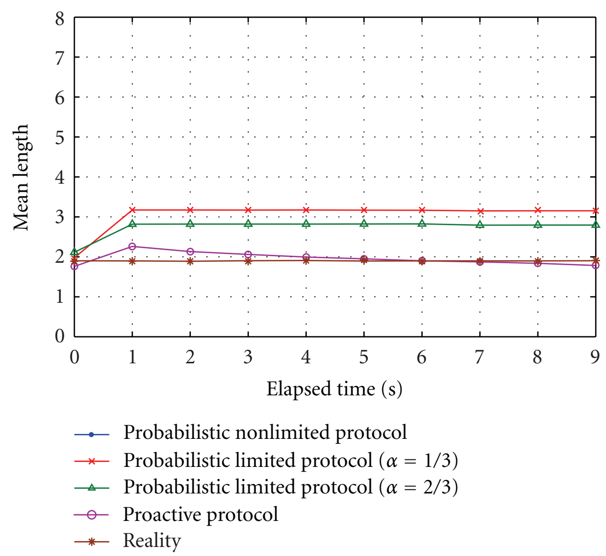

Figure 4: Average route length as a function of the elapsed time, $V = 5$ m/sec.

down side of probabilistic routing, we propose to integrate a bounding function. Let αR be a fraction of the transmission range where $(0 < \alpha \leq 1)$ and d_f be the flying distance between a source node S and a destination node D. Let $L(S, D)$ be the bounding number of hops that can be used to reach node D from node S. We readily have $L(S, D) = d_f/\alpha R$. Note here that when α goes to zero then no bounding is used and when α goes to one then only hops of length R can be used. Consequently, α is an adequate tuning parameter that can be used in conjunction with the probabilistic routing to limit the length of its provided routes. Let $L_s(S, D)$ be the length of the shortest path between S and D. Then, we readily adopt the following bounding equation governing the route length of the probabilistic routing:

$$L(S, D) = \max\left(\frac{d_f}{\alpha R}, L_s(S, D)\right). \tag{14}$$

The question naturally arises as to how to calculate now the shortest routes within our probabilistic framework. Earlier, we have just to execute a shortest path algorithm on the valued graph provided by the network cartography seen at the start of the routing period. But now, we have to further restrict these routes according to the bounding function just defined. We solved this by using the Extended Bellman-Ford Algorithm (EBFA) for shortest paths computed with 2 metrics [11]. A probabilistic route must now have the greatest probability of existence among all possible routes, yet its length should not exceed (in terms of the number of hops) the maximum threshold given by 11. In the remaining, we shall consider the value $\alpha = 1/3$ to approach the pure probabilistic routing and the value $\alpha = 2/3$ to approach rather the common proactive routing.

Figure 3 represents the average route length as a function of the elapsed time since the start of the routing period, the same as Figure 2 but without the curve corresponding to the unlimited probabilistic routing. First, we observe that the probabilistic limited $\alpha = 1/3$ provides an average route length equals to 3.8 hops which amounts to just and only one additional hop compared to the conventional proactive routing (2.8 hops) but much less than the pure probabilistic routing (11 hops, see Figure 2). This is normal since the probabilistic limited $\alpha = 1/3$ is armed to choose longer routes than the proactive routing as $\alpha = 1/3$. More interestingly, we observe that the probabilistic limited $\alpha = 2/3$ provides lower average route length than even the conventional proactive routing. To explain this, recall that this probabilistic limited protocol selects large hops as the conventional proactive routing. However, being based on the network cartography, it has a much better and comprehensive view of the network and consequently can use routes that the conventional proactive routing does not even realize and know about.

We conducted many simulations for different mobility levels. We observed that the probabilistic limited incurs a very small increase in the average route length but still remains much more efficient than the pure probabilistic protocol. For instance, Figure 4 represents the average route length but for a speed of 5 m/sec. It is interesting to note here that, as the network is dynamic, the validity of the proactive routing loses as time progresses and consequently only stable paths specially those within the closer vicinity will stay valid. As such, we notice that the average route length of the proactive routing may even get smaller than the actual real average route length. For the same exact reason, the average route length provided by the probabilistic $\alpha = 2/3$ is higher now than that given by the proactive routing. This takes us to investigate the correctness or validity of the routes given by each protocol against the actual real network.

The routing period is divided into T equally spaced observations points; namely, an observation point each second. At each observation time point, we have then five different views of the network: the actual real view provided by the underlying mobility model used in our OMNET++

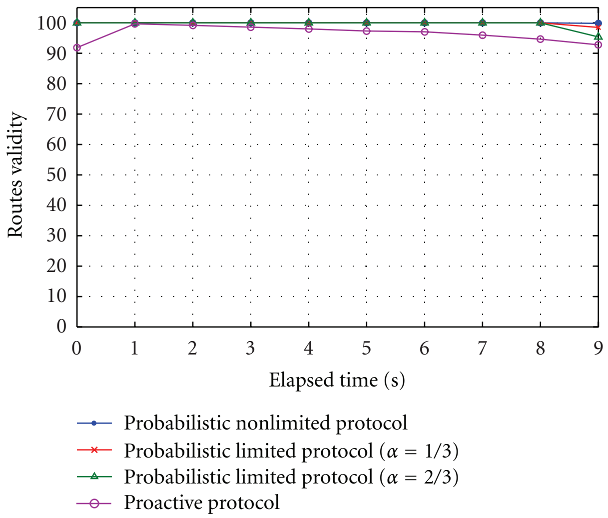

Figure 5: Percentage of valid probabilistic routes as a function of the elapsed time, $V = 1$ m/sec.

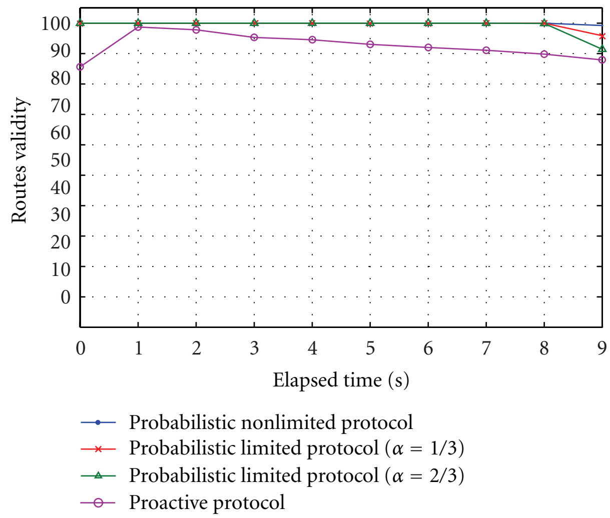

Figure 6: Percentage of valid probabilistic routes as a function of the elapsed time, $V = 2$ m/sec.

simulator, the DSDV routing table computed in the common conventional proactive way, the probabilistic nonlimited routes computed by our pure probabilistic proposed model, the probabilistic limited $\alpha = 1/3$, and the probabilistic limited $\alpha = 2/3$. A route from a source node to a destination node is termed valid if it exists in the real network regardless whether it is optimal (a shortest route) or not. To ascertain the efficiency of our proposed probabilistic routing algorithms, we first compute the percentage of valid routes from a designated node (node 1) to all the other 119 destinations given by each one of the four protocols. Figures 5, 6, 7, and 8 portray the percentage of the validity of the routes as a function of the elapsed time since the start of the routing period for all the four defined protocols when no load is applied to the network and, respectively, for the speeds 1 m/sec, 2 m/sec, 5 m/sec, and 10 m/sec. Obviously, for a speed of 0 m/sec, all four protocols provide a constant validity of one hundred percent.

We observe that as node speed gets higher, the conventional proactive routing starts losing its efficiency by delivering the lowest route validity percentage. Most interestingly, we notice that the probabilistic limited $\alpha = 1/3$ outperforms even the pure unlimited probabilistic protocol under high mobility. While this is remarkable and might not be expected, it surely needs some explanations. The crux of this phenomenon resides mainly in three different facts. Firstly, recall that the probabilistic routing chooses the path having the largest product of the links weights (the product of the probabilities of existence of the links). As a result, it selects the largest number of very short hops. The probabilistic limited, especially using $\alpha = 1/3$, provides a path comprising much less hops, and, at the same time, this path can have a very close value of the product of the links weights. For instance, the probabilistic routing selects a path with four hops, each one having a weight 3/4, while the probabilistic limited selects a path having just two hops each of which has a weight of 1/2. Secondly, the probabilistic limited inherently uses different paths for

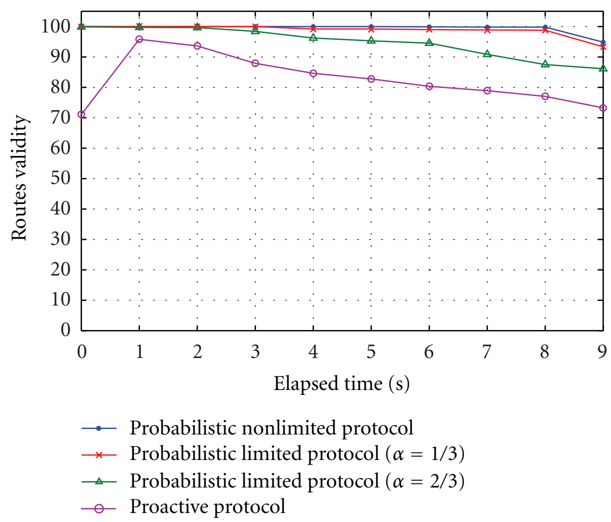

Figure 7: Percentage of valid probabilistic routes as a function of the elapsed time, $V = 5$ m/sec.

different destinations, while the probabilistic not limited is restricted to pass through the same longest path's segments as dictated by the well-known principle of optimality. Least but not last, when mobility gets higher, a breakage of just one link brings down all the probabilistic paths passing through it which are numerous according to previous observations, but very few paths in the case of the probabilistic limited. In other terms, all three probabilistic protocols provide resilient paths at the beginning of the routing period. As we get farther from the start of the routing period and as the mobility gets stronger, the probabilistic limited stands out to provide better resiliency.

Let us now investigate the betterment brought by this probabilistic framework in terms of the network throughput. We consider the same network scenario used previously; however, we also consider 8 traffic flows emanating from 8 different sources chosen randomly and destined to 8 other

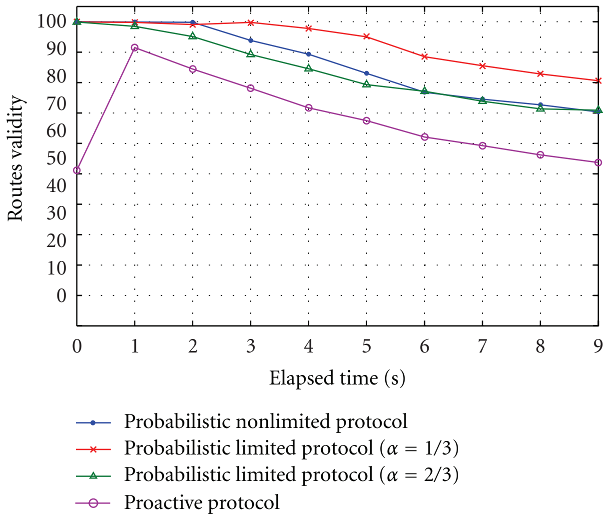

FIGURE 8: Percentage of valid probabilistic routes as a function of the elapsed time, $V = 10$ m/sec.

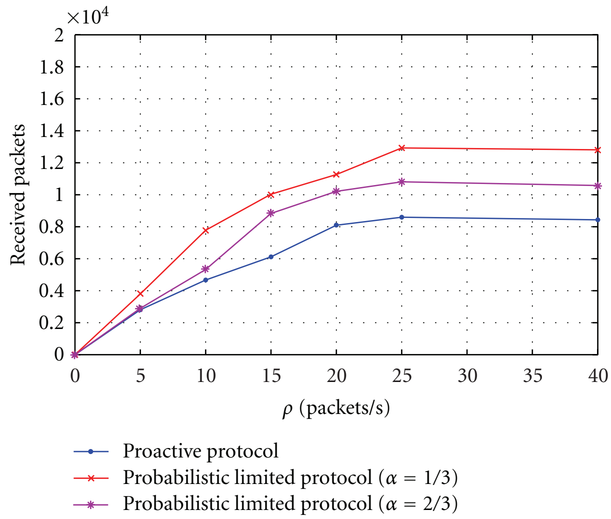

FIGURE 9: Throughput as a function of the traffic load, $V = 5$ m/sec.

destination nodes also chosen randomly. We fix the nodes' speed to 5 m/sec. The data packet size used is 200 bytes. Figure 9 portrays the average number of received packets per flow as a function of the traffic load per flow. We observe that the probabilistic limited $\alpha = 1/3$ outperforms both the conventional proactive and the probabilistic limited $\alpha = 2/3$ in delivering more throughput. The probabilistic limited $\alpha = 1/3$ stands out even for light traffic loads, for moderate traffic, for instance, 15 packets per second, it delivers around 67 percent more than the conventional proactive routing. For a higher traffic load, for example, at 40 packet per second, it delivers 50 percent more than the conventional proactive protocol. The probabilistic limited $\alpha = 2/3$ outperforms also the conventional proactive protocol; however, it delivers much less than its $\alpha = 1/3$ counterpart.

7. Conclusion

In this paper, we proposed a new probabilistic proactive routing framework that selects the most stable links to construct routes towards their destinations. A network element being a node or a link is represented as a two-state simple Markovian automaton. Furthermore, a network cartography collecting algorithm is proposed to provide distances between nodes at the start of each routing period. The cartography is then used to instantiate the required probabilities of existence of each element in the network.

Extensive simulations are conducted to demonstrate the efficiency of both the proposed probabilistic framework and the cartography collecting algorithm. Our probabilistic proposition bases its decision rather on link stability and consequently selects shorter and more resilient links. However, such an inherent behavior has a side effect of increasing the average route length. To mitigate this down side, we adopted a simple hop bounding approach that limits the number of hops within probabilistic routes. Conducted simulations showed that this is an effective technique yielding substantially much more valid routes and amounting to a substantial increase in network throughput.

References

[1] D. B. Johnson, D. A. Maltz, and Y.-C. Hu, "The dynamic source routing protocol for mobile Ad Hoc Networks (DSR)," Internet-Draft. IETF MANET Working Group, 2004.

[2] C.E. Perkins, E.M. Royer, and I.D. Chakeres, "Ad Hoc On demand distance vector routing protocol," Internet Draft, MANET Working Group, 2003.

[3] T. Clausen and P. Jacquet, "Optimized link state routing protocol (OLSR. m inus 0.4em Request for Comment 3626)," MANET Working Group, 2003.

[4] C.E. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers," in *Proceedings of the ACM Conference on Communications Architectures, Protocols and Applications (SIGCOMM'94)*, 1999.

[5] T. Clausen, P. Jacquet, and L. Viennot, "Comparative study of routing protocols for mobile Ad Hoc networks," in *Proceedings of the The 1st Annual Mediterranean Ad Hoc Networking Workshop*, 2002.

[6] J. Novatnack, L. Greenwald, and H. Arora, "Evaluating ad hoc routing protocols with respect to quality of service," in *Proceedings of the IEEE International Conference on Wireless and Mobile Computing, Networking and Communications, (WiMob'05)*, pp. 205–212, August 2005.

[7] C. Mbarushimana and A. Shahrabi, "Comparative study of reactive and proactive routing protocols performance in mobile ad hoc networks," in *Proceedings of the 21st International Conference on Advanced Information Networking and ApplicationsWorkshops, (AINAW'07)*, pp. 679–684, May 2007.

[8] W.-H. Chung, "Probabilistic analysis of routes on mobile ad hoc networks," *IEEE Communications Letters*, vol. 8, no. 8, pp. 506–508, 2004.

[9] R. Dube, C. D. Rais, K. Y. Wang, and S. K. Tripathi, "Signal stability-based adaptive routing (SSA) for ad hoc mobile networks," *IEEE Personal Communications*, vol. 4, no. 1, pp. 36–45, 1997.

[10] C.-K. Toh, "Associativity-Based Routing for Ad-Hoc Mobile Networks," *Wireless Personal Communications*, vol. 4, no. 2, pp. 103–139, 1997.

[11] M. A. Abid and A. Belghith, "Stability routing with constrained path length for improved routability in dynamic MANETs," *The International Journal of Personnal and Ubiquitous Computing*, vol. 15, no. 8, pp. 799–810, 2011.

[12] R. Beraldi, L. Querzoni, and R. Baldoni, "A hint-based probabilistic protocol for unicast communications in MANETs," *Ad Hoc Networks*, vol. 4, no. 5, pp. 547–566, 2006.

[13] H. Dubois-Ferriere, M. Grossglauser, and M. Vetterli, "Age matters: efficient route discovery in mobile ad hoc networks using encounter ages," in *Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC '03)*, pp. 257–266, Annapolis, Md, USA, June 2003.

[14] M. Roth and S. Wicker, "Termite: emergent ad-hoc networking," in *Proceedings of The 2nd Mediterranean Workshop on Ad-Hoc Networks*, Mehdia, Tunisia, 2003.

[15] M. A. Abid and A. Belghith, "Period size self tuning to enhance routing in MANETs," *International Journal of Business Data Communications and Networking*, vol. 6, no. 4, pp. 21–37, 2010.

[16] D. Yu, H. Li, and I. Gruber, "Path availability in ad hoc network," in *Proceedings of the 10th International Conference on Telecommunications (ICT'03)*, vol. 1, pp. 383–387, 2003.

[17] T. Camp, J. Boleng, and V. Davies, "A survey of mobility models for Ad Hoc network research," *Wireless Communication and Mobile Computing*, vol. 2, no. 5, pp. 483–502, 2002.

[18] H. Zhang and Y.-N. Dong, "A novel path stability computation model for wireless ad hoc networks," *IEEE Signal Processing Letters*, vol. 14, no. 12, pp. 928–931, 2007.

[19] Y. C. Tseng, Y. F. Li, and Y. C. Chang, "On route lifetime in multihop mobile ad hoc networks," *IEEE Transactions on Mobile Computing*, vol. 2, no. 4, pp. 366–376, 2003.

[20] R. Marie, M. Molnar, and H. Idoudi, "A simple automata based model for stable routing in dynamic Ad-Hoc networks," in *Proceedings of the 2nd ACM International Workshop on Performance Monitoring, Measurement, and Evaluation of Heterogeneous Wireless and Wired Network(PM2HW2N '07)*, Crete Island, Greece, 2007.

[21] L. Kleinrock, *Queueing Systems*, vol. 1: Theory, 1975, Wiley Interscience.

[22] T. H. Cormen, C. E. Leiserson, Ronald L. Rivest, and C. Stein, *Introduction to Algorithms*, MIT Press, Cambridge, Mass, USA, 2nd edition, 2001.

[23] D. Bertsekas and R. Gallager, *Data Networks*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1987.

[24] S. Giordano and I. Stojmenovic, *Position-Based Ad Hoc Routes in Ad Hoc Networks*, CRC Press, Inc, Boca Raton, FL, USA, 2003.

[25] D. Niculescu and B. Nath, "DV Based Positioning in Ad Hoc Networks," *Telecommunication Systems*, vol. 22, no. 1-4, pp. 267–280, 2003.